UNIVERSITY OF CALIFORNIA
Santa Barbara

Stochastic Search and Surveillance Strategies for Mixed Human-Robot Teams

A Dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

Mechanical Engineering

by

Vaibhav Srivastava

Committee in Charge:

Professor Francesco Bullo, Chair

Professor Bassam Bamieh

Professor João P. Hespanha

Professor Sreenivasa R. Jammalamadaka

Professor Jeff Moehlis

December 2012

The Dissertation of
Vaibhav Srivastava is approved:

Professor Bassam Bamieh

Professor João P. Hespanha

Professor Sreenivasa R. Jammalamadaka

Professor Jeff Moehlis

Professor Francesco Bullo, Committee Chairperson

October 2012

Stochastic Search and Surveillance Strategies

for Mixed Human-Robot Teams

by

Vaibhav Srivastava

to my family

Acknowledgements

This dissertation would not have been possible without the support and encouragement of several people. First and foremost, I express my most sincere gratitude towards my advisor, Prof Francesco Bullo, for his exceptional mentorship, constant encouragement, impeccable support, and the complete liberty, he granted me throughout my PhD. I have been fortunate to learn the art of research under his excellent tutelage.

I thank Prof Sreenivasa Rao Jammalamadaka for his exceptional guidance throughout my masters work and for teaching me to think from a statistical perspective. I thank my other committee members, Prof Bassam Bamieh, Prof João Hespanha, and Prof Jeff Moehlis, for their help and support throughout my PhD.

I thank all my collaborators for all their help throughout my PhD. I thank Dr Kurt Plarre for his help in the sensor selection project. I thank Prof Jeff Moehlis for his encouragement, help, and support in the bifurcations project. I thank Prof Cédric Langbort and Prof Ruggero Carli for all the helpful discussions and their encouragement during the attention allocation project. I thank Dr Fabio Pasqualetti for all his help and support throughout my PhD and especially, in the stochastic surveillance project. I thank Dr Luca Carlone and Prof Giuseppe Calafiore for all their help in the distributed random convex programming project. I thank Christopher J. Ho for all his work on the interface design for human

experiments. I thank Dr Amit Surana for inviting me for the internship at UTRC and for all the helpful discussions on the attention allocation in mixed teams.

I thank the current and the previous members of the Prof Bullo's lab for their friendship, for the intellectually stimulating discussions in the lab and for all the fun activities outside of it. I thank Dr Shaunak Bopardikar for convincing me to join UCSB. It was indeed a great decision. I thank Dr Andrzej Banaszuk and Dr Luca Bertucelli for all the discussions during my internship at UTRC.

I thank all my teachers at my pre-secondary and secondary schools, my teachers at IIT Bombay, and my teachers at UCSB for instilling in me a quest for knowledge and an appreciation for details. I particularly thank Prof Shashikanth Suryanarayanan at IIT Bombay who instigated my interest in controls.

My stay at UCSB and earlier at IIT Bombay has been blessed with great friends. I thank you all for all your support over the years. Thank you Shaunak and Dhanashree Bopardikar for all the tea sessions and all the wonderful food. Thank you Vibhor Jain for all the trips and all the fun activities. Thank you Gargi Chaudhuri for the wonderful radio jockeying time at KCSB. Thank you Sumit Singh and Sriram Venkateswaran for the Friday wine sessions. Thank you Anahita Mirtabatabei for all the activities you organized. Thank you G Kartik for the fun time at IIT Bombay.

No word of thanks would be enough for the unconditional love and the ardent support of my family. I thank my parents for the values they cultivated in me. Each day, I more understand the sacrifices, they made while bringing me up. My brother and my sisters have been a great guidance and support for me through the years. I thank them for standing by me in tough times. This achievement would be impossible without the efforts of my uncle, Mr Marut Kumar Srivastava, who inculcated in me an appreciation for mathematics at an early age. He has been a tremendous support and an inspiration in all my academic endeavors. I thank him for being there whenever I needed him. My family has been a guiding light for me and continues to inspire me in my journey ahead. I endlessly thank my family for all they have done for me.

Curriculum Vitæ

Vaibhav Srivastava

Education

M.A. in Statistics, UCSB, USA. (2012)
Specialization: Mathematical Statistics

M.S. in Mechanical Engineering, UCSB, USA. (2011)
Specialization: Dynamics, Control, and Robotics

B.Tech. in Mechanical Engineering, IIT Bombay, India. (2007)

Research Interests

Stochastic analysis and design of risk averse algorithms for uncertain systems, including (i) mixed human-robot networks, (ii) robotic and mobile sensor networks; and (iii) computational networks.

Professional Experience

- **Graduate Student Researcher** (April 2008 – present)
  Department of Mechanical Engineering, UCSB
- **Summer Intern** (August – September 2011)
  United Technologies Research Center, East Hartford, USA
- **Design Consultant** (May – July 2006)
  General Electrical HealthCare, Bangalore, India

Outreach

- **Graduate Student Mentor** (July – August 2009)
  SABRE, Institute of Collaborative Biosciences, UCSB
- **Senior Undergraduate Mentor** (July 2006 – April 2007)
  Department of Mechanical Engineering, IIT Bombay

Teaching Experience

- **Teaching Assistant, Control System Design** (Fall 2007, Spring 2010)
  Department of Mechanical Engineering, UCSB
- **Teaching Assistant, Vibrations** (Winter 2008)
  Department of Mechanical Engineering, UCSB
- **Teaching Assistant, Engineering Mechanics: Dynamics** (Spring 2011)
  Department of Mechanical Engineering, UCSB
- **Guest Lecture: Distributed Systems and Control** (Spring 2012)
  Department of Mechanical Engineering, UCSB

Undergraduate Supervision

- Christopher J. Ho (Undergraduate student, ME UCSB)
- Lemnyuy 'Bernard' Nyuykongi (Summer intern, ICB UCSB)

Publications

Journal articles

[1]. **V. Srivastava** and F. Bullo. Knapsack problems with sigmoid utility: Approximation algorithms via hybrid optimization. *European Journal of Operational Research*, October 2012. Sumitted.

[2]. L. Carlone, **V. Srivastava**, F. Bullo, and G. C. Calafiore. Distributed random convex programming via constraints consensus. *SIAM Journal on Control and Optimization*, July 2012. Submitted.

[3]. **V. Srivastava**, F. Pasqualetti, and F. Bullo. Stochastic surveillance strategies for spatial quickest detection. *International Journal of Robotics Research*, October 2012. to appear.

[4]. **V. Srivastava**, R. Carli, C. Langbort, and F. Bullo. Attention allocation for decision making queues. *Automatica*, February 2012. Submitted.

[5]. **V. Srivastava**, J. Moehlis, and F. Bullo. On bifurcations in nonlinear consensus networks. *Journal of Nonlinear Science*, 21(6):875–895, 2011.

[6]. **V. Srivastava**, K. Plarre, and F. Bullo. Randomized sensor selection in sequential hypothesis testing. *IEEE Transactions on Signal Processing*, 59(5):2342–2354, 2011.

Refereed conference proceedings

[1]. **V. Srivastava**, A. Surana, and F. Bullo. Adaptive attention allocation in human-robot systems. In *American Control Conference*, pages 2767–2774, Montréal, Canada, June 2012.

[2]. L. Carlone, **V. Srivastava**, F. Bullo, and G. C. Calafiore. A distributed algorithm for random convex programming. In *Int. Conf. on Network Games, Control and Optimization (NetGCooP)*, pages 1–7, Paris, France, October 2011.

[3]. **V. Srivastava** and F. Bullo. Stochastic surveillance strategies for spatial quickest detection. In *IEEE Conf. on Decision and Control and European Control Conference*, pages 83–88, Orlando, FL, USA, December 2011.

[4]. **V. Srivastava** and F. Bullo. Hybrid combinatorial optimization: Sample problems and algorithms. In *IEEE Conf. on Decision and Control and European Control Conference*, pages 7212–7217, Orlando, FL, USA, December 2011.

[5]. **V. Srivastava**, K. Plarre, and F. Bullo. Adaptive sensor selection in sequential hypothesis testing. In *IEEE Conf. on Decision and Control and European Control Conference*, pages 6284–6289, Orlando, FL, USA, December 2011.

[6]. **V. Srivastava**, R. Carli, C. Langbort, and F. Bullo. Task release control for decision making queues. In *American Control Conference*, pages 1855–1860, San Francisco, CA, USA, June 2011.

[7]. **V. Srivastava**, J. Moehlis, and F. Bullo. On bifurcations in nonlinear consensus networks. *Journal of Nonlinear Science*, 21(6):875–895, 2011.

Thesis

[1]. **V. Srivastava**. Topics in probabilistic graphical models. Master's thesis, Department of Statistics, University of California at Santa Barbara, September 2012.

Services

Organizer/Co-Organizer

2011 Santa Barbara Control Workshop.

Reviewer

Proceedings of IEEE, IEEE Transactions on Automatic Controls, IEEE Transactions on Information Theory, Automatica, International Journal of Robotics Research, International Journal of Robust and Nonlinear Control, European Journal of Control, IEEE Conference on Decision and Control (CDC), American Controls Conference (ACC), IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS).

Chair/Co-Chair

Session on Fault Detection, *IEEE Conf. on Decision and Control and European Control Conference,* Orlando, FL, USA, December 2011.

Professional Membership

- Student Member, Institute of Electrical and Electronics Engineers (IEEE)
- Student Member, Society of Industrial and Applied Mathematics (SIAM)

Abstract

Stochastic Search and Surveillance Strategies for Mixed Human-Robot Teams

Vaibhav Srivastava

Mixed human-robot teams are becoming increasingly important in complex and information rich systems. The purpose of the mixed teams is to exploit the human cognitive abilities in complex missions. It has been evident that the information overload in these complex missions has a detrimental effect on the human performance. The focus of this dissertation is the design of efficient human-robot teams. It is imperative for an efficient human-robot team to handle information overload and to this end, we propose a two-pronged strategy: (i) for the robots, we propose strategies for efficient information aggregation; and (ii) for the operator, we propose strategies for efficient information processing. The proposed strategies rely on team objective as well as cognitive performance of the human operator.

In the context of information aggregation, we consider two particular missions. First, we consider information aggregation for a multiple alternative decision making task and pose it as a sensor selection problem in sequential multiple hypothesis testing. We design efficient information aggregation policies that enable the human operator to decide in minimum time. Second, we consider a surveillance

problem and design efficient information aggregation policies that enable the human operator detect a change in the environment in minimum time. We study the surveillance problem in a decision-theoretic framework and rely on statistical quickest change detection algorithms to achieve a guaranteed surveillance performance.

In the context of information processing, we consider two particular scenarios. First, we consider the time-constrained human operator and study optimal resource allocation problems for the operator. We pose these resource allocation problems as knapsack problems with sigmoid utility and develop constant factor algorithms for them. Second, we consider the human operator serving a queue of decision making tasks and determine optimal information processing policies. We pose this problem in a Markov decision process framework and determine approximate solution using certainty-equivalent receding horizon framework.

Contents

Acknowledgements **v**

Abstract **xii**

1 Introduction **1**
1.1 Literature Synopsis 5
1.1.1 Sensor Selection 5
1.1.2 Search and Surveillance 6
1.1.3 Time-constrained Attention Allocation 9
1.1.4 Attention Allocation in Decision Making Queues 10
1.2 Contributions and Organization 13

2 Preliminaries on Decision Making **21**
2.1 Markov Decision Process 21
2.1.1 Existence on an optimal policy 22
2.1.2 Certainty-Equivalent Receding Horizon Control 23
2.1.3 Discretization of the Action and the State Space 24
2.2 Sequential Statistical Decision Making 26
2.2.1 Sequential Binary Hypothesis Testing 27
2.2.2 Sequential Multiple hypothesis Testing 29
2.2.3 Quickest Change Detection 32
2.3 Speed Accuracy Trade-off in Human Decision Making 34
2.3.1 Pew's Model 35
2.3.2 Drift-Diffusion Model 36

3 Randomized Sensor Selection in Sequential Hypothesis Testing **39**
3.1 MSPRT with randomized sensor selection 42
3.2 Optimal sensor selection 52

3.2.1 Optimization of conditioned decision time 54
3.2.2 Optimization of the worst case decision time 56
3.2.3 Optimization of the average decision time 62
3.3 Numerical Illustrations 70
3.4 Conclusions 74

4 Stochastic Surveillance Strategies for Spatial Quickest Detection 76
4.1 Spatial Quickest Detection 83
4.1.1 Ensemble CUSUM algorithm 83
4.2 Randomized Ensemble CUSUM Algorithm 87
4.2.1 Analysis for single vehicle 87
4.2.2 Design for single vehicle 89
4.2.3 Analysis for multiple vehicles 95
4.2.4 Design for multiple vehicles 97
4.3 Adaptive ensemble CUSUM Algorithm 100
4.4 Numerical Results 105
4.5 Experimental Results 115
4.6 Conclusions and Future Directions 122
4.7 Appendix: Probabilistic guarantee to the uniqueness of critical point 124

5 Operator Attention Allocation via Knapsack Problems 128
5.1 Sigmoid Function and Linear Penalty 130
5.2 Knapsack Problem with Sigmoid Utility 132
5.2.1 KP with Sigmoid Utility: Problem Description 132
5.2.2 KP with Sigmoid Utility: Approximation Algorithm 133
5.3 Generalized Assignment Problem with Sigmoid Utility 146
5.3.1 GAP with Sigmoid Utility: Problem Description 146
5.3.2 GAP with Sigmoid Utility: Approximation Algorithm . . . 147
5.4 Bin-packing Problem with Sigmoid Utility 152
5.4.1 BPP with Sigmoid Utility: Problem Description 152
5.4.2 BPP with Sigmoid Utility: Approximation Algorithm . . . 155
5.5 Conclusions and Future Directions 158

6 Attention Allocation in Decision Making Queues 161
6.1 Static queue with latency penalty 163
6.1.1 Problem description 163
6.1.2 Optimal solution 164
6.1.3 Numerical Illustrations 165
6.2 Dynamic queue with latency penalty 166
6.2.1 Problem description 166

6.2.2 Properties of optimal solution 169
6.3 Receding Horizon Solution to dynamic queue with latency penalty 171
6.3.1 Certainty-equivalent finite horizon optimization 172
6.3.2 Performance of receding horizon algorithm 175
6.3.3 Numerical Illustrations 178
6.4 Conclusions 180
6.5 Appendix: Finite horizon optimization for identical tasks 182

Bibliography **189**

Chapter 1

Introduction

The emergence of mobile and fixed sensor networks operating at different modalities, mobility, and coverage has enabled access to an unprecedented amount of data. In a variety of complex and information rich systems, this information is processed by a human operator [17, 31]. The inherent inability of the human operator to handle the plethora of available information has detrimental effects on their performance and may lead to unpleasant consequences [84]. To alleviate this loss in performance of the human operator, the recent national robotic initiative [38] emphasizes collaboration of humans with robotic partners, and envisions a symbiotic co-robot that facilitates efficient interaction of the human operator with the automaton. Given the complex interaction that can arise between the operator and the automaton, such a co-robotic partner will enable better interaction between the automaton and the operator by exploiting the operator's strengths while taking into account their inefficiencies, such as erroneous decisions, fatigue

and the loss of situational awareness. This encourages an investigation into algorithms that enable the co-robot to aid the human partner to focus their attention to the pertinent information and direct the automaton to efficiently collect the information.

The problems of interest in this dissertation are the search and the persistent surveillance. The search problem involves ascertaining the true state of the nature among several possible states of the nature. The objective of the search problem is to ascertain the true state of the nature in minimum time. The persistent surveillance involves continuous search of target regions with a team of fixed and mobile sensors. An efficient persistent surveillance policy has multiple objectives, including, minimizing the time between subsequent visit to a region, minimizing the detection delay at each region, and maximize visits to regions with high likelihood of target. The fundamental trade-off in persistent surveillance is between the evidence collected from the visited region and the resulting delay in evidence collection from other regions. The search and the persistent surveillance missions may be fully autonomous or may involve a human operator that processes the collected evidence and accordingly modifies the search and surveillance policy, respectively. The latter is called the mixed human-robot team search/surveillance. The purpose of the mixed teams is to exploit human cognitive abilities in complex missions, and therefore, an effective model of human cognitive performance

is fundamental to the team design. Moreover, such a model should be efficiently integrated with the automaton design enabling an effective team performance.

As a consequence of the growing interest in the mixed teams, a significant effort has been made to model human cognitive performance and integrate it with the automaton. Broadly speaking, there have been two approaches to mixed team design. In the first approach, the human is allowed to respond freely and the automaton is adaptively controlled to cater to the human operator's cognitive requirements. In the second approach, both the human and the automaton are controlled, for instance, the human operator is told the time-duration they should spend on each task, and their decision is utilized to adaptively control the automaton. The first approach typically captures human performance via their free-response reaction times on each task. The fundamental research questions in this approach include (i) optimal scheduling of automaton [65, 12, 11, 80, 81, 29, 66, 67]; (ii) controlling operator utilization to enable shorter reaction times [78, 79]; and (iii) design of efficient work-shift design to counter fatigue effects on the operator [73]. The second approach captures the human performance as the probability of making the correct decision given the time spent on the task. The fundamental research questions in this approach include (i) optimal duration allocation to each task [93, 94, 90]; (ii) controlling operator utilization to enable

better performance [98]; and (iii) controlling the automaton to collect relevant information [95, 97, 96, 98].

In this dissertation, we focus on the latter approach, although most of the concepts can be easily extended to the former approach. The objective of this dissertation is to illustrate the use of systems theory to design mixed human-robot teams. We illustrate the design principles for the mixed teams with a particular application to mixed-team surveillance. One of the major challenges in mixed human-robot teams is the information overload. The information overload can be handled/avoided by (i) only selecting pertinent sources of information; (ii) only collecting pertinent information; and (iii) efficiently allocating the operator's attention. To this end, first, we study a sensor selection problem for the human operator. The objective of this problem is to identify pertinent sources that help the operator decide on the true state of the nature in minimum time. Second, we study a persistent surveillance problem and design surveillance strategies that result in collection of the information pertinent to the quickest detection of anomalies. Third, we focus on attention allocation problems for the human operator. We study attention allocation for a time-constraint operator as well as attention allocation for an operator serving a queue of decision making tasks.

1.1 Literature Synopsis

In this section, we review the literature in areas relevant to this dissertation. We organize the literature according to the broad topics of interest in this dissertation.

1.1.1 Sensor Selection

One of the key focuses in this dissertation is the sensor selection problem in sequential hypothesis testing. Recent years have witnessed a significant interest in the problem of sensor selection for optimal detection and estimation. Tay *et al.* [99] discuss the problem of censoring sensors for decentralized binary detection. They assess the quality of sensor data by the Neyman-Pearson and a Bayesian binary hypothesis test and decide on which sensors should transmit their observation at that time instant. Gupta *et al.* [39] focus on stochastic sensor selection and minimize the error covariance of a process estimation problem. Isler *et al.* [49] propose geometric sensor selection schemes for error minimization in target detection. Debouk *et al.* [30] formulate a Markovian decision problem to ascertain some property in a dynamical system, and choose sensors to minimize the associated cost. Williams *et al.* [107] use an approximate dynamic program over a rolling time horizon to pick a sensor-set that optimizes the information-communication

trade-off. Wang *et al.* [104] design entropy-based sensor selection algorithms for target localization. Joshi *et al.* [50] present a convex optimization-based heuristic to select multiple sensors for optimal parameter estimation. Bajović *et al.* [5] discuss sensor selection problems for Neyman-Pearson binary hypothesis testing in wireless sensor networks. Katewa *et al.* [53] study sequential binary hypothesis testing using multiple sensors. For a stationary sensor selection policy, they determine the optimal sequential hypothesis test. Bai *et al.* [4] study an off-line randomized sensor selection strategy for sequential binary hypothesis testing problem constrained with sensor measurement costs. In contrast to above works, we focus on sensor selection in sequential multiple hypothesis testing.

1.1.2 Search and Surveillance

Another key focus in this dissertation is efficient vehicle routing policies for search and surveillance. This problem belongs to the broad class of routing for information aggregation which has recently attracted significant attention. Klein *et al.* [56] present a vehicle routing policy for optimal localization of an acoustic source. They consider a set of spatially distributed sensors and optimize the trade-off between the travel time required to collect a sensor observation and the information contained in the observation. They characterize the information in an observation by the volume of the Cramer-Rao ellipsoid associated with the covari-

ance of an optimal estimator. Hollinger *et al.* [46] study routing for an AUV to collect data from an underwater sensor network. They developed approximation algorithms for variants of the traveling salesperson problem to determine efficient policies that maximize the information collected while minimizing the travel time. Zhang *et al.* [108] study the estimation of environmental plumes with mobile sensors. They minimize the uncertainty of the estimate of the ensemble Kalman filter to determine optimal trajectories for a swarm of mobile sensors. We focus on decision theoretic surveillance which has also received some interest in the literature. Castañon [22] poses the search problem as a dynamic hypothesis test, and determines the optimal routing policy that maximizes the probability of detection of a target. Chung *et al.* [26] study the probabilistic search problem in a decision theoretic framework. They minimize the search decision time in a Bayesian setting. Hollinger *et al.* [45] study an active classification problem in which an autonomous vehicle classifies an object based on multiple views. They formulate the problem in an active Bayesian learning framework and apply it to underwater detection. In contrast to the aforementioned works that focus on classification or search problems, our focus is on the quickest detection of anomalies.

The problem of surveillance has received considerable attention recently. Preliminary results on this topic have been presented in [25, 33, 55]. Pasqualetti *et al.* [70] study the problem of optimal cooperative surveillance with multiple agents.

They optimize the time gap between any two visits to the same region, and the time necessary to inform every agent about an event occurred in the environment. Smith *et al.* [86] consider the surveillance of multiple regions with changing features and determine policies that minimize the maximum change in features between the observations. A persistent monitoring task where vehicles move on a given closed path has been considered in [87, 69], and a speed controller has been designed to minimize the time lag between visits of regions. Stochastic surveillance and pursuit-evasion problems have also fetched significant attention. In an earlier work, Hespanha *et al.* [43] studied multi-agent probabilistic pursuit evasion game with the policy that, at each instant, directs pursuers to a location that maximizes the probability of finding an evader at that instant. Grace *et al.* [37] formulate the surveillance problem as a random walk on a hypergraph and parametrically vary the local transition probabilities over time in order to achieve an accelerated convergence to a desired steady state distribution. Sak *et al.* [77] present partitioning and routing strategies for surveillance of regions for different intruder models. Srivastava *et al.* [89] present a stochastic surveillance problem in centralized and decentralized frameworks. They use Markov chain Monte Carlo method and message passing based auction algorithm to achieve the desired surveillance criterion. They also show that the deterministic strategies fail to satisfy the surveillance criterion under general conditions. We focus on

stochastic surveillance policies. In contrast to aforementioned works on stochastic surveillance that assume a surveillance criterion is known, this work concerns the design of the surveillance criterion.

1.1.3 Time-constrained Attention Allocation

In this dissertation, we also focus on attention allocation policies for a time constrained human operator. The resource allocation problems with sigmoid utility well model the situation of a time constrained human operator. To this end, we present versions of the knapsack problem, the bin-packing problem, and the generalized assignment problem in which each item has a sigmoid utility. If the utilities are step functions, then these problems reduce to the standard knapsack problem, the bin-packing problem, and the generalized assignment problem [57, 61], respectively. Similarly, if the utilities are concave functions, then these problems reduce to standard convex resource allocation problems [48]. We will show that with sigmoid utilities the optimization problem becomes a hybrid of discrete and continuous optimization problems.

The knapsack problems [54, 57, 61] have been extensively studied. Considerable emphasis has been on the discrete knapsack problem [57] and the knapsack problems with concave utilities; a survey is presented in [16]. Non-convex knapsack problems have also received significant attention. Kameshwaran *et al.* [52]

study knapsack problems with piecewise linear utilities. Moré *et al.* [63] and Burke *et al.* [18] study knapsack problem with convex utilities. In an early work, Ginsberg [36] studies a knapsack problem in which each item has identical sigmoid utility. Freeland *et al.* [34] discuss the implication of sigmoid functions on decision models and present an approximation algorithm for the knapsack problem with sigmoid utilities that constructs a concave envelope of the sigmoid functions and thus solves the resulting convex problem. In a recent work, Ağrali *et al.* [1] consider the knapsack problem with sigmoid utility and show that this problem is NP-hard. They relax the problem by constructing a concave envelope of the sigmoid function and then determine the global optimal solution using branch and bound techniques. They also develop an FPTAS for the case in which the decision variable is discrete. In contrast to above works, we determine constant factor algorithms for the knapsack problem, the bin-packing problem, and the generalized assignment problem with sigmoid utility.

1.1.4 Attention Allocation in Decision Making Queues

The last focus of this dissertation is on attention allocation policies for a human operator serving a queue of decision making tasks (*decision making queues*). There has been a significant interest in the study of the performance of a human operator serving a queue. In an early work, Schmidt [82] models the human as

a server and numerically studies a queueing model to determine the performance of a human air traffic controller. Recently, Savla *et al.* [81] study human supervisory control for unmanned aerial vehicle operations: they model the system by a simple queuing network with two components in series, the first of which is a spatial queue with vehicles as servers and the second is a conventional queue with human operators as servers. They design joint motion coordination and operator scheduling policies that minimize the expected time needed to classify a target after its appearance. The performance of the human operator based on their utilization history has been incorporated to design maximally stabilizing task release policies for a human-in-the-loop queue in [79, 78]. Bertuccelli *et al.* [12] study the human supervisory control as a queue with re-look tasks. They study the policies in which the operator can put the tasks in an orbiting queue for a re-look later. An optimal scheduling problem in the human supervisory control is studied in [11]. Crandall *et al.* [29] study optimal scheduling policy for the operator and discuss if the operator or the automation should be ultimately responsible for selecting the task. Powel *et al.* [73] model mixed team of humans and robots as a multi-server queue and incorporate a human fatigue model to determine the performance of the team. They present a comparative study of the fixed and the rolling work-shifts of the operators. In contrast to the aforementioned works in queues with human operator, we do not assume that the tasks require a fixed (potentially stochastic)

processing time. We consider that each task may be processed for any amount of time, and the performance on the task is known as a function of processing time.

The optimal control of queueing systems [83] is a classical problem in queueing theory. There has been significant interest in the dynamic control of queues; e.g., see [51] and references therein. In particular, Stidham *et al.* [51] study the optimal servicing policies for an M/G/1 queue of identical tasks. They formulate a semi-Markov decision process, and describe the qualitative features of the solution under certain technical assumptions. In the context of M/M/1 queues, George *et al.* [35] and Adusumilli *et al.* [2] relax some of technical assumptions in [51]. Hernández-Lerma *et al.* [42] determine optimal servicing policies for the identical tasks and some arrival rate. They adapt the optimal policy as the arrival rate is learned. The main differences between these works and the problem considered in this dissertation are: (i) we consider a deterministic service process, and this yields an optimality equation quite different from the optimality equation obtained for Markovian service process; (ii) we consider heterogeneous tasks while the aforementioned works consider identical tasks.

1.2 Contributions and Organization

In this section, we outline the organization of the chapters in this dissertation and detail the contributions in each chapter.

Chapter 2: In this chapter, we review some decision making concepts. We start with a review of Markov decision processes (Section 2.1). We then review sequential statistical decision making (Section 2.2). We close the chapter with a review of human decision making (Section 2.3).

Chapter 3: In this chapter, we analyze the problem of time-optimal sequential decision making in the presence of multiple switching sensors and determine a randomized sensor selection strategy to achieve the same. We consider a sensor network where all sensors are connected to a fusion center. Such topology is found in numerous sensor networks with cameras, sonars or radars, where the fusion center can communicate with any of the sensors at each time instant. The fusion center, at each instant, receives information from only one sensor. Such a situation arises when we have interfering sensors (e.g., sonar sensors), a fusion center with limited attention or information processing capabilities, or sensors with shared communication resources. The sensors may be heterogeneous (e.g., a camera sensor, a sonar sensor, a radar sensor, etc), hence, the time needed to collect, transmit, and process data may differ significantly for these sensors. The fusion

center implements a sequential hypothesis test with the gathered information. The material in this chapter is from [97] and [96].

The major contributions of this work are twofold. First, we develop a version of the MSPRT algorithm in which the sensor is randomly switched at each iteration and characterize its performance. In particular, we determine the asymptotic expressions for the thresholds and the expected sample size for this sequential test. We also incorporate the random processing time of the sensors into these models to determine the expected decision time (Section 3.1). Second, we identify the set of sensors that minimize the expected decision time. We consider three different cost functions, namely, the conditioned decision time, the worst case decision time, and the average decision time. We show that, to minimize the conditioned expected decision time, the optimal sensor selection policy requires only one sensor to be observed. We show that, for a generic set of sensors and M underlying hypotheses, the optimal average decision time policy requires the fusion center to consider at most M sensors. For the binary hypothesis case, we identify the optimal set of sensors in the worst case and the average decision time minimization problems. Moreover, we determine an optimal probability distribution for the sensor selection. In the worst case and the average decision time minimization problems, we encounter the problem of minimization of sum and maximum of

linear-fractional functionals. We treat these problems analytically, and provide insight into their optimal solutions (Section 3.2).

Chapter 4: In this chapter, we study persistent surveillance of an environment comprising of potentially disjoint regions of interest. We consider a team of autonomous vehicles that visit the regions, collect information, and send it to a control center. We study a spatial quickest detection problem with multiple vehicles, that is, the simultaneous quickest detection of anomalies at spatially distributed regions when the observations for anomaly detection are collected by autonomous vehicles. We design vehicle routing policies to collect observations at different regions that result in quickest detection of anomalies at different regions. The material in this chapter is from [95] and [91].

The main contributions of this chapter are fivefold. First, we formulate the stochastic surveillance problem for spatial quickest detection of anomalies. We propose the ensemble CUSUM algorithm for a control center to detect concurrent anomalies at different regions from collected observations (Section 4.1). For the ensemble CUSUM algorithm we characterize lower bounds for the expected detection delay and for the average (expected) detection delay at each region. Our bounds take into account the processing times for collecting observations, the prior probability of anomalies at each region, and the anomaly detection difficulty at each region.

Second, for the case of stationary routing policies, we provide bounds on the expected delay in detection of anomalies at each region (Section 4.2). In particular, we take into account both the processing times for collecting observations and the travel times between regions. For the single vehicle case, we explicitly characterize the expected number of observations necessary to detect an anomaly at a region, and the corresponding expected detection delay. For the multiple vehicles case, we characterize lower bounds for the expected detection delay and the average detection delay at the regions. As a complementary result, we show that the expected detection delay for a single vehicle is, in general, a non-convex function. However, we provide probabilistic guarantees that it admits a unique global minimum.

Third, we design stationary vehicle routing policies to collect observations from different regions (Section 4.2). For the single vehicle case, we design an efficient stationary policy by minimizing an upper bound for the average detection delay at the regions. For the multiple vehicles case, we first partition the regions among the vehicles, and then we let each vehicle survey the assigned regions by using the routing policy as in the single vehicle case. In both cases we characterize the performance of our policies in terms of expected detection delay and average (expected) detection delay.

Fourth, we describe our adaptive ensemble CUSUM algorithm, in which the routing policy is adapted according to the learned likelihood of anomalies in the regions (Section 4.3). We derive an analytic bound for the performance of our adaptive policy. Finally, our numerical results show that our adaptive policy outperforms the stationary counterpart.

Fifth and finally, we report the results of extensive numerical simulations and a persistent surveillance experiment (Sections 4.4 and 4.5). Besides confirming our theoretical findings, these practical results show that our algorithms are robust against realistic noise models, and sensors and motion uncertainties.

Chapter 5: In this chapter, we study optimization problems with sigmoid functions. We show that sigmoid utility renders a combinatorial element to the problem and resource allocated to each item under optimal policy is either zero or more than a critical value. Thus, the optimization variable has both continuous and discrete features. We exploit this interpretation of the optimization variable and merge algorithms from continuous and discrete optimization to develop efficient hybrid algorithms. We study versions of the knapsack problem, the generalized assignment problem and the bin-packing problem in which the utility is a sigmoid function. These problems model situations where human operators are looking at the feeds from a camera network and deciding on the presence of some malicious activity. The first problem determines the optimal fraction of work-hours

an operator should allocate to each feed such that their overall performance is optimal. The second problem determines the allocations of the tasks to identical and independently working operators as well as the optimal fraction of work-hours each operator should allocate to each feed such that the overall performance of the team is optimal. Assuming that the operators work in an optimal fashion, the third problem determines the minimum number of operators and an allocation of each feed to some operator such that each operator allocates non-zero fraction of work-hours to each feed assigned to them. The material in this chapter is from [90] and [92].

The major contributions of this chapter are fourfold. First, we investigate the root-cause of combinatorial effects in optimization problems with sigmoid utility (Section 5.1). We show that for a sigmoid function subject to a linear penalty, the optimal allocation jumps down to zero with increasing penalty rate. This jump in the optimal allocation imparts a combinatorial effect to the problems involving sigmoid functions.

Second, we study the knapsack problem with sigmoid utility (Section 5.2). We exploit the above combinatorial interpretation of the sigmoid functions and utilize a combination of approximation algorithms for the binary knapsack problems and algorithms for continuous univariate optimization to determine a constant factor approximation algorithm for the knapsack problem with sigmoid utility.

Third, we study the generalized assignment problem with sigmoid utility (Section 5.3). We first show that the generalized assignment problem with sigmoid utility is NP-hard. We then exploit a knapsack problem based algorithm for binary generalized assignment problem to develop an equivalent algorithm for generalized assignment problem with sigmoid utility.

Fourth and finally, we study bin-packing problem with sigmoid utility (Section 5.4). We first show that the bin-packing problem with sigmoid utility is NP-hard. We then utilize the solution of the knapsack problem with sigmoid utility to develop a next-fit type algorithm for the bin-packing problem with sigmoid utility.

Chapter 6: In this chapter, we study the problem of optimal time-duration allocation in a queue of binary decision making tasks with a human operator. We refer to such queues as *decision making queues*. We assume that tasks come with processing deadlines and incorporate these deadlines as a soft constraint, namely, latency penalty (penalty due to delay in processing of a task). We consider two particular problems. First, we consider a static queue with latency penalty. Here, the human operator has to serve a given number of tasks. The operator incurs a penalty due to the delay in processing of each task. This penalty can be thought of as the loss in value of the task over time. Second, we consider a dynamic queue of the decision making tasks. The tasks arrive according to a stochastic process and

the operator incurs a penalty for the delay in processing each task. In both the problems, there is a trade-off between the reward obtained by processing a task and the penalty incurred due to the resulting delay in processing other tasks. We address this particular trade-off. The material in this chapter is from [93] and [94].

The major contributions of this chapter are fourfold. First, we determine the optimal duration allocation policy for the static decision making queue with latency penalty (Section 6.1). We show that the optimal policy may not process all the tasks in the queue and may drop a few tasks.

Second, we pose a Markov decision process (MDP) to determine the optimal allocations for the dynamic decision making queue (Section 6.2). We then establish some properties of this MDP. In particular, we show an optimal policy exists and it drops task if queue length is greater than a critical value.

Third, we employ certainty-equivalent receding horizon optimization to approximately solve this MDP (Section 6.3). We establish performance bounds on the certainty-equivalent receding horizon solution.

Fourth and finally, we suggest guidelines for the design of decision making queues (Section 6.3). These guidelines suggest the maximum expected arrival rate at which the operator expects a new task to arrive soon after optimally processing the current task.

Chapter 2

Preliminaries on Decision Making

In this chapter, we survey decision making in three contexts: (i) optimal decision making in Markov decision processes, (ii) sequential statistical decision making, and (iii) human decision making.

2.1 Markov Decision Process

A Markov decision process (MDP) is a stochastic dynamic system described by five elements: (i) a Borel state space $\mathcal{X}$; (ii) a Borel action space $\mathcal{U}$; (iii) a set of admissible actions defined by a strict, measurable, compact-valued multifunction $U : \mathcal{X} \mapsto \mathbb{B}(\mathcal{U})$, where $\mathbb{B}(\cdot)$ represents the Borel sigma-algebra; (iv) a Borel stochastic transition kernel $\mathbb{P}(X|\cdot) : K \to [0, 1]$, for each $X \in \mathbb{B}(\mathcal{X})$, where $K \in \mathbb{B}(\mathcal{X} \times \mathcal{U})$; and (v) a measurable one stage reward function $g : K \times W \to \mathbb{R}$, where $W \in \mathbb{B}(\mathcal{W})$ and $\mathcal{W}$ is a Borel space of random disturbance.

We focus on discrete time MDPs. The objective of the MDP is to maximize the infinite horizon reward called the value function. There are two formulations for the value function: (i) α-discounted value function; and (ii) average value function. For a discount factor $\alpha \in (0,1)$, the α-discounted value function $J_\alpha : \mathcal{X} \times U_{\text{stat}}(\mathcal{X}, U) \to \mathbb{R}$ is defined by

$$J_\alpha(x, u_{\text{stat}}) = \max_{u_{\text{stat}} \in U_{\text{stat}}} \mathbb{E}\Big[\sum_{\ell=1}^{\infty} \alpha^{\ell-1} g(x_\ell, u_{\text{stat}}(x_\ell), w_\ell)\Big],$$

where $U_{\text{stat}}(\mathcal{X}, U)$ is the space of functions defined from $\mathcal{X}$ to U, u_{stat} is the stationary policy, $x_1 = x$, and the expectation is over the uncertainty w_ℓ that may only depend on x_ℓ and $u_{\text{stat}}(x_\ell)$. Note that a stationary policy is a function that allocates a unique action to each state.

The average value function $J_{\text{avg}} : \mathcal{X} \times U_{\text{stat}}(\mathcal{X}, U) \to \mathbb{R}$ is defined by

$$J_{\text{avg}}(x, u_{\text{stat}}) = \max_{u_{\text{stat}} \in U_{\text{stat}}} \lim_{N \to +\infty} \frac{1}{N} \mathbb{E}\Big[\sum_{\ell=1}^{N} g(x_\ell, u_{\text{stat}}(x_\ell), w_\ell)\Big].$$

2.1.1 Existence on an optimal policy

For the MDP with a compact action space, the optimal discounted value function exists if the following conditions hold [3]:

(i). the reward function $\mathbb{E}[g(\cdot, \cdot, w)]$ is bounded above and continuous for each state and action;

(ii). the transition kernel $\mathbb{P}(X|K)$ is weakly continuous in K, i.e., for a continuous and bounded function $h : \mathcal{X} \to \mathbb{R}$, the function $h_{\text{new}} : K \to \mathbb{R}$ defined by

$$h_{\text{new}}(x, u) = \sum_{X \in \mathcal{X}} h(X)\mathbb{P}(X|x, u),$$

is a continuous and bounded function.

(iii). the multifunction U is continuous.

Let $J^*_\alpha : \mathcal{X} \to \mathbb{R}$ be the optimal α-discounted value function. The optimal average value function exists if there exists a finite constant $M \in \mathbb{R}$ such that

$$|J^*_\alpha(x) - J^*_\alpha(\hat{x})| < M,$$

for each $\alpha \in (0, 1)$, for each $x \in \mathcal{X}$, and for some $\hat{x} \in \mathcal{X}$, and the transition kernel $\mathbb{P}$ is continuous in action variable [3].

If the optimal value function for the MDP exists, then it needs to be efficiently computed. The exact computation of the optimal value function is in general intractable and approximation techniques are employed [9]. We now present on some standard techniques to approximately solve the discounted value MDP. The case of average value MDP follows analogously.

2.1.2 Certainty-Equivalent Receding Horizon Control

We now focus on certainty-equivalent receding horizon control [9, 23, 62] to approximately compute the optimal average value function. According to the

certainty-equivalent approximation, the future uncertainties in the MDP are replaced by their nominal values. The receding horizon control approximates the infinite-horizon average value function with a finite horizon average value function. Therefore, under certainty-equivalent receding horizon control, the approximate average value function $\bar{J}_{\text{avg}} : \mathcal{X} \to \mathbb{R}$ is defined by

$$\bar{J}_{\text{avg}}(x) = \max_{u_1,\ldots,u_N} \frac{1}{N} \sum_{\ell=1}^{N} g(\bar{x}_\ell, u_\ell, \bar{w}_\ell),$$

where $\bar{x}_\ell$ is the certainty-equivalent evolution of the system, i.e., the evolution of the system obtained by replacing the uncertainty in the evolution at each stage by its nominal value, $\bar{x}_1 = x$, and $\bar{w}_\ell$ is the nominal value of the uncertainty at stage ℓ. The certainty-equivalent receding horizon control at a given state x computes the approximate average value function $\bar{J}_{\text{avg}}(x)$, implements the first control action u_1, lets the system evolve to a new state x', and repeats the procedure at the new state. There are two salient features the approximate average value function computation: (i) it approximates the value function at a given state by a deterministic dynamic program; and (ii) it utilizes a open loop strategy, i.e., the action variables u_ℓ do not depend on $\bar{x}_\ell$.

2.1.3 Discretization of the Action and the State Space

Note that the certainty-equivalent evolution $\bar{x}_\ell$ may not belong to $\mathcal{X}$, for instance, $\mathcal{X}$ may be the set of natural numbers and $\bar{x}_\ell$ may be a positive real

number. In general, certainty-equivalent state $\bar{x}_\ell$ will belong to a compact and uncountable set and therefore, the finite-horizon deterministic dynamic program associated with the computation of the approximate average value function involves compact and uncountable action and state space. A popular technique to approximately solve such dynamic programs involve discretization of action and state space followed by the standard backward induction algorithm. We now focus on efficiency of such discretization. Let the certainty-equivalent evolution of the state be represented by $\bar{x}_{\ell+1} = \texttt{evol}(\bar{x}_\ell, u_\ell, \bar{w}_\ell)$, where $\texttt{evol} : \bar{\mathcal{X}}_\ell \times \mathcal{U}_\ell \times \mathcal{W}_\ell \to \bar{\mathcal{X}}_{\ell+1}$ represents the certainty-equivalent evolution function, $\bar{\mathcal{X}}_\ell$ represents the compact space to which the certainty-equivalent state at stage ℓ belongs, $\mathcal{U}_\ell$ is the compact action space at stage ℓ and $\mathcal{W}_\ell$ is the disturbance space at stage ℓ. Let the action space and the state space be discretized (see [10] for details of discretization procedure) such that Δx and Δu are the maximum grid diameter in the discretized state and action space, respectively. Let $\hat{J}^*_{\text{avg}} : \mathcal{X} \to \mathbb{R}$ be the approximate average value function obtained via discretized state and action space. If the action and state variables are continuous and belong to compact sets, and the reward functions and the state evolution functions are Lipschitz continuous, then

$$|\bar{J}^*_{\text{avg}}(x) - \hat{J}^*_{\text{avg}}(x)| \leq \beta(\Delta x + \Delta u),$$

for each $x \in \mathcal{X}$, where β is some constant independent of the discretization grid [10].

2.2 Sequential Statistical Decision Making

We now present sequential statistical decision making problems. Opposed to the classical statistical procedures in which the number of available observations are fixed, the sequential analysis collects observations sequentially until some stopping criterion is met. We focus on three statistical decision making problems, namely, sequential binary hypothesis testing, sequential multiple hypothesis testing, and quickest change detection. We first define the notion of Kullback-Leibler divergence that will be used throughout this section.

Definition 2.1 (***Kullback-Leibler divergence***, [28])**.** Given two probability mass functions $f_1 : \mathcal{S} \to \mathbb{R}_{\geq 0}$ and $f_2 : \mathcal{S} \to \mathbb{R}_{\geq 0}$, where $\mathcal{S}$ is some countable set, the Kullback-Leibler divergence $\mathcal{D} : \mathcal{L}^1 \times \mathcal{L}^1 \to \mathbb{R} \cup \{+\infty\}$ is defined by

$$\mathcal{D}(f_1, f_2) = \mathbb{E}_{f_1}\left[\log \frac{f_1(X)}{f_2(X)}\right] = \sum_{x \in \text{supp}(f_1)} f_1(x) \log \frac{f_1(x)}{f_2(x)},$$

where $\mathcal{L}^1$ is the set of integrable functions and $\text{supp}(f_1)$ is the support of f_1. It is known that (i) $0 \leq \mathcal{D}(f_1, f_2) \leq +\infty$, (ii) the lower bound is achieved if and only if $f_1 = f_2$, and (iii) the upper bound is achieved if and only if the support of f_2 is a strict subset of the support of f_1. Note that equivalent statements can be given for probability density functions. □

2.2.1 Sequential Binary Hypothesis Testing

Consider two hypotheses H_0 and H_1 with their conditional probability distribution functions $f^0(y) := f(y|H_0)$ and $f^1(y) := f(y|H_1)$. Let $\pi_0 \in (0,1)$ and $\pi_1 \in (0,1)$ be the associated prior probabilities. The objective of the sequential hypothesis testing is to determine a rule to stop collecting further observations and make a decision on the true hypothesis. Suppose that the cost of collecting an observation be $c \in \mathbb{R}_{\geq 0}$ and penalties $L_0 \in \mathbb{R}_{\geq 0}$ and $L_1 \in \mathbb{R}_{\geq 0}$ are imposed for a wrong decision on hypothesis H_0 and H_1, respectively. The optimal stopping rule determined through an MDP. The state of this MDP is the probability of hypothesis H_0 being true and the action space comprises of three actions, namely, declare H_0, declare H_1, and continue sampling. Let the state at stage $\tau \in \mathbb{N}$ of the MDP be p_τ and an observation Y_τ be collected at stage τ. The value function associated with sequential binary hypothesis testing at its τ-th stage is

$$J_\tau(p_\tau) = \min\{pL_1, (1-p)L_0, c + \mathbb{E}_{f^*}[J_{\tau+1}(p_{\tau+1}(Y_\tau))]\},$$

where the argument of min operator comprises of three components associated with three possible decisions, $f^* = p_\tau f^0 + (1-p_\tau)f^1$, $\mathbb{E}_{f^*}$ denote expected value with respect to the measure f^*, $p_1 = \pi_0$, and $p_{\tau+1}(Y_\tau)$ is the posterior probability defined by

$$p_{\tau+1}(Y_\tau) = \frac{p_\tau f^0(Y_\tau)}{p_\tau f^0(Y_\tau) + (1-p_\tau)f^1(Y_\tau)}.$$

Algorithm 2.1: *Sequential Probability Ratio Test*

Input : threshold η_0, η_1, pdfs f^0, f^1 ;

Output : decision on true hypothesis ;

1 at time $\tau \in \mathbb{N}$, collect sample y_τ;

2 compute the log likelihood ratio $\lambda_\tau := \log \frac{f^1(y_\tau)}{f^0(y_\tau)}$

3 integrate evidence up to current time $\Lambda_\tau := \sum_{t=1}^{\tau} \lambda_t$

% decide only if a threshold is crossed

4 **if** $\Lambda_\tau > \eta_1$ **then** accept H_1;

5 **else if** $\Lambda_\tau < \eta_0$ **then** accept H_0;

6 **else** continue sampling (step **1**)

The solution to the above MDP is the famous *sequential probability ratio test*(SPRT), first proposed in [103]. The SPRT algorithm collects evidence about the hypotheses and compares the integrated evidence to two thresholds η_0 and η_1 in order to decide on a hypothesis. The SPRT procedure is presented in Algorithm 2.1.

Given the probability of missed detection $\mathbb{P}(H_0|H_1) = \alpha_0$ and probability of false alarm $\mathbb{P}(H_1|H_0) = \alpha_1$, the Wald's thresholds η_0 and η_1 are defined by

$$\eta_0 = \log \frac{\alpha_0}{1-\alpha_1}, \quad \text{and} \quad \eta_1 = \log \frac{1-\alpha_0}{\alpha_1}. \tag{2.1}$$

Let N_d denote the number of samples required for decision using SPRT. Its expected value is approximately [85] given by

$$\begin{aligned} \mathbb{E}[N_d|H_0] &\approx -\frac{(1-\alpha_1)\eta_0 + \alpha_1\eta_1}{\mathcal{D}(f^0, f^1)}, \quad \text{and} \\ \mathbb{E}[N_d|H_1] &\approx \frac{\alpha_0\eta_0 + (1-\alpha_0)\eta_1}{\mathcal{D}(f^1, f^0)}. \end{aligned} \tag{2.2}$$

The approximations in equation (2.2) are referred to as the Wald's approximations [85]. According to the Wald's approximation, the value of the aggregated SPRT statistic is exactly equal to one of the thresholds at the time of decision. It is known that the Wald's approximation is accurate for large thresholds and small error probabilities.

2.2.2 Sequential Multiple hypothesis Testing

Consider M hypotheses $H_k \in \{0, \dots, M-1\}$ with their conditional probability density functions $f^k(y) := f(y|H_k)$. The sequential multiple hypothesis testing problem is posed similar to the binary case. Suppose the cost of collecting an observation be $c \in \mathbb{R}_{\geq 0}$ and a penalty $L_k \in \mathbb{R}_{\geq 0}$ is incurred for wrongly accepting hypothesis H_k. The optimal stopping rule determined through an MDP. The state of this MDP is the probability of each hypothesis being true and the action space comprises of $M+1$ actions, namely, declare $H_k, k \in \{0, \dots, M-1\}$, and continue sampling. Let the state at stage $\tau \in \mathbb{N}$ of the MDP be $\boldsymbol{p}_\tau = (p_\tau^0, \dots, p_\tau^{M-1})$ and an observation Y_τ be collected at stage τ. The value function associated with

sequential multiple hypothesis testing at its τ-th iteration is

$$J_\tau(\boldsymbol{p}_\tau) = \min\{(1-p_\tau^0)L_0, \ldots, (1-p_\tau^{M-1})L_{M-1}, c + \mathbb{E}_{f^*}[J_{\tau+1}(\boldsymbol{p}_{\tau+1}(Y_\tau))]\},$$

where the argument of min operator comprises of $M+1$ components associated with $M+1$ possible decisions, $f^* = \sum_{k=0}^{M-1} p_k f^k$, $\mathbb{E}_{f^*}$ denote expected value with respect to the measure f^* and $p_{\tau+1}(Y_\tau)$ is the posterior probability defined by

$$p_{\tau+1}^k(Y_\tau) = \frac{p_\tau^k f^k(Y_\tau)}{\sum_{j=0}^{M-1} p_j f^j(Y_\tau)}, \quad \text{for each } k \in \{0, \ldots, M-1\}. \tag{2.3}$$

Opposed to the binary hypothesis case, there exists no closed form algorithmic solution to the above MDP for $M > 2$. In particular, the optimal stopping rule no longer comprises of constant functions, rather it comprises of curves in state space that are difficult to characterize explicitly. However, in the asymptotic regime, i.e., when the cost of observations $c \to 0^+$ and the penalties for wrong decisions are identical, these curves can be approximated by constant thresholds. In this asymptotic regime, a particular solution to the above MDP is the *multiple hypothesis sequential probability ratio test* (MSPRT) proposed in [7]. Note that the MSPRT reduces to SPRT for $M = 2$. The MSPRT is described in Algorithm 2.2.

The thresholds η_k are designed as functions of the frequentist error probabilities (i.e., the probabilities to accept a given hypothesis wrongly) α_k, $k \in \{0, \ldots, M-1\}$. Specifically, the thresholds are given by

$$\eta_k = \frac{\alpha_k}{\gamma_k}, \tag{2.4}$$

Algorithm 2.2: *Multiple hypothesis sequential probability ratio test*

Input : threshold η_0, η_1, pdfs f^0, f^1 ;

Output : decision on true hypothesis ;

1 at time $\tau \in \mathbb{N}$, collect sample y_τ;

2 compute the posteriors p_τ^k, $k \in \{0, \dots, M-1\}$ as in (2.3)

% decide only if a threshold is crossed

3 **if** $p_\tau^h > \dfrac{1}{1+\eta_h}$ *for at least one* $h \in \{0, \dots, M-1\}$ **then** accept H_1;

4 **else if** $\Lambda_\tau < \eta_0$ **then** accept H_k with maximum p_τ^k;

5 **else** continue sampling (step **1**)

where $\gamma_k \in (0,1)$ is a constant function of f^k (see [7]).

Let $\eta_{\max} = \max\{\eta_j \mid j \in \{0, \dots, M-1\}\}$. It is known [7] that the expected sample size of the MSPRT N_d, conditioned on a hypothesis, satisfies

$$\mathbb{E}[N_d|H_k] \to \frac{-\log \eta_k}{\mathcal{D}^*(k)}, \quad \text{as } \eta_{\max} \to 0^+,$$

where $\mathcal{D}^*(k) = \min\{\mathcal{D}(f^k, f^j) \mid j \in \{0, \dots, M-1\}, j \neq k\}$ is the minimum Kullback-Leibler divergence from the distribution f^k to all other distributions f^j, $j \neq k$.

2.2.3 Quickest Change Detection

Consider a sequence of observations $\{y_1, y_2, \ldots\}$ such that $\{y_1, \ldots, y_{\upsilon-1}\}$ are i.i.d. with probability density function f^0 and $\{y_\upsilon, y_{\upsilon+1}, \ldots\}$ are i.i.d. with probability density function f^1 with υ unknown. The quickest change detection problem concerns detection of change in minimum possible observations (iterations) and in studied in Bayesian and non-Bayesian settings. In the Bayesian formulation, a prior probability distribution of the change time υ is known and is utilized to determine the quickest detection scheme. On the other hand, in the non-Bayesian formulation the worst possible instance of the change time is considered to determine the quickest detection scheme. We focus on the non-Bayesian formulation. Let $\upsilon_{\text{det}} \geq \upsilon$ be the iteration at which the change is detected. The non-Bayesian quickest detection problem (Lorden's problem) [72, 85] is posed as follows:

$$\begin{aligned} \text{minimize} \quad & \sup_{\upsilon \geq 1} \mathbb{E}_\upsilon[\upsilon_{\text{det}} - \upsilon + 1 | \upsilon_{\text{det}} \geq \upsilon] \\ \text{subject to} \quad & \mathbb{E}_{f^0}[\upsilon_{\text{det}}] \geq 1/\gamma, \end{aligned} \tag{2.5}$$

where $\mathbb{E}_\upsilon[\cdot]$ represents the expected value with respect to the distribution of observation at iteration υ, and $\gamma \in \mathbb{R}_{>0}$ is a large constant and is called the *false alarm rate.*

The cumulative sum (CUSUM) algorithm, first proposed in [68], is shown to be the solution to the problem (2.5) in [64]. The CUSUM algorithm is presented in Algorithm 2.3. The CUSUM algorithm can be interpreted as repeated SPRT

Algorithm 2.3: *Cumulative Sum Algorithm*

Input : threshold η, pdfs f^0, f^1 ;

Output : decision on change

1 at time $\tau \in \mathbb{N}$, collect sample y_τ;

2 integrate evidence $\Lambda_\tau := (\Lambda_{\tau-1} + \log \frac{f^1(y_\tau)}{f^0(y_\tau)})^+$

% decide only if a threshold is crossed

3 **if** $\Lambda_\tau > \eta$ **then** declare change is detected;

4 **else** continue sampling (step **1**)

with thresholds 0 and η. For a given threshold η, the expected time between two false alarms and the worst expected number of observations for CUSUM algorithm are

$$\mathbb{E}_{f^0}(N) \approx \frac{e^\eta - \eta - 1}{\mathcal{D}(f^0, f^1)}, \text{ and } \mathbb{E}_{f^1}(N) \approx \frac{e^{-\eta} + \eta - 1}{\mathcal{D}(f^1, f^0)}, \tag{2.6}$$

respectively. The approximations in equation (2.6) are the Wald's approximations, see [85], and are known to be accurate for large values of the threshold η.

2.3 Speed Accuracy Trade-off in Human Decision Making

We now consider information processing by a human operator. The reaction time and error rate are two elements that determine the efficiency of information processing by a human operator. In general, a fast reaction is erroneous, while the returns on error rate are diminishing after some critical time. This trade-off between reaction time (speed) and error rates (accuracy) has been extensively studied in cognitive psychology literature. We review two particular models that capture speed-accuracy trade-off in human decision making, namely, Pew's model and drift-diffusion model. Both these models suggest a sigmoid performance function for the human operator. Before we present these models, we define the sigmoid functions:

Definition 2.2 (***Sigmoid functions***)**.** A Lipschitz-continuous function $f : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ defined by

$$f(t) = f_{\text{cvx}}(t)\mathbf{1}(t < t^{\text{inf}}) + f_{\text{cnv}}(t)\mathbf{1}(t \geq t^{\text{inf}}),$$

where f_{cvx} and f_{cnv} are monotonically non-decreasing convex and concave functions, respectively, $\mathbf{1}(\cdot)$ is the indicator function, and t^{inf} is the inflection point. The sub-derivative of a sigmoid function is unimodal and achieves its maximum

at t^{inf}. Moreover, $\lim_{t\to+\infty} \partial f(t) = 0$, where ∂f represents sub-derivative of the function f. A typical graph of a smooth sigmoid function and its derivative is shown in Figure 2.1.

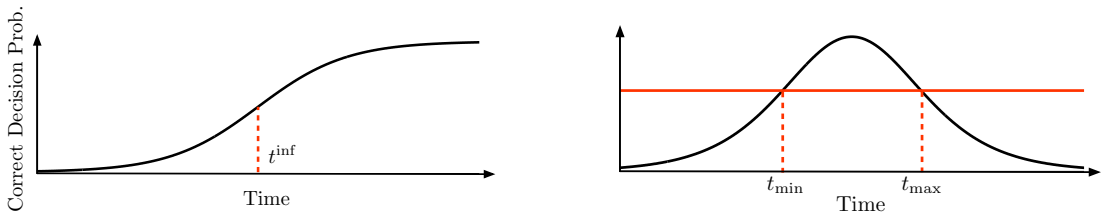


Figure 2.1: A typical sigmoid function and its derivative.

2.3.1 Pew's Model

Pew [71, 106] studied the evolution of the log odds of the probability of correct reaction with reaction time. He demonstrated that the log odds of the correct reaction probability is a linear function. Consequently, for a reaction time $t \in \mathbb{R}_{\geq 0}$,

$$\mathbb{P}(\text{ correct reaction } | t) = \frac{p_0}{1 + e^{-(at-b)}}, \tag{2.7}$$

where $p_0 \in [0, 1]$, $a \in \mathbb{R}_{>0}$ and $b \in \mathbb{R}$ are some parameters specific to the human operator. Note that for a negative b the probability of correct decision in equation (2.7) is a concave function of reaction time t, while for a positive b it is convex for $t < b/a$ and concave otherwise. In both the cases, the probability of correct decision is a sigmoid function of reaction time t.

2.3.2 Drift-Diffusion Model

The drift-diffusion model [13] models the performance of a human operator on a two alternative forced choice (TAFC) task. A TAFC task models a situation in which an operator has to decide among one of the two alternative hypotheses. The TAFC task models rely on three assumptions: (a) evidence is collected over time in favor of each alternative; (b) the evidence collection process is random; and (c) a decision is made when the collected evidence is sufficient to choose one alternative over the other. A TAFC task is well modeled by the drift-diffusion model (DDM) [13]. The DDM captures the evidence aggregation in favor of an alternative by

$$dx(t) = \mu dt + \sigma dW(t), \quad x(0) = x_0, \tag{2.8}$$

where $\mu \in \mathbb{R}$ in the drift rate, $\sigma \in \mathbb{R}_{>0}$ is the diffusion rate, $W(\cdot)$ is the standard Weiner process, and $x_0 \in \mathbb{R}$ is the initial evidence. For an unbiased operator, the initial evidence $x_0 = 0$, while for a biased operator x_0 captures the odds of prior probabilities of alternative hypotheses; in particular, $x_0 = \sigma^2 \log(\pi/(1-\pi))/2\mu$, where π is the prior probability of the first alternative.

For the information aggregation model (2.8), the human decision making is studied in two paradigms, namely, *free response* and *interrogation*. In the free response paradigm, the operator take their own time to decide on an alternative, while in the interrogation paradigm, the operator works under time pressure

and needs to decide within a given time. The free response paradigm is modeled via two thresholds (positive and negative) and the operator decides in favor of first/second alternative if the positive (negative) threshold is crossed from below (above). Under the free response, the DDM is akin to the SPRT and is in fact the continuum limit to the SPRT [13]. The reaction times under the free response paradigm is a random variable with a unimodal probability distribution that is skewed to right. Such unimodal skewed reaction time probability distributions also capture human performance in several other tasks as well [88]. In this paradigm, the operator's performance is well captured by the cumulative distribution function of the reaction time. The cumulative distribution function associated with a unimodal probability distribution function is also a sigmoid function.

The interrogation paradigm is modeled via a single threshold. In particular, for a given deadline $t \in \mathbb{R}_{>0}$, the operator decides in favor of first (second) alternative if the evidence collected by time t, i.e., $x(t)$ is greater (smaller) than a threshold $\nu \in \mathbb{R}$. If the two alternatives are equally likely, then the threshold ν is chosen to be zero. According to equation (2.8), the evidence collected until time t is a Gaussian random variable with mean $\mu t + x_0$ and variance $\sigma^2 t$. Thus, the probability to decide in favor of first alternative is

$$\mathbb{P}(x(t) > \nu) = 1 - \mathbb{P}(x(t) < \nu) = 1 - \Phi\Big(\frac{\nu - \mu t - x_0}{\sigma\sqrt{t}}\Big),$$

where $\Phi(\cdot)$ is the standard normal cumulative distribution function. The probability of making the correct decision at time t is $\mathbb{P}(x(t) > \nu)$ and is a metric that captures the operator's performance. For an unbiased operator $x_0 = 0$, and the performance $\mathbb{P}(x(t) > \nu)$ is a sigmoid function of allocated time t.

In addition to the above decision making performance, the sigmoid function also models the quality of human-machine communication [106], the human performance in multiple target search [47], and the advertising response function [102].

Chapter 3

Randomized Sensor Selection in Sequential Hypothesis Testing

We consider a group of n agents (e.g., robots, sensors, or cameras), which take measurements and transmit them to a fusion center. We generically call these agents "sensors." We identify the fusion center with a person supervising the agents, and call it the "supervisor." The goal of the supervisor is to decide, based on the measurements it receives, which one of M alternative hypotheses or "states of nature" is correct. To do so, the supervisor implements the MSPRT with the collected observations. Given pre-specified accuracy thresholds, the supervisor aims to make a decision in minimum time.

We assume that there are more sensors than hypotheses (i.e., $n > M$), and that only one sensor can transmit to the supervisor at each (discrete) time instant. Equivalently, the supervisor can process data from only one of the n sensors at each time. Thus, at each time, the supervisor must decide which sensor should

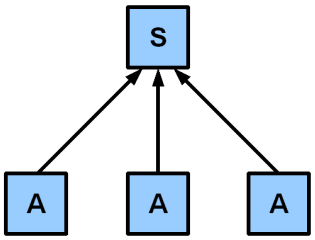


Figure 3.1: The agents A transmit their observation to the supervisor S, one at the time. The supervisor performs a sequential hypothesis test to decide on the underlying hypothesis.

transmit its measurement. This setup also models a sequential search problem, where one out of n sensors is sequentially activated to establish the most likely intruder location out of M possibilities; see [22] for a related problem. In this chapter, our objective is to determine the optimal sensor(s) that the supervisor must observe in order to minimize the decision time.

We adopt the following notation. Let $\{H_0, \dots, H_{M-1}\}$ denote the $M \geq 2$ hypotheses. The time required by sensor $s \in \{1, \dots, n\}$ to collect, process and transmit its measurement is a random variable $T_s \in \mathbb{R}_{>0}$, with finite first and second moment. We denote the mean processing time of sensor s by $\bar{T}_s \in \mathbb{R}_{>0}$. Let $s_t \in \{1, \dots, n\}$ indicate which sensor transmits its measurement at time $t \in \mathbb{N}$. The measurement of sensor s at time t is $y(t, s)$. For the sake of convenience, we denote $y(t, s_t)$ by y_t. For $k \in \{0, \dots, M-1\}$, let $f_s^k : \mathbb{R} \to \mathbb{R}$ denote the probability density function of the measurement y at sensor s conditioned on the hypothesis H_k. Let $f^k : \{1, \dots, n\} \times \mathbb{R} \to \mathbb{R}$ be the probability density function of the pair

(s, y), conditioned on hypothesis H_k. For $k \in \{0, \ldots, M-1\}$, let α_k denote the desired bound on probability of incorrect decision conditioned on hypothesis H_k. We make the following standard assumption:

Conditionally-independent observations: Conditioned on hypothesis H_k, the measurement $y(t, s)$ is independent of $y(\bar{t}, \bar{s})$, for $(t, s) \neq (\bar{t}, \bar{s})$.

We adopt a *randomized strategy* in which the supervisor chooses a sensor randomly at each time instant; the probability to choose sensor s is stationary and given by q_s, for $s \in \{1, \ldots, n\}$. Also, the supervisor uses the data collected from the randomized sensors to execute a multi-hypothesis sequential hypothesis test. For the stationary randomized strategy, note that $f^k(s, y) = q_s f_s^k(y)$. We study our proposed randomized strategy under the following assumptions about the sensors.

Distinct sensors: There are no two sensors with identical conditioned probability density $f_s^k(y)$ and mean processing time $\bar{T}_s$. (If there are such sensors, we club them together in a single node, and distribute the probability assigned to that node equally among them.)

Finitely-informative sensors: Each sensor $s \in \{1, \ldots, n\}$ has the following property: for any two hypotheses $k, j \in \{0, \ldots, M-1\}$, $k \neq j$,

(i). the support of f_s^k is equal to the support of f_s^j,

(ii). $f_s^k \neq f_s^j$ almost surely in f_s^k, and

(iii). conditioned on hypothesis H_k, the first and second moment of $\log(f_s^k(Y)/f_s^j(Y))$ are finite.

Remark 3.1 (***Finitely informative sensors***)**.** The finitely-informative sensors assumption is equivalently restated as follows: each sensor $s \in \{1, \ldots, n\}$ satisfies $0 < \mathcal{D}(f_s^k, f_s^j) < +\infty$ for any two hypotheses $k, j \in \{0, \ldots, M-1\}$, $k \neq j$. □

Remark 3.2 (***Stationary policy***)**.** We study a stationary policy because it is simple to implement, it is amenable to rigorous analysis and it has intuitively-appealing properties (e.g., we show that the optimal stationary policy requires the observation of only as many sensors as the number of hypothesis). On the contrary, if we do not assume a stationary policy, the optimal solution would be based on dynamic programming and, correspondingly, would be complex to implement, analytically intractable, and would lead to only numerical results. □

3.1 MSPRT with randomized sensor selection

We call the MSPRT with the data collected from n sensors while observing only one sensor at a time as the MSPRT with randomized sensor selection. For each sensor s, define $\mathcal{D}_s^*(k) = \min\{\mathcal{D}(f_s^k, f_s^j) \mid j \in \{0, \ldots, M-1\}, j \neq k\}$. The

sensor to be observed at each time is determined through a randomized policy, and the probability of choosing sensor s is stationary and given by q_s. Assume that the sensor $s_t \in \{1, \dots, n\}$ is chosen at time instant t, then the posterior probability after the observations y_t, $t \in \{1, \dots, \tau\}$, is given by

$$\begin{aligned} p_\tau^k = \mathbb{P}(H_k | y_1, \dots, y_\tau) &= \frac{\prod_{t=1}^{\tau} f^k(s_t, y_t)}{\sum_{j=0}^{M-1} \prod_{t=1}^{\tau} f^j(s_t, y_t)} \\ &= \frac{\prod_{t=1}^{\tau} q_{s_t} f_{s_t}^k(y_t)}{\sum_{j=0}^{M-1} \prod_{t=1}^{\tau} q_{s_t} f_{s_t}^j(y_t)} = \frac{\prod_{t=1}^{\tau} f_{s_t}^k(y_t)}{\sum_{j=0}^{M-1} \prod_{t=1}^{\tau} f_{s_t}^j(y_t)}, \end{aligned} \tag{3.1}$$

and, at any given time τ, the hypothesis with maximum posterior probability p_τ^k is the one maximizing $\prod_{t=1}^{\tau} f_{s_t}^k(y_t)$. Note that the sequence $\{(s_t, y_t)\}_{t \in \mathbb{N}}$ is an i.i.d. realization of the pair (s, Y_s), where Y_s is the measurement of sensor s.

For thresholds η_k, $k \in \{0, \dots, M-1\}$, defined in equation (2.4), the MSPRT with randomized sensor selection is defined identically to the Algorithm 2.2, where the first two instructions (steps **1** and **2**) are replaced by:

1 at time $\tau \in \mathbb{N}$, select a random sensor s_τ according to the probability vector q and collect a sample y_τ

2 compute the posteriors p_τ^k, $k \in \{0, \dots, M-1\}$ as in (3.1)

Lemma 3.1 (***Asymptotics***)**.** Assume finitely informative sensors $\{1, \dots, n\}$. Conditioned on hypothesis H_k, $k \in \{0, \dots, M-1\}$, the sample size for decision $N_d \to \infty$ almost surely as $\eta_{\max} \to 0^+$.

Proof.

$$\begin{aligned}
\mathbb{P}(N_d \leq \tau | H_k) &= \mathbb{P}\Big(\min_{a\in\{1,\dots,\tau\}} \sum_{\substack{j=1\\ j\neq v}}^{M-1} \prod_{t=1}^{a} \frac{f^j_{s_t}(y_t)}{f^v_{s_t}(y_t)} < \eta_v, \text{for some } v \in \{0,\dots,M-1\} \Big| H_k \Big) \\
&\leq \mathbb{P}\Big(\min_{a\in\{1,\dots,\tau\}} \prod_{t=1}^{a} \frac{f^j_{s_t}(y_t)}{f^v_{s_t}(y_t)} < \eta_v, \text{ for some } v, \text{ and any } j \neq v \Big| H_k \Big) \\
&= \mathbb{P}\Big(\max_{a\in\{1,\dots,\tau\}} \sum_{t=1}^{a} \log \frac{f^v_{s_t}(y_t)}{f^j_{s_t}(y_t)} > -\log \eta_v, \text{ for some } v, \text{ and any } j \neq v \Big| H_k \Big) \\
&\leq \sum_{\substack{v=0\\ v\neq k}}^{M-1} \mathbb{P}\Big(\max_{a\in\{1,\dots,\tau\}} \sum_{t=1}^{a} \log \frac{f^v_{s_t}(y_t)}{f^k_{s_t}(y_t)} > -\log \eta_v \Big| H_k \Big) \\
&\quad + \mathbb{P}\Big(\max_{a\in\{1,\dots,\tau\}} \sum_{t=1}^{a} \log \frac{f^k_{s_t}(y_t)}{f^{j^*}_{s_t}(y_t)} > -\log \eta_k \Big| H_k \Big),
\end{aligned}$$

for some $j^* \in \{0,\dots,M-1\}\backslash\{k\}$. Observe that since $0 < \mathcal{D}(f^k_s, f^j_s) < \infty$, for each $j,k \in \{0,\dots,M-1\}, j \neq k$, and $s \in \{1,\dots,n\}$, the above right hand side goes to zero as $\eta_{\max} \to 0^+$. Hence, conditioned on a hypothesis H_k, the sample size for decision $N_d \to \infty$ in probability. This means that there exists a subsequence such that $N_d \to \infty$ almost surely. We further observe that N_d is a non decreasing as we decrease $\eta_{\max}$. Hence, conditioned on hypothesis H_k, $N_d \to \infty$, almost surely, as $\eta_{\max} \to 0^+$. □

Lemma 3.2 (***Theorem 5.2, [7]***)**.** Assume the sequences of random variables $\{Z^j_t\}_{t\in\mathbb{N}}$, $j \in \{1,\dots,d\}$, converge to μ_j almost surely as $t \to \infty$, with $0 < \min_{j\in\{1,\dots,d\}} \mu_j < \infty$. Then as $t \to \infty$, almost surely,

$$-\frac{1}{t} \log\Big(\sum_{j=1}^{d} e^{-tZ^j_t} \Big) \to \min_{j\in\{1,\dots,d\}} \mu_j.$$

□

Lemma 3.3 (***Corollary 7.4.1, [76]***)**.** Let $\{Z_t\}_{t\in\mathbb{N}}$ be independent sequence of random variables satisfying $\mathbb{E}[Z_t^2] < \infty$, for all $t \in \mathbb{N}$, and $\{b_t\}_{t\in\mathbb{N}}$ be a monotone sequence such that $b_t \to \infty$ as $t \to \infty$. If $\sum_{i=1}^{\infty} \text{Var}\ (Z_i/b_i) < \infty$, then

$$\frac{\sum_{i=1}^{t} Z_i - \mathbb{E}[\sum_{i=1}^{t} Z_i]}{b_t} \to 0, \text{ almost surely as } t \to \infty.$$

□

Lemma 3.4 (***Theorem 2.1, [40]***)**.** Let $\{Z_t\}_{t\in\mathbb{N}}$ be a sequence of random variables and $\{\tau(a)\}_{a\in\mathbb{R}_{\geq 0}}$ be a family of positive, integer valued random variables. Suppose that $Z_t \to Z$ almost surely as $t \to \infty$, and $\tau(a) \to \infty$ almost surely as $a \to \infty$. Then $Z_{\tau(a)} \to Z$ almost surely as $a \to \infty$. □

We now present the main result of this section, whose proof is a variation of the proofs for MSPRT in [7].

Theorem 3.5 (***MSPRT with randomized sensor selection***)**.** Assume finitely-informative sensors $\{1, \ldots, n\}$, and independent observations conditioned on hypothesis H_k, $k \in \{0, \ldots, M-1\}$. For the MSPRT with randomized sensor selection, the following statements hold:

(i). conditioned on a hypothesis, the sample size for decision N_d is finite almost surely;

(ii). conditioned on hypothesis H_k, the sample size for decision N_d, as $\eta_{\max} \to 0^+$, satisfies

$$\frac{N_d}{-\log \eta_k} \to \frac{1}{\sum_{s=1}^n q_s \mathcal{D}_s^*(k)} \quad \text{almost surely;}$$

(iii). the expected sample size satisfies

$$\frac{\mathbb{E}[N_d|H_k]}{-\log \eta_k} \to \frac{1}{\sum_{s=1}^n q_s \mathcal{D}_s^*(k)}, \quad \text{as } \eta_{\max} \to 0^+; \tag{3.2}$$

(iv). conditioned on hypothesis H_k, the decision time T_d, as $\eta_{\max} \to 0^+$, satisfies

$$\frac{T_d}{-\log \eta_k} \to \frac{\sum_{s=1}^n q_s \bar{T}_s}{\sum_{s=1}^n q_s \mathcal{D}_s^*(k)} \quad \text{almost surely;}$$

(v). the expected decision time satisfies

$$\frac{\mathbb{E}[T_d|H_k]}{-\log \eta_k} \to \frac{\sum_{s=1}^n q_s \bar{T}_s}{\sum_{s=1}^n q_s \mathcal{D}_s^*(k)} \equiv \frac{\boldsymbol{q} \cdot \boldsymbol{T}}{\boldsymbol{q} \cdot \boldsymbol{\mathcal{D}}^k}, \tag{3.3}$$

where $\boldsymbol{T}, \boldsymbol{\mathcal{D}}^k \in \mathbb{R}^n_{>0}$ are arrays of mean processing times $\bar{T}_s$ and minimum Kullback-Leibler distances $\mathcal{D}_s^*(k)$.

Proof. We start by establishing the first statement. We let $\eta_{\min} = \min\{\eta_j \mid j \in \{0, \ldots, M-1\}\}$. For any fixed $k \in \{0, \ldots, M-1\}$, the sample size for decision, denoted by N_d, satisfies

$$\begin{aligned} N_d &\leq \left(\text{first } \tau \geq 1 \text{ such that } \sum_{\substack{j=0 \\ j \neq k}}^{M-1} \prod_{t=1}^{\tau} \frac{f_{s_t}^j(y_t)}{f_{s_t}^k(y_t)} < \eta_{\min} \right) \\ &\leq \left(\text{first } \tau \geq 1 \text{ such that } \prod_{t=1}^{\tau} \frac{f_{s_t}^j(y_t)}{f_{s_t}^k(y_t)} < \frac{\eta_{\min}}{M-1}, \text{for all } j \in \{0, \ldots, M-1\}, j \neq k \right). \end{aligned}$$

Therefore, it follows that

$$
\begin{aligned}
&\mathbb{P}(N_d > \tau | H_k) \\
&\leq \mathbb{P}\Big(\prod_{t=1}^{\tau} \frac{f_{s_t}^j(y_t)}{f_{s_t}^k(y_t)} \geq \frac{\eta_{\min}}{M-1}, j \in \{0, \dots, M-1\} \setminus \{k\} \Big| H_k \Big) \\
&\leq \sum_{\substack{j=0 \\ j \neq k}}^{M-1} \mathbb{P}\Big(\prod_{t=1}^{\tau} \frac{f_{s_t}^j(y_t)}{f_{s_t}^k(y_t)} \geq \frac{\eta_{\min}}{M-1} \Big| H_k \Big) \\
&= \sum_{\substack{j=0 \\ j \neq k}}^{M-1} \mathbb{P}\Big(\prod_{t=1}^{\tau} \sqrt{\frac{f_{s_t}^j(y_t)}{f_{s_t}^k(y_t)}} \geq \sqrt{\frac{\eta_{\min}}{M-1}} \Big| H_k \Big) \\
&\leq \sum_{\substack{j=0 \\ j \neq k}}^{M-1} \sqrt{\frac{M-1}{\eta_{\min}}} \mathbb{E}\left[\sqrt{\frac{f_{s^*(j)}^j(Y)}{f_{s^*(j)}^k(Y)}} \Big| H_k \right]^{\tau} \qquad (3.4) \\
&\leq \frac{(M-1)^{\frac{3}{2}}}{\sqrt{\eta_{\min}}} \big(\max_{j \in \{0,\dots,M-1\} \setminus \{k\}} \rho_j \big)^{\tau},
\end{aligned}
$$

where $s^*(j) = \operatorname{argmax}_{s \in \{1,\dots,n\}} \mathbb{E}\left[\sqrt{\frac{f_s^j(Y)}{f_s^k(Y)}} \Big| H_k \right]$, and

$$
\begin{aligned}
\rho_j &= \mathbb{E}\left[\sqrt{\frac{f_{s^*(j)}^j(Y)}{f_{s^*(j)}^k(Y)}} \Big| H_k \right] = \int_{\mathbb{R}} \sqrt{f_{s^*(j)}^j(Y) f_{s^*(j)}^k(Y)} dY \\
&< \sqrt{\int_{\mathbb{R}} f_{s^*(j)}^j(Y) dY} \sqrt{\int_{\mathbb{R}} f_{s^*(j)}^k(Y) dY} = 1.
\end{aligned}
$$

The inequality (3.4) follows from the Markov inequality, while $\rho_j < 1$ follows from the Cauchy-Schwarz inequality. Note that the Cauchy-Schwarz inequality is strict because $f_{s^*(j)}^j \neq f_{s^*(j)}^k$ almost surely in $f_{s^*(j)}^k$. To establish almost sure convergence, note that

$$
\sum_{\tau=1}^{\infty} \mathbb{P}(N_d > \tau | H_k) \leq \sum_{\tau=1}^{\infty} \frac{(M-1)^{\frac{3}{2}}}{\sqrt{\eta_{\min}}} \big(\max_{j \in \{0,\dots,M-1\} \setminus \{k\}} \rho_j \big)^{\tau} < \infty.
$$

Therefore, by Borel-Cantelli lemma [76], it follows that

$$\mathbb{P}(\limsup_{\tau\to\infty}[N_d > \tau]) = 1 - \mathbb{P}(\liminf_{\tau\to\infty}[N_d \leq \tau]) = 0.$$

Thus, for τ large enough, all realizations in the set $\liminf_{\tau\to\infty}[N_d \leq \tau]$, converge in finite number of steps. This proves the almost sure convergence on the MSPRT with randomized sensor selection.

To prove the second statement, for hypothesis H_k, let

$$\tilde{N}_d = \left(\text{first } \tau \geq 1 \text{ such that } \sum_{\substack{j=0\\ j\neq k}}^{M-1} \prod_{t=1}^{\tau} \frac{f_{s_t}^j(y_t)}{f_{s_t}^k(y_t)} < \eta_k\right),$$

and, accordingly, note that

$$\sum_{\substack{j=0\\ j\neq k}}^{M-1} \prod_{t=1}^{\tilde{N}_d-1} \frac{f_{s_t}^j(y_t)}{f_{s_t}^k(y_t)} \geq \eta_k, \quad \text{and} \quad \sum_{\substack{j=0\\ j\neq k}}^{M-1} \prod_{t=1}^{\tilde{N}_d} \frac{f_{s_t}^j(y_t)}{f_{s_t}^k(y_t)} < \eta_k.$$

Some algebraic manipulations on these inequalities yield

$$\begin{aligned} \frac{-1}{\tilde{N}_d - 1} \log\left(\sum_{\substack{j=0\\ j\neq k}}^{M-1} \mathbb{E}\Big(-\sum_{t=1}^{\tilde{N}_d-1} \log \frac{f_{s_t}^k(y_t)}{f_{s_t}^j(y_t)}\Big)\right) &\leq \frac{-\log \eta_k}{\tilde{N}_d - 1}, \\ \frac{-1}{\tilde{N}_d} \log\left(\sum_{\substack{j=0\\ j\neq k}}^{M-1} \mathbb{E}\Big(-\sum_{t=1}^{\tilde{N}_d} \log \frac{f_{s_t}^k(y_t)}{f_{s_t}^j(y_t)}\Big)\right) &> \frac{-\log \eta_k}{\tilde{N}_d}. \end{aligned} \tag{3.5}$$

Observe that $\tilde{N}_d \geq N_d$, hence from Lemma 3.1, $\tilde{N}_d \to \infty$ almost surely as $\eta_{\max} \to 0^+$. In the limit $\tilde{N}_d \to \infty$, the supremum and infimum in inequalities (3.5) converge to the same value. From Lemma 3.3, and Lemma 3.4

$$\frac{1}{\tilde{N}_d}\sum_{t=1}^{\tilde{N}_d} \log \frac{f_{s_t}^k(y_t)}{f_{s_t}^j(y_t)} \to \frac{1}{\tilde{N}_d}\sum_{t=1}^{\tilde{N}_d} \mathbb{E}\left[\log \frac{f_{s_t}^k(y_t)}{f_{s_t}^j(y_t)}\middle| H_k\right] \to \sum_{s=1}^{n} q_s \mathcal{D}(f_s^k, f_s^j), \text{ almost surely,}$$

as $\tilde{N}_d \to \infty$. Lemma 3.2 implies that the left hand sides of the inequalities (3.5) almost surely converge to

$$\min_{j\in\{0,\dots,M-1\}\setminus\{k\}} \mathbb{E}\left[\log\frac{f_s^k(Y)}{f_s^j(Y)}\middle| H_k\right] = \sum_{s=1}^{n} q_s \mathcal{D}_s^*(k).$$

Hence, conditioned on hypothesis H_k

$$\frac{\tilde{N}_d}{-\log\eta_k} \to \frac{1}{\sum_{s=1}^n q_s\mathcal{D}_s^*(k)}$$

almost surely, as $\eta_{\max}\to 0^+$.

Now, notice that

$$\begin{aligned}
&\mathbb{P}\left(\left|\frac{N_d}{-\log\eta_k} - \frac{1}{\sum_{s=1}^n q_s\mathcal{D}_s^*(k)}\right| > \epsilon \middle| H_k\right)\\
=&\sum_{v=0}^{M-1}\mathbb{P}\left(\left|\frac{N_d}{-\log\eta_k} - \frac{1}{\sum_{s=1}^n q_s\mathcal{D}_s^*(k)}\right| > \epsilon \text{ \& accept } H_v \middle| H_k\right)\\
=&\mathbb{P}\left(\left|\frac{\tilde{N}_d}{-\log\eta_k} - \frac{1}{\sum_{s=1}^n q_s\mathcal{D}_s^*(k)}\right| > \epsilon \middle| H_k\right)\\
+&\sum_{\substack{v=0\\v\neq k}}^{M-1}\mathbb{P}\left(\left|\frac{N_d}{-\log\eta_k} - \frac{1}{\sum_{s=1}^n q_s\mathcal{D}_s^*(k)}\right| > \epsilon \text{ \& accept } H_v \middle| H_k\right).
\end{aligned}$$

Note that $\alpha_j \to 0^+$, for all $j\in\{0,\dots,M-1\}$, as $\eta_{\max}\to 0^+$. Hence, the right hand side terms above converge to zero as $\eta_{\max}\to 0^+$. This establishes the second statement. We have proved almost sure convergence of $\frac{N_d}{-\log\eta_k}$. To establish convergence in expected value, we construct a Lebesgue integrable upper bound of N_d. Define $\xi_0 = 0$, and for all $m\geq 1$,

$$\xi_m = \left(\text{first } \tau\geq 1 \text{ such that } \sum_{t=\xi_{m-1}+1}^{\xi_{m-1}+\tau}\log\frac{f_{s_t}^k(y_t)}{f_{s_t}^j(y_t)} > 1, \text{ for } j\in\{0,\dots,M-1\}\setminus\{k\}\right).$$

Note that the variables in the sequence $\{\xi_i\}_{i\in\mathbb{N}}$ are i.i.d., and moreover $\mathbb{E}[\xi_1|H_k] < \infty$, since $\mathcal{D}(f_k^s, f_j^s) > 0$, for all $s \in \{1,\ldots,n\}$, and $j \in \{0,\ldots,M-1\} \setminus \{k\}$.

Choose $\tilde{\eta} = \lceil \log \frac{M-1}{\eta_k} \rceil$. Note that

$$\sum_{t=1}^{\xi_1+\ldots+\xi_{\tilde{\eta}}} \log \frac{f_{s_t}^k(y_t)}{f_{s_t}^j(y_t)} > \tilde{\eta}, \text{ for } j \in \{0,\ldots,M-1\} \setminus \{k\}.$$

Hence, $\xi_1 + \ldots + \xi_{\tilde{\eta}} \geq N_d$. Further, $\xi_1 + \ldots + \xi_{\tilde{\eta}}$ is Lebesgue integrable. The third statement follows from the Lebesgue dominated convergence theorem [76].

To establish the next statement, note that the decision time of MSPRT with randomized sensor selection is the sum of sensor's processing time at each iteration, i.e.,

$$T_d = T_{s_1} + \ldots + T_{s_{N_d}}.$$

From Lemma 3.3, Lemma 3.1 and Lemma 3.4, it follows that

$$\frac{T_d}{N_d} \to \frac{1}{N_d} \sum_{t=1}^{N_d} \mathbb{E}[T_{s_t}] \to \sum_{s=1}^{n} q_s \bar{T}_s,$$

almost surely, as $\eta_{\max} \to 0^+$. Thus, conditioned on hypothesis H_k,

$$\lim_{\eta_{\max}\to 0^+} \frac{T_d}{-\log \eta_k} = \lim_{\eta_{\max}\to 0^+} \frac{T_d}{N_d} \frac{N_d}{-\log \eta_k} = \lim_{\eta_{\max}\to 0^+} \frac{T_d}{N_d} \lim_{\eta_{\max}\to 0^+} \frac{N_d}{-\log \eta_k} = \frac{\sum_{s=1}^n q_s \bar{T}_s}{\sum_{s=1}^n q_s \mathcal{D}_s^*(k)},$$

almost surely. Now, note that $\{(s_t, T_{s_t})\}_{t\in\mathbb{N}}$ is an i.i.d. realization of the pair (s, T_s). Therefore, by the Wald's identity [76]

$$\mathbb{E}[T_{\xi_1}] = \mathbb{E}\big[\sum_{t=1}^{\xi_1} T_{s_t}\big] = \mathbb{E}[\xi_1]\mathbb{E}[T_s] < \infty.$$

Also, $T_d \leq T_{\xi_1} + \ldots + T_{\xi_{\tilde{\eta}}} \in \mathcal{L}^1$. Thus, by the Lebesgue dominated convergence theorem [76]

$$\frac{\mathbb{E}[T_d|H_k]}{-\log \eta_k} \to \frac{\sum_{s=1}^n q_s \bar{T}_s}{\sum_{s=1}^n q_s \mathcal{D}_s^*(k)} = \frac{\boldsymbol{q} \cdot \boldsymbol{T}}{\boldsymbol{q} \cdot \boldsymbol{\mathcal{D}}^k} \quad \text{as } \eta_{\max} \to 0^+.$$

□

Remark 3.3. The results in Theorem 3.5 hold if we have at least one sensor with positive minimum Kullback-Leibler divergence $\mathcal{D}_s^*(k)$, which is chosen with a positive probability. Thus, the MSPRT with randomized sensor selection is robust to sensor failure and uninformative sensors. In what follows, we assume that at least M sensors are finitely informative. □

Remark 3.4. In the remainder of the chapter, we assume that the error probabilities are chosen small enough, so that the expected decision time is arbitrarily close to the expression in equation (3.3). □

Remark 3.5. The MSPRT with randomized sensor selection may not be the optimal sequential test. In fact, this test corresponds to a stationary open-loop strategy. In this chapter we wish to determine a time-optimal stationary open-loop strategy, as motivated in Remark 3.2. □

Remark 3.6. If the minimum Kullback-Leibler divergence $\mathcal{D}_s^*(k)$ is the same for any given $s \in \{1, \ldots, n\}$, and for each $k \in \{0, \ldots, M-1\}$, and all thresholds

η_k are identical, then the expected decision time is the same conditioned on any hypothesis H_k. For example, if conditioned on hypothesis $H_k, k \in \{0, \dots, M-1\}$, and sensor $s \in \{1, \dots, n\}$, the observation is generated from a Gaussian distribution with mean k and variance σ_s^2, then the minimum Kullback-Leibler divergence from hypothesis k, for sensor s is $\mathcal{D}_s^*(k) = 1/2\sigma_s^2$, which is independent of k. □

3.2 Optimal sensor selection

In this section we consider sensor selection problems with the aim to minimize the expected decision time of a sequential hypothesis test with randomized sensor selection. As exemplified in Theorem 3.5, the problem features multiple conditioned decision times and, therefore, multiple distinct cost functions are of interest. In Section 3.2.1 below, we aim to minimize the decision time conditioned upon one specific hypothesis being true; in Section 3.2.2 and 3.2.3 we will consider worst-case and average decision times. In all three scenarios the decision variables take values in the probability simplex.

Minimizing decision time conditioned upon a specific hypothesis may be of interest when fast reaction is required in response to the specific hypothesis being indeed true. For example, in change detection problems one aims to quickly detect a change in a stochastic process; the CUSUM algorithm (also referred to as

Page's test) [68] is widely used in such problems. It is known [6] that, with fixed threshold, the CUSUM algorithm for quickest change detection is equivalent to an SPRT on the observations taken after the change has occurred. We consider the minimization problem for a single conditioned decision time in Section 3.2.1 below and we show that, in this case, observing the best sensor each time is the optimal strategy.

In general, no specific hypothesis might play a special role in the problem and, therefore, it is of interest to simultaneously minimize multiple decision times over the probability simplex. This is a multi-objective optimization problem, and may have Pareto-optimal solutions. We tackle this problem by constructing a single aggregate objective function. In the binary hypothesis case, we construct two single aggregate objective functions as the maximum and the average of the two conditioned decision times. These two functions are discussed in Section 3.2.2 and Section 3.2.3 respectively. In the multiple hypothesis setting, we consider the single aggregate objective function constructed as the average of the conditioned decision times. An analytical treatment of this function for $M > 2$, is difficult. We determine the optimal number of sensors to be observed, and direct the interested reader to some iterative algorithms to solve such optimization problems. This case is also considered under Section 3.2.3.

Before we pose the problem of optimal sensor selection, we introduce the following notation. We denote the probability simplex in $\mathbb{R}^n$ by Δ_{n-1}, and the vertices of the probability simplex Δ_{n-1} by $\boldsymbol{e}_i$, $i \in \{1, \ldots, n\}$. We refer to the line joining any two vertices of the simplex as an *edge*. Finally, we define $g^k : \Delta_{n-1} \to \mathbb{R}$, $k \in \{0, \ldots, M-1\}$, by $g^k(\boldsymbol{q}) = \boldsymbol{q} \cdot \boldsymbol{T}/\boldsymbol{q} \cdot \boldsymbol{I}^k$, where $\boldsymbol{I}^k = -\boldsymbol{\mathcal{D}}^k / \log \eta_k$.

We also recall following definition from [15]:

Definition 3.1 (***Linear-fractional function***)**.** Given parameters $A \in \mathbb{R}^{l \times p}$, $B \in \mathbb{R}^l$, $c \in \mathbb{R}^p$, and $d \in \mathbb{R}$, the function $g : \{z \in \mathbb{R}^p \mid c^T z + d > 0\} \to \mathbb{R}^l$, defined by

$$g(x) = \frac{Ax + B}{c^T x + d},$$

is called a *linear-fractional function* [15]. A linear-fractional function is quasi-convex as well as quasi-concave. In particular, if $l = 1$, then any scalar linear-fractional function g satisfies

$$\begin{aligned} g(\nu x + (1-\nu)y) &\leq \max\{g(x), g(y)\}, \\ g(\nu x + (1-\nu)y) &\geq \min\{g(x), g(y)\}, \end{aligned} \tag{3.6}$$

for all $\nu \in [0, 1]$ and $x, y \in \{z \in \mathbb{R}^p \mid c^T z + d > 0\}$.

3.2.1 Optimization of conditioned decision time

We consider the case when the supervisor is trying to detect a particular hypothesis, irrespective of the present hypothesis. The corresponding optimization

problem for a fixed $k \in \{0, \dots, M-1\}$ is posed in the following way:

$$\begin{aligned} &\text{minimize} \quad g^k(\boldsymbol{q}) \\ &\text{subject to} \quad \boldsymbol{q} \in \Delta_{n-1}. \end{aligned} \tag{3.7}$$

The solution to this minimization problem is given in the following theorem.

Theorem 3.6 (***Optimization of conditioned decision time***)**.** The solution to the minimization problem (3.7) is $\boldsymbol{q}^* = \boldsymbol{e}_{s^*}$, where s^* is given by

$$s^* = \underset{s \in \{1,\dots,n\}}{\operatorname{argmin}} \frac{T_s}{I_s^k},$$

and the minimum objective function is

$$\mathbb{E}[T_d^* | H_k] = \frac{T_{s^*}}{I_{s^*}^k}. \tag{3.8}$$

Proof. We notice that objective function is a linear-fractional function. In the following argument, we show that the minima occurs at one of the vertices of the simplex.

We first notice that the probability simplex is the convex hull of the vertices, i.e., any point $\tilde{\boldsymbol{q}}$ in the probability simplex can be written as

$$\tilde{\boldsymbol{q}} = \sum_{s=1}^{n} \alpha_s \boldsymbol{e}_s, \qquad \sum_{s=1}^{n} \alpha_s = 1, \text{ and } \alpha_s \geq 0.$$

We invoke equation (3.6), and observe that for some $\beta \in [0,1]$ and for any $s, r \in \{1, \dots, n\}$

$$g^k(\beta \boldsymbol{e}_s + (1-\beta)\boldsymbol{e}_r) \geq \min\{g^k(\boldsymbol{e}_s), g^k(\boldsymbol{e}_r)\}, \tag{3.9}$$

which can be easily generalized to

$$g^k(\tilde{\boldsymbol{q}}) \geq \min_{s\in\{1,\dots,n\}} g^k(\boldsymbol{e}_s), \tag{3.10}$$

for any point $\tilde{\boldsymbol{q}}$ in the probability simplex Δ_{n-1}. Hence, minima will occur at one of the vertices $\boldsymbol{e}_{s^*}$, where s^* is given by

$$s^* = \operatorname*{argmin}_{s\in\{1,\dots,n\}} g^k(\boldsymbol{e}_s) = \operatorname*{argmin}_{s\in\{1,\dots,n\}} \frac{T_s}{I_s^k}.$$

□

3.2.2 Optimization of the worst case decision time

For the binary hypothesis testing, we consider the multi-objective optimization problem of minimizing both decision times simultaneously. We construct single aggregate objective function by considering the maximum of the two objective functions. This turns out to be a worst case analysis, and the optimization problem for this case is posed in the following way:

$$\begin{aligned} \text{minimize} \quad & \max\left\{g^0(\boldsymbol{q}), g^1(\boldsymbol{q})\right\}, \\ \text{subject to} \quad & \boldsymbol{q} \in \Delta_{n-1}. \end{aligned} \tag{3.11}$$

Before we move on to the solution of above minimization problem, we state the following results.

Lemma 3.7 (***Monotonicity of conditioned decision times***)**.** The functions g^k, $k \in \{0, \dots, M-1\}$ are monotone on the probability simplex Δ_{n-1}, in the sense that given two points $\boldsymbol{q}_a, \boldsymbol{q}_b \in \Delta_{n-1}$, the function g^k is monotonically non-increasing or monotonically non-decreasing along the line joining $\boldsymbol{q}_a$ and $\boldsymbol{q}_b$.

Proof. Consider probability vectors $\boldsymbol{q}_a, \boldsymbol{q}_b \in \Delta_{n-1}$. Any point $\boldsymbol{q}$ on line joining $\boldsymbol{q}_a$ and $\boldsymbol{q}_b$ can be written as $\boldsymbol{q}(\nu) = \nu\boldsymbol{q}_a + (1-\nu)\boldsymbol{q}_b$, $\nu \in]0,1[$. We note that $g^k(\boldsymbol{q}(\nu))$ is given by:

$$g^k(\boldsymbol{q}(\nu)) = \frac{\nu(\boldsymbol{q}_a \cdot T) + (1-\nu)(\boldsymbol{q}_b \cdot \boldsymbol{T})}{\nu(\boldsymbol{q}_a \cdot \boldsymbol{I}^k) + (1-\nu)(\boldsymbol{q}_b \cdot \boldsymbol{I}^k)}.$$

The derivative of g^k along the line joining $\boldsymbol{q}_a$ and $\boldsymbol{q}_b$ is given by

$$\frac{d}{d\nu} g^k(\boldsymbol{q}(\nu)) = \big(g^k(\boldsymbol{q}_a) - g^k(\boldsymbol{q}_b)\big) \frac{(\boldsymbol{q}_a \cdot \boldsymbol{I}^k)(\boldsymbol{q}_b \cdot \boldsymbol{I}^k)}{(\nu(\boldsymbol{q}_a \cdot \boldsymbol{I}^k) + (1-\nu)(\boldsymbol{q}_b \cdot \boldsymbol{I}^k))^2}.$$

We note that the sign of the derivative of g^k along the line joining two points $\boldsymbol{q}_a, \boldsymbol{q}_b$ is fixed by the choice of $\boldsymbol{q}_a$ and $\boldsymbol{q}_b$. Hence, the function g^k is monotone over the line joining $\boldsymbol{q}_a$ and $\boldsymbol{q}_b$. Moreover, note that if $g^k(\boldsymbol{q}_a) \neq g^k(\boldsymbol{q}_b)$, then g^k is strictly monotone. Otherwise, g^k is constant over the line joining $\boldsymbol{q}_a$ and $\boldsymbol{q}_b$. □

Lemma 3.8 (***Location of min-max***)**.** Define $g : \Delta_{n-1} \to \mathbb{R}_{\geq 0}$ by $g = \max\{g^0, g^1\}$. A minimum of g lies at the intersection of the graphs of g^0 and g^1, or at some vertex of the probability simplex Δ_{n-1}.

Proof. The idea of the proof is illustrated in Figure 3.2. We now prove it rigorously.

Case 1: The graphs of g^0 and g^1 do not intersect at any point in the simplex Δ_{n-1}.

In this case, one of the functions g^0 and g^1 is an upper bound to the other function at every point in the probability simplex Δ_{n-1}. Hence, $g = g^k$, for some $k \in \{0, 1\}$, at every point in the probability simplex Δ_{n-1}. From Theorem 3.6, we know that the minima of g^k on the probability simplex Δ_{n-1} lie at some vertex of the probability simplex Δ_{n-1}.

Case 2: The graphs of g^0 and g^1 intersect at a set Q in the probability simplex Δ_{n-1}, and let $\bar{\boldsymbol{q}}$ be some point in the set Q.

Suppose, a minimum of g occurs at some point $\boldsymbol{q}^* \in \text{relint}(\Delta_{n-1})$, and $\boldsymbol{q}^* \notin Q$, where relint$(\cdot)$ denotes the relative interior. With out loss of generality, we can assume that $g^0(\boldsymbol{q}^*) > g^1(\boldsymbol{q}^*)$. Also, $g^0(\bar{\boldsymbol{q}}) = g^1(\bar{\boldsymbol{q}})$, and $g^0(\boldsymbol{q}^*) < g^0(\bar{\boldsymbol{q}})$ by assumption.

We invoke Lemma 3.7, and notice that g^0 and g^1 can intersect at most once on a line. Moreover, we note that $g^0(\boldsymbol{q}^*) > g^1(\boldsymbol{q}^*)$, hence, along the half-line from $\bar{\boldsymbol{q}}$ through $\boldsymbol{q}^*$, $g^0 > g^1$, that is, $g = g^0$. Since $g^0(\boldsymbol{q}^*) < g^0(\bar{\boldsymbol{q}})$, g is decreasing along this half-line. Hence, g should achieve its minimum at the boundary of the simplex Δ_{n-1}, which contradicts that $\boldsymbol{q}^*$ is in the relative interior of the simplex

Δ_{n-1}. In summary, if a minimum of g lies in the relative interior of the probability simplex Δ_{n-1}, then it lies at the intersection of the graphs of g^0 and g^1.

The same argument can be applied recursively to show that if a minimum lies at some point $\boldsymbol{q}^\dagger$ on the boundary, then either $g^0(\boldsymbol{q}^\dagger) = g^1(\boldsymbol{q}^\dagger)$ or the minimum lies at the vertex. □

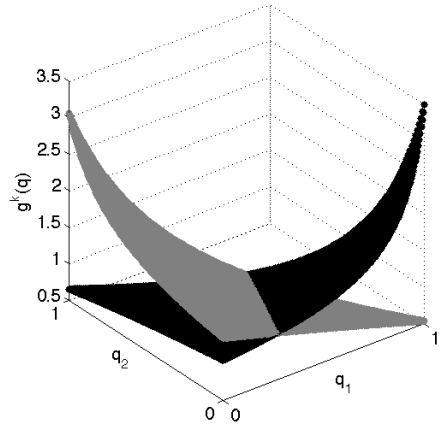


Figure 3.2: Linear-fractional functions. Both the functions achieve their minima at some vertex of the simplex. The maximum of the two functions achieves its minimum at the intersection of two graphs.

In the following arguments, let Q be the set of points in the simplex Δ_{n-1}, where $g^0 = g^1$, that is,

$$Q = \{\boldsymbol{q} \in \Delta_{n-1} \mid \boldsymbol{q} \cdot (\boldsymbol{I}^0 - \boldsymbol{I}^1) = 0\}. \tag{3.12}$$

Also notice that the set Q is non empty if and only if $\boldsymbol{I}^0 - \boldsymbol{I}^1$ has at least one non-negative and one non-positive entry. If the set Q is empty, then it follows from Lemma 3.8 that the solution of optimization problem in equation (3.11) lies

at some vertex of the probability simplex Δ_{n-1}. Now we consider the case when Q is non empty. We assume that the sensors have been re-ordered such that the entries in $\boldsymbol{I}^0 - \boldsymbol{I}^1$ are in ascending order. We further assume that, for $\boldsymbol{I}^0 - \boldsymbol{I}^1$, the first m entries, $m < n$, are non positive, and the remaining entries are positive.

Lemma 3.9 (***Intersection polytope***)**.** If the set Q defined in equation (3.12) is non empty, then the polytope generated by the points in the set Q has vertices given by:

$$\tilde{Q} = \{\tilde{\boldsymbol{q}}^{sr} \mid s \in \{1, \ldots, m\} \text{ and } r \in \{m+1, \ldots, n\}\},$$

where for each $i \in \{1, \ldots, n\}$

$$\tilde{q}_i^{sr} = \begin{cases} \dfrac{(I_r^0 - I_r^1)}{(I_r^0 - I_r^1) - (I_s^0 - I_s^1)}, & \text{if } i = s, \\ 1 - \tilde{q}_s^{sr}, & \text{if } i = r, \\ 0, & \text{otherwise.} \end{cases} \tag{3.13}$$

Proof. Any $q \in Q$ satisfies the following constraints

$$\sum_{s=1}^{n} q_s = 1, \quad q_s \in [0, 1], \tag{3.14}$$

$$\sum_{s=1}^{n} q_s (I_s^0 - I_s^1) = 0, \tag{3.15}$$

Eliminating q_n, using equation (3.14) and equation (3.15), we get:

$$\sum_{s=1}^{n-1} \beta_s q_s = 1, \text{ where } \beta_s = \frac{(I_n^0 - I_n^1) - (I_s^0 - I_s^1)}{(I_n^0 - I_n^1)}. \tag{3.16}$$

The equation (3.16) defines a hyperplane, whose extreme points in $\mathbb{R}^{n-1}_{\geq 0}$ are given by

$$\tilde{\boldsymbol{q}}^{sn} = \frac{1}{\beta_s}\boldsymbol{e}_s, \quad s \in \{1, \ldots, n-1\}.$$

Note that for $s \in \{1, \ldots, m\}$, $\tilde{\boldsymbol{q}}^{sn} \in \Delta_{n-1}$. Hence, these points define some vertices of the polytope generated by points in the set Q. Also note that the other vertices of the polytope can be determined by the intersection of each pair of lines through $\tilde{\boldsymbol{q}}^{sn}$ and $\tilde{\boldsymbol{q}}^{rn}$, and $\boldsymbol{e}_s$ and $\boldsymbol{e}_r$, for $s \in \{1, \ldots, m\}$, and $r \in \{m+1, \ldots, n-1\}$. In particular, these vertices are given by $\tilde{\boldsymbol{q}}^{sr}$ defined in equation (3.13).

Hence, all the vertices of the polytopes are defined by $\tilde{\boldsymbol{q}}^{sr}$, $s \in \{1, \ldots, m\}$, $r \in \{m+1, \ldots, n\}$. Therefore, the set of vertices of the polygon generated by the points in the set Q is $\tilde{Q}$. □

Before we state the solution to the optimization problem (3.11), we define the following:

$$(s^*, r^*) \in \underset{\substack{r \in \{m+1, \ldots, n\} \\ s \in \{1, \ldots, m\}}}{\operatorname{argmin}} \frac{(I_r^0 - I_r^1)T_s - (I_s^0 - I_s^1)T_r}{I_s^1 I_r^0 - I_s^0 I_r^1}, \quad \text{and}$$

$$g_{\text{two-sensors}}(s^*, r^*) = \frac{(I_{r^*}^0 - I_{r^*}^1)T_{s^*} - (I_{s^*}^0 - I_{s^*}^1)T_{r^*}}{I_{s^*}^1 I_{r^*}^0 - I_{s^*}^0 I_{r^*}^1}.$$

We also define

$$w^* = \underset{w \in \{1, \ldots, n\}}{\operatorname{argmin}} \max\left\{\frac{T_w}{I_w^0}, \frac{T_w}{I_w^1}\right\}, \text{ and}$$

$$g_{\text{one-sensor}}(w^*) = \max\left\{\frac{T_{w^*}}{I_{w^*}^0}, \frac{T_{w^*}}{I_{w^*}^1}\right\}.$$

Theorem 3.10 (***Worst case optimization***)**.** For the optimization problem (3.11), an optimal probability vector is given by:

$$\boldsymbol{q}^* = \begin{cases} \boldsymbol{e}_{w^*}, & \text{if } g_{\text{one-sensor}}(w^*) \leq g_{\text{two-sensors}}(s^*, r^*), \\ \tilde{\boldsymbol{q}}^{s^* r^*}, & \text{if } g_{\text{one-sensor}}(w^*) > g_{\text{two-sensors}}(s^*, r^*), \end{cases}$$

and the minimum value of the function is given by:

$$\min \left\{ g_{\text{one-sensor}}(w^*), g_{\text{two-sensors}}(s^*, r^*) \right\}.$$

Proof. We invoke Lemma 3.8, and note that a minimum should lie at some vertex of the simplex Δ_{n-1}, or at some point in the set Q. Note that $g^0 = g^1$ on the set Q, hence the problem of minimizing $\max\{g^0, g^1\}$ reduces to minimizing g^0 on the set Q. From Theorem 3.6, we know that g^0 achieves the minima at some extreme point of the feasible region. From Lemma 3.9, we know that the vertices of the polytope generated by points in set Q are given by set $\tilde{Q}$. We further note that $g_{\text{two-sensors}}(s, r)$ and $g_{\text{one-sensor}}(w)$ are the value of objective function at the points in the set $\tilde{Q}$ and the vertices of the probability simplex Δ_{n-1} respectively, which completes the proof. □

3.2.3 Optimization of the average decision time

For the multi-objective optimization problem of minimizing all the decision times simultaneously on the simplex, we formulate the single aggregate objec-

tive function as the average of these decision times. The resulting optimization problem, for $M \geq 2$, is posed in the following way:

$$\begin{aligned} \text{minimize} \quad & \frac{1}{M}(g^0(\boldsymbol{q}) + \ldots + g^{M-1}(\boldsymbol{q})), \\ \text{subject to} \quad & \boldsymbol{q} \in \Delta_{n-1}. \end{aligned} \tag{3.17}$$

In the following discussion we assume $n > M$, unless otherwise stated. We analyze the optimization problem in equation (3.17) as follows:

Lemma 3.11 (***Non-vanishing Jacobian***)**.** The objective function in optimization problem in equation (3.17) has no critical point on Δ_{n-1} if the vectors $\boldsymbol{T}, \boldsymbol{I}^0, \ldots, \boldsymbol{I}^{M-1} \in \mathbb{R}^n_{>0}$ are linearly independent.

Proof. The Jacobian of the objective function in the optimization problem in equation (3.17) is

$$\frac{1}{M}\frac{\partial}{\partial \boldsymbol{q}} \sum_{k=0}^{M-1} g^k = \Gamma \psi(\boldsymbol{q}),$$

$$\text{where } \Gamma = \frac{1}{M}\begin{bmatrix} \boldsymbol{T} & -\boldsymbol{I}^0 & \ldots & -\boldsymbol{I}^{M-1} \end{bmatrix} \in \mathbb{R}^{n \times (M+1)}, \text{ and}$$

$\psi : \Delta_{n-1} \to \mathbb{R}^{M+1}$ is defined by

$$\psi(\boldsymbol{q}) = \begin{bmatrix} \displaystyle\sum_{k=0}^{M-1} \frac{1}{\boldsymbol{q} \cdot \boldsymbol{I}^k} & \dfrac{\boldsymbol{q} \cdot \boldsymbol{T}}{(\boldsymbol{q} \cdot \boldsymbol{I}^0)^2} & \cdots & \dfrac{\boldsymbol{q} \cdot \boldsymbol{T}}{(\boldsymbol{q} \cdot \boldsymbol{I}^{M-1})^2} \end{bmatrix}^{\mathrm{T}}.$$

For $n > M$, if the vectors $\boldsymbol{T}, \boldsymbol{I}^0, \ldots, \boldsymbol{I}^{M-1}$ are linearly independent, then Γ is full rank. Further, the entries of ψ are non-zero on the probability simplex Δ_{n-1}. Hence, the Jacobian does not vanish anywhere on the probability simplex Δ_{n-1}. □

Lemma 3.12 (***Case of Independent Information***)**.** For $M = 2$, if $\boldsymbol{I}^0$ and $\boldsymbol{I}^1$ are linearly independent, and $\boldsymbol{T} = \alpha_0 \boldsymbol{I}^0 + \alpha_1 \boldsymbol{I}^1$, for some $\alpha_0, \alpha_1 \in \mathbb{R}$, then the following statements hold:

(i). if α_0 and α_1 have opposite signs, then $g^0 + g^1$ has no critical point on the simplex Δ_{n-1}, and

(ii). for $\alpha_0, \alpha_1 > 0$, $g^0 + g^1$ has a critical point on the simplex Δ_{n-1} if and only if there exists $v \in \Delta_{n-1}$ perpendicular to the vector $\sqrt{\alpha_0}\boldsymbol{I}^0 - \sqrt{\alpha_1}\boldsymbol{I}^1$.

Proof. We notice that the Jacobian of $g^0 + g^1$ satisfies

$$\begin{aligned} &(\boldsymbol{q}\cdot\boldsymbol{I}^0)^2(\boldsymbol{q}\cdot\boldsymbol{I}^1)^2\frac{\partial}{\partial \boldsymbol{q}}(g^0+g^1) \\ &\quad= \boldsymbol{T}\left((\boldsymbol{q}\cdot\boldsymbol{I}^0)(\boldsymbol{q}\cdot\boldsymbol{I}^1)^2 + (\boldsymbol{q}\cdot\boldsymbol{I}^1)(\boldsymbol{q}\cdot\boldsymbol{I}^0)^2\right) \\ &\qquad - \boldsymbol{I}^0(\boldsymbol{q}\cdot\boldsymbol{T})(\boldsymbol{q}\cdot\boldsymbol{I}^1)^2 - \boldsymbol{I}^1(\boldsymbol{q}\cdot\boldsymbol{T})(\boldsymbol{q}\cdot\boldsymbol{I}^0)^2. \end{aligned} \tag{3.18}$$

Substituting $\boldsymbol{T} = \alpha_0\boldsymbol{I}^0 + \alpha_1\boldsymbol{I}^1$, equation (3.18) becomes

$$\begin{aligned} &(\boldsymbol{q}\cdot\boldsymbol{I}^0)^2(\boldsymbol{q}\cdot\boldsymbol{I}^1)^2\frac{\partial}{\partial \boldsymbol{q}}(g^0+g^1) \\ &\qquad= \left(\alpha_0(\boldsymbol{q}\cdot\boldsymbol{I}^0)^2 - \alpha_1(\boldsymbol{q}\cdot\boldsymbol{I}^1)^2\right)\left((\boldsymbol{q}\cdot\boldsymbol{I}^1)\boldsymbol{I}^0 - (\boldsymbol{q}\cdot\boldsymbol{I}^0)\boldsymbol{I}^1\right). \end{aligned}$$

Since $\boldsymbol{I}^0$, and $\boldsymbol{I}^1$ are linearly independent, we have

$$\frac{\partial}{\partial \boldsymbol{q}}(g^0+g^1) = 0 \iff \alpha_0(\boldsymbol{q}\cdot\boldsymbol{I}^0)^2 - \alpha_1(\boldsymbol{q}\cdot\boldsymbol{I}^1)^2 = 0.$$

Hence, $g^0 + g^1$ has a critical point on the simplex Δ_{n-1} if and only if

$$\alpha_0(\boldsymbol{q} \cdot \boldsymbol{I}^0)^2 = \alpha_1(\boldsymbol{q} \cdot \boldsymbol{I}^1)^2. \tag{3.19}$$

Notice that, if α_0, and α_1 have opposite signs, then equation (3.19) can not be satisfied for any $\boldsymbol{q} \in \Delta_{n-1}$, and hence, $g^0 + g^1$ has no critical point on the simplex Δ_{n-1}.

If $\alpha_0, \alpha_1 > 0$, then equation (3.19) leads to

$$\boldsymbol{q} \cdot (\sqrt{\alpha_0}\boldsymbol{I}^0 - \sqrt{\alpha_1}\boldsymbol{I}^1) = 0.$$

Therefore, $g^0 + g^1$ has a critical point on the simplex Δ_{n-1} if and only if there exists $v \in \Delta_{n-1}$ perpendicular to the vector $\sqrt{\alpha_0}\boldsymbol{I}^0 - \sqrt{\alpha_1}\boldsymbol{I}^1$. □

Lemma 3.13 (***Optimal number of sensors***)**.** For $n > M$, if each $(M+1) \times (M+1)$ sub-matrix of the matrix

$$\Gamma = \begin{bmatrix} \boldsymbol{T} & -\boldsymbol{I}^0 & \dots & -\boldsymbol{I}^{M-1} \end{bmatrix} \in \mathbb{R}^{n \times (M+1)}$$

is full rank, then the following statements hold:

(i). every solution of the optimization problem (3.17) lies on the probability simplex $\Delta_{M-1} \subset \Delta_{n-1}$; and

(ii). every time-optimal policy requires at most M sensors to be observed.

Proof. From Lemma 3.11, we know that if $\boldsymbol{T}, \boldsymbol{I}^0, \ldots, \boldsymbol{I}^{M-1}$ are linearly independent, then the Jacobian of the objective function in equation (3.17) does not vanish anywhere on the simplex Δ_{n-1}. Hence, a minimum lies at some simplex Δ_{n-2}, which is the boundary of the simplex Δ_{n-1}. Notice that, if $n > M$ and the condition in the lemma holds, then the projections of $\boldsymbol{T}, \boldsymbol{I}^0, \ldots, \boldsymbol{I}^{M-1}$ on the simplex Δ_{n-2} are also linearly independent, and the argument repeats. Hence, a minimum lies at some simplex Δ_{M-1}, which implies that optimal policy requires at most M sensors to be observed. □

Lemma 3.14 (***Optimization on an edge***)**.** Given two vertices $\boldsymbol{e}_s$ and $\boldsymbol{e}_r$, $s \neq r$, of the probability simplex Δ_{n-1}, then for the objective function in the problem (19) with $M = 2$, the following statements hold:

(i). if $g^0(\boldsymbol{e}_s) < g^0(\boldsymbol{e}_r)$, and $g^1(\boldsymbol{e}_s) < g^1(\boldsymbol{e}_r)$, then the minima, along the edge joining $\boldsymbol{e}_s$ and $\boldsymbol{e}_r$, lies at $\boldsymbol{e}_s$, and optimal value is given by $\frac{1}{2}(g^0(\boldsymbol{e}_s)+g^1(\boldsymbol{e}_s))$; and

(ii). if $g^0(\boldsymbol{e}_s) < g^0(\boldsymbol{e}_r)$, and $g^1(\boldsymbol{e}_s) > g^1(\boldsymbol{e}_r)$, then the minima, along the edge joining $\boldsymbol{e}_s$ and $\boldsymbol{e}_r$, lies at the point $\boldsymbol{q}^* = (1-\nu^*)\boldsymbol{e}_s + \nu^*\boldsymbol{e}_r$, where

$$\nu^* = \min\left\{1, \left(\frac{1}{1+\mu}\right)^+\right\},$$
$$\mu = \frac{I_r^0\sqrt{T_sI_r^1 - T_rI_s^1} - I_r^1\sqrt{T_rI_s^0 - T_sI_r^0}}{I_s^1\sqrt{T_rI_s^0 - T_sI_r^0} - I_s^0\sqrt{T_sI_r^1 - T_rI_s^1}},$$

and the optimal value is given by

$$\begin{cases} \frac{1}{2}(g^0(\boldsymbol{e}_s) + g^1(\boldsymbol{e}_s)), & \text{if } \nu^* = 0, \\ \frac{1}{2}(g^0(\boldsymbol{e}_r) + g^1(\boldsymbol{e}_r)), & \text{if } \nu^* = 1, \\ \frac{1}{2}\left(\sqrt{\frac{T_s I_r^1 - T_r I_s^1}{I_s^0 I_r^1 - I_r^0 I_s^1}} + \sqrt{\frac{T_r I_s^0 - T_s I_r^0}{I_s^0 I_r^1 - I_r^0 I_s^1}}\right)^2, & \text{otherwise.} \end{cases}$$

Proof. We observe from Lemma 5 that both g^0, and g^1 are monotonically non-increasing or non-decreasing along any line. Hence, if $g^0(\boldsymbol{e}_s) < g^0(\boldsymbol{e}_r)$, and $g^1(\boldsymbol{e}_s) < g^1(\boldsymbol{e}_r)$, then the minima should lie at $\boldsymbol{e}_s$. This concludes the proof of the first statement. We now establish the second statement. We note that any point on the line segment connecting $\boldsymbol{e}_s$ and $\boldsymbol{e}_r$ can be written as $\boldsymbol{q}(\nu) = (1-\nu)\boldsymbol{e}_s + \nu\boldsymbol{e}_r$. The value of $g^0 + g^1$ at $\boldsymbol{q}$ is

$$g^0(\boldsymbol{q}(\nu)) + g^1(\boldsymbol{q}(\nu)) = \frac{(1-\nu)T_s + \nu T_r}{(1-\nu)I_s^0 + \nu I_r^0} + \frac{(1-\nu)T_s + \nu T_r}{(1-\nu)I_s^1 + \nu I_r^1}.$$

Differentiating with respect to ν, we get

$$g^{0\prime}(\boldsymbol{q}(\nu)) + g^{1\prime}(\boldsymbol{q}(\nu)) = \frac{I_s^0 T_r - T_s I_r^0}{(I_s^0 + \nu(I_r^0 - I_s^0))^2} + \frac{I_s^1 T_r - T_s I_r^1}{(I_s^1 + \nu(I_r^1 - I_s^1))^2}. \tag{3.20}$$

Notice that the two terms in equation (3.20) have opposite sign. Setting the derivative to zero, and choosing the value of ν in $[0, 1]$, we get $\nu^* = \min\{1, 1/(1+\mu)^+\}$, where μ is as defined in the statement of the theorem. The optimal value of the function can be obtained, by substituting $\nu = \nu^*$ in the expression for $\frac{1}{2}(g^0(\boldsymbol{q}(\nu)) + g^1(\boldsymbol{q}(\nu)))$. □

Theorem 3.15 (***Optimization of average decision time***)**.** For the optimization problem (3.17) with $M = 2$, the following statements hold:

(i). if $\boldsymbol{I}^0$, $\boldsymbol{I}^1$ are linearly dependent, then the solution lies at some vertex of the simplex Δ_{n-1};

(ii). if $\boldsymbol{I}^0$ and $\boldsymbol{I}^1$ are linearly independent, and $\boldsymbol{T} = \alpha_0 \boldsymbol{I}^0 + \alpha_1 \boldsymbol{I}^1$, $\alpha_0, \alpha_1 \in \mathbb{R}$, then the following statements hold:

  (a) if α_0 and α_1 have opposite signs, then the optimal solution lies at some edge of the simplex Δ_{n-1};

  (b) if $\alpha_0, \alpha_1 > 0$, then the optimal solution may lie in the interior of the simplex Δ_{n-1};

(iii). if every 3×3 sub-matrix of the matrix $\begin{bmatrix} \boldsymbol{T} & \boldsymbol{I}^0 & \boldsymbol{I}^1 \end{bmatrix} \in \mathbb{R}^{n \times 3}$ is full rank, then a minimum lies at an edge of the simplex Δ_{n-1}.

Proof. We start by establishing the first statement. Since, $\boldsymbol{I}^0$ and $\boldsymbol{I}^1$ are linearly dependent, there exists a $\gamma > 0$ such that $\boldsymbol{I}^0 = \gamma \boldsymbol{I}^1$. For $\boldsymbol{I}^0 = \gamma \boldsymbol{I}^1$, we have $g^0 + g^1 = (1 + \gamma) g^0$. Hence, the minima of $g^0 + g^1$ lies at the same point where g^0 achieves the minima. From Theorem 3.6, it follows that g^0 achieves the minima at some vertex of the simplex Δ_{n-1}.

To prove the second statement, we note that from Lemma 3.12, it follows that if α_0, and α_1 have opposite signs, then the Jacobian of $g^0 + g^1$ does not vanish

anywhere on the simplex Δ_{n-1}. Hence, the minima lies at the boundary of the simplex. Notice that the boundary, of the simplex Δ_{n-1}, are n simplices Δ_{n-2}. Notice that the argument repeats till $n > 2$. Hence, the optima lie on one of the $\binom{n}{2}$ simplices Δ_1, which are the edges of the original simplex. Moreover, we note that from Lemma 3.12, it follows that if $\alpha_0, \alpha_1 > 0$, then we can not guarantee the number of optimal set of sensors. This concludes the proof of the second statement.

To prove the last statement, we note that it follows immediately from Lemma 3.13 that a solution of the optimization problem in equation (3.17) would lie at some simplex Δ_1, which is an edge of the original simplex. □

Note that, we have shown that, for $M = 2$ and a generic set of sensors, the solution of the optimization problem in equation (3.17) lies at an edge of the simplex Δ_{n-1}. The optimal value of the objective function on a given edge was determined in Lemma 3.14. Hence, an optimal solution of this problem can be determined by a comparison of the optimal values at each edge.

For the multiple hypothesis case, we have determined the time-optimal number of the sensors to be observed in Lemma 3.13. In order to identify these sensors, one needs to solve the optimization problem in equation (3.17). We notice that the objective function in this optimization problem is non-convex, and is hard to tackle analytically for $M > 2$. Interested reader may refer to some efficient

iterative algorithms in linear-fractional programming literature (e.g., [8]) to solve these problems.

3.3 Numerical Illustrations

We now elucidate on the results obtained in previous sections through some numerical examples. We present three examples, which provide further insights into the scenarios considered in Section 3.2. In the first one, we consider four sensors with ternary outputs, and three hypotheses. We compare the conditioned asymptotic decision times, obtained in Theorem 3.5, with the decision times obtained numerically through Monte-Carlo simulations. In the second example, for the same set of sensors and hypothesis, we compare the optimal average decision time, obtained in Theorem 3.15, with some particular average decision times. In the third example, we compare the worst case optimal decision time obtained in Theorem 3.10 with some particular worst-case expected decision times.

Example 3.1 (***Conditional expected decision time***)**.** We consider four sensors connected to a fusion center, and three underlying hypothesis. We assume that the sensors take ternary measurements $\{0, 1, 2\}$. The probabilities of their measurement being zero and one, under three hypotheses, are randomly chosen and are shown in Tables 3.1 and 3.2, respectively. The probability of measurement

being two is obtained by subtracting these probabilities from one. The processing times on the sensors are randomly chosen to be 0.68, 3.19, 5.31, and 6.55 seconds, respectively.

Table 3.1: Conditional probabilities of measurement being zero

Sensor	Probability(0)		
	Hypothesis 0	Hypothesis 1	Hypothesis 2
1	0.4218	0.2106	0.2769
2	0.9157	0.0415	0.3025
3	0.7922	0.1814	0.0971
4	0.9595	0.0193	0.0061

Table 3.2: Conditional probabilities of measurement being one

Sensor	Probability(1)		
	Hypothesis 0	Hypothesis 1	Hypothesis 2
1	0.1991	0.6787	0.2207
2	0.0813	0.7577	0.0462
3	0.0313	0.7431	0.0449
4	0.0027	0.5884	0.1705

We performed Monte-Carlo simulations to numerically obtain the expected decision time, conditioned on hypothesis H_0. For different sensor selection probabilities, a comparison of the numerically obtained expected decision times with the theoretical expected decision times is shown in Figure 3.3. These results suggest that the asymptotic decision times obtained in Theorem 3.5 provide a lower bound to the conditional expected decision times for the larger error probabilities.

It can be seen from Figure 3.3, and verified from Theorem 3.6 that conditioned on hypothesis H_0, sensor 4 is the optimal sensor. Notice the processing time and information trade-off. Despite having the highest processing time, conditioned on hypothesis H_0, the sensor 4 is optimal. This is due to the fact that sensor 4 is highly informative on hypothesis H_0.

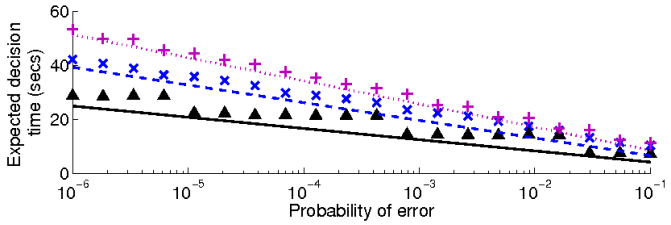


Figure 3.3: Expected decision time conditioned on hypothesis H_0 plotted on semi-log axes. The dotted magenta line and magenta "+" represent the theoretical and numerical expected decision time for average expected decision time-optimal sensor selection policy, respectively. The dashed blue line and blue "×" represent the theoretical and numerical expected decision time for the uniform sensor selection policy, respectively. The solid black line and black triangles represent the theoretical and numerical expected decision time when only optimal sensor 4 is selected.

Example 3.2 (***Optimal average expected decision time***)**.** For the same set of data in Example 1, we now determine the optimal policies for the average expected decision time. A comparison of average expected decision time for different sensor selection probabilities is shown in Figure 3.4. An optimal average expected decision time sensor selection probability distribution is $\boldsymbol{q} = [0\ 0.98\ 0\ 0.02]$. It can be seen that the optimal policy significantly improves the average expected decision time over the uniform policy. The sensor 4 which is the optimal sensor conditioned on hypothesis H_0 is now chosen with a very small probability. This is

due to the poor performance of the sensor 4 under hypothesis H_1 and H_2 and its high processing time. Good performance under one hypothesis and poor performance under other hypothesis is common in weather-sensitive sensors, e.g., sensor performs extremely well in sunny conditions, but in cloudy or rainy conditions its performance deteriorates significantly.

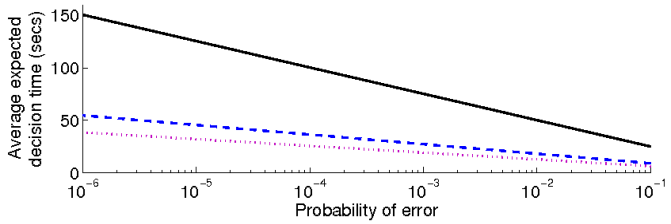


Figure 3.4: Average expected decision times plotted on semi-log axes. The black solid line represents the policy where only sensor 4 is selected. The blue dashed line represents the uniform sensor selection policy. The magenta dotted line is average expected decision time-optimal policy.

Example 3.3 (***Optimal worst case decision time***)**.** For the same set of data in Example 1, we now determine the optimal policies for the average expected decision time. For this data, the optimal worst-case sensor selection probability distribution is $\boldsymbol{q} = [0 \;\; 0.91 \;\; 0 \;\; 0.09]$. A comparison of the optimal worst case expected decision time with some particular worst case decision times is shown in Figure 3.5. It may be verified that for the optimal sensor selection probabilities, the expected decision time, conditioned on hypothesis H_0 and H_2 are the same. This suggests that even for more that two hypothesis, the optimal policy may lie at the intersection of the graphs of the expected decision times.

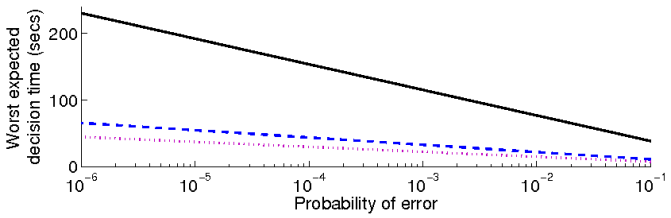


Figure 3.5: Worst case expected decision times plotted on semi-log axes. The black solid line represents the policy where only sensor 4 is selected. The blue dashed line represents the uniform sensor selection policy. The magenta dotted line is worst expected decision time-optimal policy.

Remark 3.7. The optimal results we obtained, may only be sub-optimal because of the asymptotic approximations in equations (3.3). We further note that, for small error probabilities and large sample sizes, these asymptotic approximations yield fairly accurate results [7], and in fact, this is the regime in which it is of interest to minimize the expected decision time. Therefore, for all practical purposes the obtained optimal scheme is very close to the actual optimal scheme. □

3.4 Conclusions

In this chapter, we considered a sequential decision making problem with randomized sensor selection. We developed a version of the MSPRT algorithm where the sensor switches at each observation. We used this sequential procedure to decide reliably. We studied the set of optimal sensors to be observed in order to decide in minimum time. We observed the trade off between the information car-

ried by a sensor and its processing time. A randomized sensor selection strategy was adopted. It was shown that, conditioned on a hypothesis, only one sensor is optimal. Indeed, if the true hypothesis is not known beforehand, then a randomized strategy is justified. For the binary hypothesis case, three performance metrics were considered and it was found that for a generic set of sensors at most two sensors are optimal. Further, it was shown that for M underlying hypotheses, and a generic set of sensors, an optimal policy requires at most M sensors to be observed. It was observed that the optimal set of the sensors is not necessarily the set of optimal sensors conditioned on each hypothesis. A procedure for the identification of the optimal sensors was developed. In the binary hypothesis case, the computational complexity of the procedure for the three scenarios, namely, the conditioned decision time, the worst case decision time, and the average decision time, was $\mathcal{O}(n)$, $\mathcal{O}(n^2)$, and $\mathcal{O}(n^2)$, respectively.

Chapter 4

Stochastic Surveillance Strategies for Spatial Quickest Detection

In this chapter we consider the *persistent surveillance* of a set of n disjoint regions with a team of $m < n$ identical[1] autonomous vehicles capable of sensing, communicating, and moving from one region to another. In persistent surveillance, the vehicles visit the regions according to some routing policy, collect evidence (sensor observation), and send it to a control center. The control center runs an anomaly detection algorithm with the evidence collected by the vehicles to determine the likelihood of an anomaly being present at some region (the control center declares an anomaly if substantial evidence is present). Finally, the control center utilizes the likelihood of an anomaly at each region to determine a vehicle routing policy. The objective of the control center is to detect an anomaly at any region in minimum time subject to a desired bound on the expected time

[1]The vehicle routing policies designed in this chapter also work for non-identical vehicles. We make this assumption for the convenience of analysis.

between any two subsequent false alarms. Notice that the time required to detect an anomaly depends on the anomaly detection algorithm and the time vehicles take to travel the regions. Thus, the control center needs to minimize the anomaly detection time jointly over anomaly detection policies and vehicle routing policies. Our problem setup is shown in Fig. 4.1.

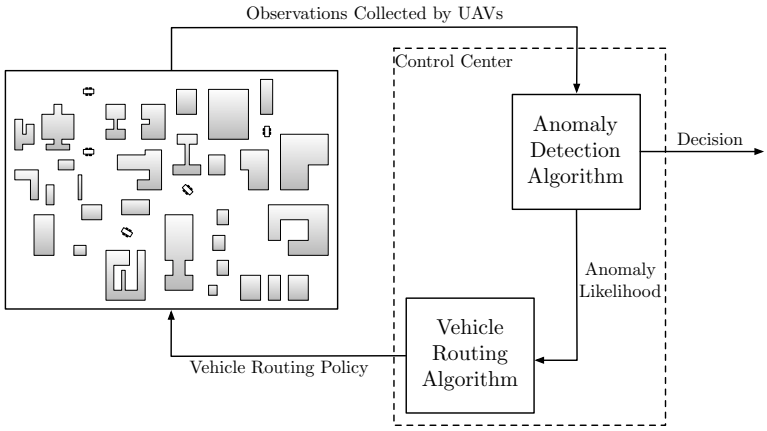


Figure 4.1: Persistent Surveillance Setup. A set of n regions is surveyed by $m < n$ vehicles. Each vehicle visits the regions according to some policy and collects evidence from the visited region. The collected evidence is sent to an anomaly detection algorithm. The anomaly detection algorithm processes the collected evidence and decides on the presence of an anomaly. It also provides the likelihood of an anomaly being present, which in turn is used by the vehicle routing algorithm. The anomaly detection algorithm and vehicle routing algorithm constitute the control center, which can be implemented on-board of a vehicle.

We adopt the standard motion planning notation in [59]. We denote the k-th region by $\mathcal{R}_k, k \in \{1, \ldots, n\}$, and r-th vehicle by $\mathcal{U}_r, r \in \{1, \ldots, m\}$. Let the likelihood of an anomaly at region $\mathcal{R}_k$ be $\pi_k \in (0, 1)$. We study the persistent surveillance problem under the following assumptions.

Regarding the vehicles, we do not assume any specific dynamics and we assume that:

(A1). each vehicle takes time d_{ij} to travel from region $\mathcal{R}_i$ to region $\mathcal{R}_j$, $i, j \in \{1, \ldots, n\}$;

(A2). the sensors on each vehicle take a random time T_k to collect an informative observation[2] from region $\mathcal{R}_k, k \in \{1, \ldots, n\}$.

Regarding the observations, we assume that:

(A3). the observation collected by a vehicle from region $\mathcal{R}_k$ is sampled from probability density functions $f_k^0 : \mathbb{R} \to \mathbb{R}_{\geq 0}$ and $f_k^1 : \mathbb{R} \to \mathbb{R}_{\geq 0}$, respectively, in the presence and in the absence of anomalies;

(A4). for each $k \in \{1, \ldots, n\}$, probability density functions f_k^1 and f_k^0 are non-identical with some non-zero probability, and have the same support;

(A5). conditioned on the presence or absence of anomalies, the observations in each region are mutually independent, and

(A6). observations in different regions are also mutually independent.

[2]An informative observation may require the acquisition of several observations from different locations at the same region. In this case the processing time equals the total time required to collect all these observations.

Regarding the anomaly detection algorithm at the control center, we employ the cumulative sum (CUSUM) algorithm (see Section 2.2) for anomaly detection at each region. In particular, we run n parallel CUSUM algorithms (one for each region) and declare an anomaly being present at a region as soon as a substantial evidence is present. We refer to such parallel CUSUM algorithms by *ensemble CUSUM algorithm*.

Remark 4.1 (***Knowledge of distributions***)**.** For the ease of presentation, we assume that the probability density functions in presence and absence of an anomaly are known. In general, only the probability density function in absence of any anomaly may be known, or both the probability density functions may be unknown. In the first case, the CUSUM algorithm can be replaced by the weighted CUSUM algorithm or the Generalized Likelihood Ratio (GLR) algorithm, see [6], while in the second case, it can be replaced by the robust minimax quickest change detection algorithm proposed in [101]. The ideas presented in this chapter extend to these cases in a straightforward way. A related example is in Section 4.4. □

Remark 4.2 (***Independence of observations***)**.** For the ease of presentation, we assume that the observations collected from each region are independent conditioned on the presence and absence of anomalies. In general, the observations may be dependent and the dependence can be captured through an appropriate hidden Markov model. If the observations can be modeled as a hidden Markov

model, then the CUSUM like algorithm in [24] can be used instead of the standard CUSUM algorithm. The analysis presented in this chapter holds in this case as well but in an asymptotic sense, i.e., in the limit when a large number of observations are needed for anomaly detection.

We also assumed that the observations collected from different regions are mutually independent. Although the ideas in this chapter also work when the observations at different regions are dependent, the performance can be improved with a slight modification in the procedure presented here (see Remark 4.4). In this case the algorithm performance improves because each observation is now informative about more than one region. □

Regarding the vehicle routing policy, we propose the *randomized routing policy*, and the *adaptive routing policy*. In the randomized routing policy, each vehicle (i) selects a region from a stationary distribution, (ii) visits that region, (iii) collects evidence, and (iv) transmits this evidence to the control center and iterates this process endlessly. In the randomized routing policy, the evidence collected by the vehicles is not utilized to modify their routing policy. In other words, there is no feedback from the anomaly detection algorithm to the vehicle routing algorithm. In the adaptive routing policy, instead, the evidence collected by the vehicles is used to modify the routing policy, and thus, the loop between the vehicle routing algorithm and the anomaly detection algorithm is closed. The adaptive

routing policy follows the same steps as in the randomized routing policy, with the exception that the distribution in step (i) is no longer stationary and is adapted based on the collected evidence.

For brevity of notation, we will refer to the joint anomaly detection and vehicle routing policy comprising of the ensemble CUSUM algorithm and the randomized routing policy by *randomized ensemble CUSUM algorithm.* We will show that the randomized ensemble CUSUM algorithm provides a solution that is within a factor of optimality. Similarly, we refer to the joint anomaly detection and vehicle routing policy comprising of the ensemble CUSUM algorithm and adaptive routing policy by *adaptive ensemble CUSUM algorithm.* We will show that adaptive ensemble CUSUM algorithm makes the vehicles visit anomalous regions with high probability, and thus it improves upon the performance of the randomized ensemble CUSUM algorithm.

Remark 4.3 (***Randomized routing policy***)**.** The randomized routing policy samples regions to visit from a stationary distribution; this assumes that each region can be visited from another region in a single hop. While this is the case for aerial vehicles, it may not be true for ground vehicles. In the latter case, the motion from one region to another can be modeled as a Markov chain. The transition probabilities of this Markov chain can be designed to achieve a desired stationary distribution. This can optimally be done, for instance, by picking the

fastest mixing Markov chain proposed in [14] or heuristically by using the standard Metropolis-Hastings algorithm, see [105]. Related examples are presented in Section 4.4 and 4.5. It should be noted that under the randomized routing policy, the desired stationary distribution of the Markov chain is fixed, and the Markov chain converges to this distribution exponentially. Thus, the policy designed using Markov chain is arbitrarily close to the desired policy. However, in the case of adaptive routing policy, the desired stationary distribution keeps on changing, and the performance of the Markov chain based policy depends on rate of convergence of the Markov chain and the rate of change of desired stationary distribution. □

We now introduce some notations that will be used throughout the chapter. We denote the probability simplex in $\mathbb{R}^n$ by Δ_{n-1}, and the space of vehicle routing policies by Ω. For the processing time T_k, we let $\bar{T}_k$ denote its expected value. Consider m realizations of the processing time T_k, we denote the expected value of the minimum of these m realized values by $\bar{T}_k^{m\text{-smlst}}$. Note that $\bar{T}_k^{1\text{-smlst}} = \bar{T}_k$. We also define $\bar{T}_{\max} = \max\{\bar{T}_k \mid k \in \{1, \ldots, n\}\}$ and $\bar{T}_{\min} = \min\{\bar{T}_k \mid k \in \{1, \ldots, n\}\}$. We denote the Kullback-Leibler divergence between the probability density functions f_k^1 and f_k^0 by $\mathcal{D}_k$. Finally, $\mathcal{D}_{\max} = \max\{\mathcal{D}_k \mid k \in \{1, \ldots, n\}\}$ and $\mathcal{D}_{\min} = \min\{\mathcal{D}_k \mid k \in \{1, \ldots, n\}\}$.

4.1 Spatial Quickest Detection

In this section, we propose the ensemble CUSUM algorithm for the simultaneous quickest detection of anomalies in spatially distributed regions.

4.1.1 Ensemble CUSUM algorithm

We run n parallel CUSUM algorithms (one for each region), and update the CUSUM statistic for region $\mathcal{R}_k$ only if an observation is received from region $\mathcal{R}_k$. We refer to such parallel CUSUM algorithms by *ensemble CUSUM algorithm* (Algorithm 4.1). Notice that an iteration of this algorithm is initiated by the collection of an observation.

We are particularly interested in the performance of the ensemble CUSUM algorithm when the observations are collected by autonomous vehicles. In this case, the performance of the ensemble CUSUM algorithm is a function of the vehicle routing policy. For the ensemble CUSUM algorithm with autonomous vehicles collecting observation, let the number of iterations (collection of observations) required to detect an anomaly at region $\mathcal{R}_k$ be $N_k : \Omega \to \mathbb{N} \cup \{+\infty\}$, and let the detection delay, i.e., the time required to detect an anomaly, at region $\mathcal{R}_k$ be $\delta_k : \Omega \to \mathbb{R}_{>0} \cup \{+\infty\}$, for each $k \in \{1, \ldots, n\}$, where Ω is the space of vehicle routing policies. We also define average detection delay as follows:

Algorithm 4.1: *Ensemble CUSUM Algorithm*

Input : threshold η, pdfs $f_k^0, f_k^1, k \in \{1, \ldots, n\}$;

Output : decision on presence of an anomaly ;

1 at time τ receive observation y_τ for region $\mathcal{R}_k$;

2 update the CUSUM statistic at each region:

$$\Lambda_\tau^j = \begin{cases} \left(\Lambda_{\tau-1}^k + \log \frac{f_k^1(y_\tau)}{f_k^0(y_\tau)}\right)^+, & \text{if } j = k; \\ \Lambda_{\tau-1}^j, & \text{if } j \in \{1, \ldots, n\} \setminus \{k\}; \end{cases}$$

3 **if** $\Lambda_\tau^k > \eta$ **then** change detected at region $\mathcal{R}_k$;

4 **else** wait for next observations and iterate.

Definition 4.1 (***Average detection delay***)**.** For any vector of weights $(w_1, \ldots, w_n) \in \Delta_{n-1}$, define the average detection delay $\delta_{\text{avg}} : \Omega \to \mathbb{R}_{>0} \cup \{+\infty\}$ for the ensemble CUSUM algorithm with autonomous vehicles collecting observations by

$$\delta_{\text{avg}}(\omega) = \sum_{k=1}^{n} w_k \mathbb{E}[\delta_k(\omega)]. \tag{4.1}$$

For the ensemble CUSUM algorithm with m vehicles collecting observation, define $\delta_k^{m\text{-min}}$ and $\delta_{\text{avg}}^{m\text{-min}}$ by

$$\delta_k^{m\text{-min}} = \inf\{\mathbb{E}[\delta_k(\omega)] \mid \omega \in \Omega\}, \text{ and}$$

$$\delta_{\text{avg}}^{m\text{-min}} = \inf\{\delta_{\text{avg}}(\omega) \mid \omega \in \Omega\},$$

respectively. Note that $\delta_k^{m\text{-min}}$ and $\delta_{\text{avg}}^{m\text{-min}}$ are lower bounds for the expected detection delay and average detection delay at region $\mathcal{R}_k$, respectively, independently of the routing policy. Let $\bar{\eta} = e^{-\eta} + \eta - 1$. We now state lower bounds on the performance of the ensemble CUSUM algorithm with autonomous vehicles collecting observations.

Lemma 4.1 (***Global lower bound***)**.** The following statements hold for the ensemble CUSUM algorithm with m vehicles collecting information:

(i). the lower bound $\delta_k^{m\text{-min}}$ for the expected detection delay at region $\mathcal{R}_k$ satisfies

$$\delta_k^{m\text{-min}} \geq \frac{\bar{\eta}\, \bar{T}_k^{m\text{-smlst}}}{m\mathcal{D}_k};$$

(ii). the lower bound $\delta_{\text{avg}}^{m\text{-min}}$ for the average detection delay satisfies

$$\delta_{\text{avg}}^{m\text{-min}} \geq \frac{\bar{\eta}\, \bar{T}_{\text{min}}^{m\text{-smlst}}}{m\mathcal{D}_{\max}},$$

where $\bar{T}_{\text{min}}^{m\text{-smlst}} = \min\{\bar{T}_k^{m\text{-smlst}} \mid k \in \{1, \ldots, n\}\}$.

Proof. We start by establishing the first statement. We note that a lower bound on the expected detection delay at region $\mathcal{R}_k$ is obtained if all the vehicles always stay at region $\mathcal{R}_k$. Since, each observation is collected from region $\mathcal{R}_k$, the number of iterations of the ensemble CUSUM algorithm required to detect an anomaly at region $\mathcal{R}_k$ satisfies $\mathbb{E}[N_k] = \bar{\eta}/\mathcal{D}_k$. Let $T_k^r(b)$ be realized value of the processing time of vehicle $\mathcal{U}_r$ at its b-th observation. It follows that

$T_k^{m\text{-smlst}}(b) = \min\{T_k^r(b) \mid r \in \{1,\dots,m\}\}$ is a lower bound on the processing time of each vehicle for its b-th observation. Further, $T_k^{m\text{-smlst}}(b)$ is identically distributed for each b and $\mathbb{E}[T_k^{m\text{-smlst}}(b)] = \bar{T}_k^{m\text{-smlst}}$. Consider a modified stochastic process where the realized processing time of each vehicle for its b-th observation in $T_k^{m\text{-smlst}}(b)$. Indeed, such a stochastic process underestimates the time required to collect each observation and, hence, provides a lower bound to the expected detection delay. Therefore, the detection delay satisfies the following bound

$$\delta_k(\omega) \geq \sum_{b=1}^{\lceil N_k/m \rceil} T_k^{m\text{-smlst}}(b), \text{ for each } \omega \in \Omega.$$

It follows from Wald's identity, see [76], that

$$\mathbb{E}[\delta_k(\omega)] \geq \bar{T}_k^{m\text{-smlst}} \mathbb{E}[\lceil N_k/m \rceil] \geq \bar{T}_k^{m\text{-smlst}} \mathbb{E}[N_k]/m.$$

This proves the first statement.

The second statement follows from Definition 4.1 and the first statement. □

Remark 4.4 (***Dependence across regions***)**.** We assumed that the observations collected from different regions are mutually independent. If the observations from different regions are dependent, then, at each iteration, instead of updating only one CUSUM statistic, the CUSUM statistic at each region should be updated with an appropriate marginal distribution. □

4.2 Randomized Ensemble CUSUM Algorithm

We now study the persistent surveillance problem under randomized ensemble CUSUM algorithm. First, we derive an exact expression for the expected detection delay for the randomized ensemble CUSUM algorithm with a single vehicle, and use the derived expressions to develop an efficient stationary policy for a single vehicle. Second, we develop a lower bound on the expected detection delay for the randomized ensemble CUSUM algorithm with multiple vehicles, and develop a generic partitioning policy that (i) constructs a complete and disjoint m-partition of the regions, (ii) allocates one partition each to a vehicle, and (iii) lets each vehicle survey its assigned region with some single vehicle policy. Finally, we show that the partitioning policy where each vehicle implements the efficient stationary policy in its regions is within a factor of an optimal policy.

4.2.1 Analysis for single vehicle

Consider the randomized ensemble CUSUM algorithm with a single vehicle. Let $q_k \in [0, 1]$ denote the probability to select region $\mathcal{R}_k$, and let $\boldsymbol{q} = (q_1, \ldots, q_n) \in \Delta_{n-1}$. Let the threshold for the CUSUM algorithm at each region be uniform and equal to $\eta \in \mathbb{R}_{>0}$. We note that for the randomized ensemble CUSUM algorithm with a single vehicle the space of vehicle routing policies is $\Omega = \Delta_{n-1}$.

Theorem 4.2 (***Single vehicle randomized ensemble CUSUM***)**.** For the randomized ensemble CUSUM algorithm with a single vehicle and stationary routing policy $\boldsymbol{q} \in \Delta_{n-1}$, the following statements hold:

(i). the number of observations $N_k(\boldsymbol{q})$ required to detect a change at region $\mathcal{R}_k$ satisfies

$$\mathbb{E}_{f_k^1}[N_k(\boldsymbol{q})] = \frac{\bar{\eta}}{q_k \mathcal{D}_k};$$

(ii). the detection delay $\delta_k(\boldsymbol{q})$ at region $\mathcal{R}_k$ satisfies

$$\mathbb{E}_{f_k^1}[\delta_k(\boldsymbol{q})] = \Big(\sum_{i=1}^{n} q_i \bar{T}_i + \sum_{i=1}^{n} \sum_{j=1}^{n} q_i q_j d_{ij} \Big) \mathbb{E}_{f_k^1}[N_k(\boldsymbol{q})].$$

Proof. Let $\tau \in \{1, \ldots, N_k\}$ be the iterations at which the vehicle collects and sends information about the regions, where N_k denotes the iteration at which an anomaly is detected at region $\mathcal{R}_k$. Let the log likelihood ratio at region $\mathcal{R}_k$ at iteration τ be λ_τ^k. We have

$$\lambda_\tau^k = \begin{cases} \log \frac{f_k^1(y_\tau)}{f_k^0(y_\tau)}, & \text{with probability } q_k, \\ 0, & \text{with probability } 1 - q_k. \end{cases}$$

Therefore, conditioned on the presence of an anomaly, $\{\lambda_\tau^k\}_{\tau \in \mathbb{N}}$ are i.i.d., and

$$\mathbb{E}_{f_k^1}[\lambda_\tau^k] = q_k \mathcal{D}_k.$$

The remaining proof of the first statement follows similar to the proof for CUSUM in [85].

To prove the second statement, note that the information aggregation time T^{agr} comprises of the processing time and the travel time. At an iteration the vehicle is at region $\mathcal{R}_i$ with probability q_i and picks region $\mathcal{R}_j$ with probability q_j. Additionally, the vehicle travels between the two regions in d_{ij} units of time. Thus, the average travel time at each iteration is

$$\mathbb{E}[T_{\text{travel}}] = \sum_{i=1}^{n} \sum_{j=1}^{n} q_i q_j d_{ij}.$$

Hence, the expected information aggregation time at each iteration is

$$\mathbb{E}[T^{\text{agr}}] = \mathbb{E}[T_{\text{travel}} + T_{\text{process}}] = \sum_{i=1}^{n} \sum_{j=1}^{n} q_i q_j d_{ij} + \sum_{i=1}^{n} q_i \bar{T}_i.$$

Let $\{T_\tau^{\text{agr}}\}_{\tau \in \{1,\dots,N_k\}}$, be the information aggregation times at each iteration. We have that $\delta_k = \sum_{\tau=1}^{N_k} T_\tau^{\text{agr}}$, and it follows from Wald's identity, see [76], that

$$\mathbb{E}[\delta_k] = \mathbb{E}[T^{\text{agr}}]\mathbb{E}[N_k].$$

This completes the proof of the statement. □

4.2.2 Design for single vehicle

Our objective is to design a stationary policy that simultaneously minimizes the detection delay at each region, that is, to design a stationary policy that minimizes each term in $(\delta_1(\boldsymbol{q}), \dots, \delta_n(\boldsymbol{q}))$ simultaneously. For this multiple-objective

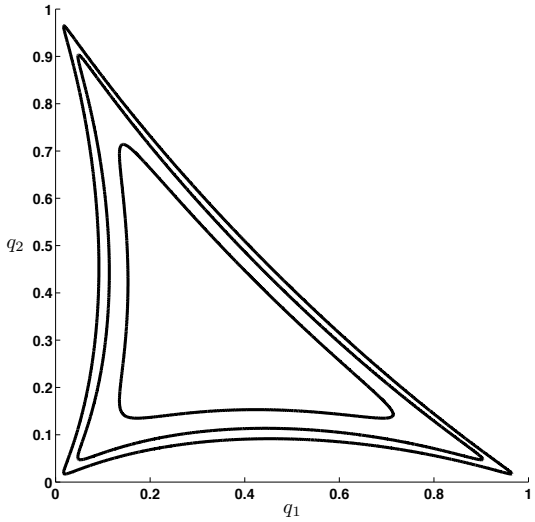


Figure 4.2: Level-sets of the objective function in problem (4.3). It can be seen that the level sets are not convex.

optimization problem, we construct a single aggregate objective function as the average detection delay. After incorporating the expressions for the expected detection delays derived in Theorem 4.2, the average detection delay becomes

$$\delta_{\text{avg}}(\boldsymbol{q}) = \Big(\sum_{k=1}^{n} \frac{w_k \bar{\eta}}{q_k \mathcal{D}_k} \Big) \Big(\sum_{i=1}^{n} q_i T_i + \sum_{i=1}^{n} \sum_{j=1}^{n} q_i q_j d_{ij} \Big), \tag{4.2}$$

where $w_k = \pi_k / (\sum_{i=1}^{n} \pi_i)$ is the weight on the expected detection delay at region $\mathcal{R}_k$ and π_k is the prior probability of an anomaly being present at region $\mathcal{R}_k$. Our objective is to solve the average detection delay minimization problem:

$$\underset{\boldsymbol{q} \in \Delta_{n-1}}{\text{minimize}} \quad \delta_{\text{avg}}(\boldsymbol{q}). \tag{4.3}$$

In general, the objective function δ_{avg} is non-convex. For instance, let $n = 3$, and consider the level sets of δ_{avg} on the two dimensional probability simplex

(Fig. 4.2). It can be seen that the level sets are non-convex, yet there exists a unique critical point and it corresponds to a minimum. We now state the following conjecture about the average detection delay:

Conjecture 4.3 (***Single vehicle optimal stationary policy***)**.** For the randomized ensemble CUSUM algorithm with a single vehicle, the average detection delay function δ_{avg} has a unique critical point at which the minimum of δ_{avg} is achieved. □

In the Appendix of this chapter we provide probabilistic guarantees that, for a particular stochastic model of the parameters in δ_{avg}, with at least confidence level 99.99% and probability at least 99%, the optimization problem (4.3) has a unique critical point at which the minimum is achieved. Such a minimum can be computed via standard gradient-descent methods, see [15].

We now construct an upper bound for the expected detection delay. We will show that minimization of this upper bound yields a policy that is within a factor of an optimal policy. From equation (4.2), we define the upper bound $\delta_{\text{upper}} : \Delta_{n-1} \to \mathbb{R}_{>0} \cup \{+\infty\}$ as

$$\delta_{\text{avg}}(\boldsymbol{q}) \leq \delta_{\text{upper}}(\boldsymbol{q}) = \Big(\sum_{k=1}^{n} \frac{w_k \bar{\eta}}{q_k \mathcal{D}_k} \Big) (\bar{T}_{\max} + d_{\max}),$$

where $d_{\max} = \max\{d_{ij} \mid i, j \in \{1, \ldots, n\}\}$.

Theorem 4.4 (***Single vehicle efficient stationary policy***)**.** The following statements hold for the randomized ensemble CUSUM algorithm with single vehicle:

(i). the upper bound on the expected detection delay satisfies

$$\min_{\boldsymbol{q}\in\Delta_{n-1}} \delta_{\text{upper}}(\boldsymbol{q}) = \Big(\sum_{k=1}^{n}\sqrt{\frac{w_k}{\mathcal{D}_k}}\Big)^2 \bar{\eta}(\bar{T}_{\max}+d_{\max}),$$

and the minimum is achieved at $\boldsymbol{q}^\dagger$ defined by

$$q_k^\dagger = \frac{\sqrt{w_k/\mathcal{D}_k}}{\sum_{j=1}^{n}\sqrt{w_j/\mathcal{D}_j}},\ k\in\{1,\ldots,n\};$$

(ii). the average detection delay satisfies the following lower bound

$$\delta_{\text{avg}}(\boldsymbol{q}) \geq \Big(\sum_{k=1}^{n}\sqrt{\frac{w_k}{\mathcal{D}_k}}\Big)^2 \bar{\eta}\,\bar{T}_{\min},$$

for all $\boldsymbol{q}\in\Delta_{n-1}$;

(iii). the stationary policy $\boldsymbol{q}^\dagger$ is within a factor of optimal, that is

$$\frac{\delta_{\text{avg}}(\boldsymbol{q}^\dagger)}{\delta_{\text{avg}}(\boldsymbol{q}^*)} \leq \frac{\bar{T}_{\max}+d_{\max}}{\bar{T}_{\min}},\ \text{and}$$
$$\frac{\delta_{\text{avg}}(\boldsymbol{q}^\dagger)}{\delta_{\text{avg}}^{\text{1-min}}} \leq n\frac{\bar{T}_{\max}+d_{\max}}{\bar{T}_{\min}}\frac{\mathcal{D}_{\max}}{\mathcal{D}_{\min}},$$

where $\boldsymbol{q}^*$ is an optimal stationary policy;

(iv). the expected detection delay at region $\mathcal{R}_k$ under policy $\boldsymbol{q}^\dagger$ satisfy

$$\frac{\mathbb{E}[\delta_k(\boldsymbol{q}^\dagger)]}{\delta_k^{\text{1-min}}} \leq \frac{(\bar{T}_{\max}+d_{\max})}{\bar{T}_k}\sqrt{\frac{n\mathcal{D}_k}{w_k\mathcal{D}_{\min}}}.$$

Proof. We start by establishing the first statement. It follows from the stationarity conditions on the Lagrangian that the minimizer $\boldsymbol{q}^\dagger$ of δ_{upper} satisfy $q_k^\dagger \propto \sqrt{w_k \bar{\eta}/\mathcal{D}_k}$, for each $k \in \{1, \ldots, n\}$. Incorporating this fact into $\sum_{k=1}^n q_k^\dagger = 1$ yields the expression for $q_k^\dagger$. The expression for $\delta_{\text{upper}}(\boldsymbol{q}^\dagger)$ can be verified by substituting the expression for $\boldsymbol{q}^\dagger$ into δ_{upper}.

To prove the second statement, we construct a lower bound $\delta_{\text{lower}} : \Delta_{n-1} \to \mathbb{R}_{>0} \cup \{+\infty\}$ to the average detection delay δ_{avg} defined by $\delta_{\text{lower}}(\boldsymbol{q}) = \sum_{k=1}^n w_k \bar{\eta} \bar{T}_{\min}/\mathcal{D}_k q_k$. It can be verified that δ_{lower} also achieves its minimum at $\boldsymbol{q}^\dagger$, and

$$\delta_{\text{lower}}(\boldsymbol{q}^\dagger) = \Big(\sum_{k=1}^n \sqrt{\frac{w_k}{\mathcal{D}_k}} \Big)^2 \bar{\eta}\ \bar{T}_{\min}.$$

We note that

$$\delta_{\text{lower}}(\boldsymbol{q}^\dagger) \leq \delta_{\text{lower}}(\boldsymbol{q}^*) \leq \delta_{\text{avg}}(\boldsymbol{q}^*) \leq \delta_{\text{avg}}(\boldsymbol{q}), \forall \boldsymbol{q} \in \Delta_{n-1}.$$

Thus, the second statement follows.

To prove the first part of the third statement, we note that

$$\delta_{\text{lower}}(\boldsymbol{q}^\dagger) \leq \delta_{\text{avg}}(\boldsymbol{q}^*) \leq \delta_{\text{avg}}(\boldsymbol{q}^\dagger) \leq \delta_{\text{upper}}(\boldsymbol{q}^\dagger).$$

Therefore, the policy $\boldsymbol{q}^\dagger$ is within $\delta_{\text{upper}}(\boldsymbol{q}^\dagger)/\delta_{\text{lower}}(\boldsymbol{q}^\dagger) = (T_{\max} + d_{\max})/T_{\min}$ factor of optimal stationary policy.

To prove the second part of the third statement, we note

$$\begin{aligned}\frac{\delta_{\text{avg}}(\boldsymbol{q}^\dagger)}{\delta_{\text{avg}}^{\text{1-min}}} &\leq \frac{\mathcal{D}_{\max}(\bar{T}_{\max}+d_{\max})}{\mathcal{D}_{\min}\bar{T}_{\min}}(\sqrt{w_1}+\ldots+\sqrt{w_n})^2\\ &\leq n\frac{\bar{T}_{\max}+d_{\max}}{\bar{T}_{\min}}\frac{\mathcal{D}_{\max}}{\mathcal{D}_{\min}},\end{aligned}$$

where the last inequality follows from the fact: $\max\{\sqrt{w_1}+\ldots+\sqrt{w_n} \mid w_1+\ldots+w_n=1\}=\sqrt{n}$.

To establish the last statement, we note that

$$\begin{aligned}\frac{\mathbb{E}[\delta_k(\boldsymbol{q}^\dagger)]}{\delta_k^{\text{1-min}}} &\leq \frac{(\bar{T}_{\max}+d_{\max})}{q_k^\dagger\bar{T}_k}\\ &\leq \frac{(\bar{T}_{\max}+d_{\max})}{\bar{T}_k}\sqrt{\frac{\mathcal{D}_k}{w_k\mathcal{D}_{\min}}}(\sqrt{w_1}+\ldots+\sqrt{w_n})\\ &\leq \frac{(\bar{T}_{\max}+d_{\max})}{\bar{T}_k}\sqrt{\frac{n\mathcal{D}_k}{w_k\mathcal{D}_{\min}}}.\end{aligned}$$

This concludes the proof of the theorem. □

In the following, we would refer to $\boldsymbol{q}^\dagger$ as the *single vehicle efficient stationary policy.*

Remark 4.5 (***Efficient stationary policy***)**.** As opposed to the average detection delay δ_{avg}, the upper bound δ_{upper} does not depend upon any travel time information. Then, our efficient policy does not take this information into account, and it may not be optimal. Instead, an optimal policy allocates higher visiting probabilities to regions that are located more centrally in the environment. We

resort to the efficient policy because (i) if the problem (4.3) does not admit a unique minimum, then the optimal policy can not be computed efficiently; and (ii) efficient policy has an intuitive, tractable, and closed form expression. □

4.2.3 Analysis for multiple vehicles

We now consider the randomized ensemble CUSUM with $m > 1$ vehicles. In this setting the vehicles operate in an asynchronous fashion. This asynchronicity, which did not occur in the single vehicle case, is due to (i) different travel times between two different pair of regions, and (ii) different realized value of processing time at each iteration. Such an asynchronous operation makes the time durations between two subsequent iterations non-identically distributed and makes it difficult to obtain closed form expressions for the expected detection delay at each region.

Motivated by the above discussion, we determine a lower bound on the expected detection delay for the randomized ensemble CUSUM algorithm with multiple vehicles. Let $\boldsymbol{q}^r = (q_1^r, \ldots, q_n^r) \in \Delta_{n-1}$ denote the *stationary policy* for vehicle $\mathcal{U}_r$, i.e., the vector of probabilities of selecting different regions for vehicle $\mathcal{U}_r$, and let $\vec{\boldsymbol{q}}_m = (\boldsymbol{q}^1, \ldots, \boldsymbol{q}^m) \in \Delta_{n-1}^m$. We note that for the randomized ensemble CUSUM algorithm with m vehicles the space of vehicle routing policies is $\Omega = \Delta_{n-1}^m$. We construct a lower bound on the processing times at differ-

ent regions for different vehicles in the following way. Let Ξ be the set of all the sets with cardinality m in which each entry is an arbitrarily chosen region; equivalently, $\Xi = \{\mathcal{R}_1, \ldots, \mathcal{R}_n\}^m$. Let a realization of the processing times at the regions in a set $\xi \in \Xi$ be $t_1^\xi, \ldots, t_m^\xi$. We now define a lower bound $\bar{T}_{\text{one}}$ to the expected value of the minimum of the processing times at m arbitrary regions as $\bar{T}_{\text{one}} = \min\{\mathbb{E}[\min\{t_1^\xi, \ldots, t_m^\xi\}] \mid \xi \in \Xi\}$.

Theorem 4.5 (***Multi-vehicle randomized ensemble CUSUM***)**.** For the randomized ensemble CUSUM algorithm with m vehicles and stationary region selection policies $\boldsymbol{q}^r$, $r \in \{1, \ldots, m\}$, the detection delay δ^k at region $\mathcal{R}_k$ satisfies:

$$\mathbb{E}_{f_k^1}[\delta_k(\vec{\boldsymbol{q}}_m)] \geq \frac{\bar{\eta}\, \bar{T}_{\text{one}}}{\sum_{r=1}^m q_k^r \mathcal{D}_k}.$$

Proof. We construct a modified stochastic process to determine a lower bound on the expected detection delay. For the randomized ensemble CUSUM algorithm with multiple vehicles, let t_r^b be the the processing time for the vehicle $\mathcal{U}_r$ during its b-th visit to any region. We assume that the sampling time for each vehicle at its b-th visit in the modified process is $\min\{t_1^b, \ldots, t_m^b\}$. Therefore, the sampling time for the modified process is the same at each region. Further, it is identically distributed for each visit and has expected value greater than or equal to $\bar{T}_{\text{one}}$. We further assume that the distances between the regions are zero. Such a process underestimates the processing and travel time required to collect each observation

in the randomized ensemble CUSUM algorithm. Hence, the expected detection delay for this process provides a lower bound to the expected detection delay for randomized ensemble CUSUM algorithm. Further, for this process the vehicles operate synchronously and the expected value of the likelihood ratio at region k at each iteration is $\sum_{r=1}^{m} q_k^r \mathcal{D}(f_k^1, f_k^0)$. The remainder of the proof follows similar to the proof for single vehicle case in Theorem 4.2. □

4.2.4 Design for multiple vehicles

We now design an efficient stationary policy for randomized ensemble CUSUM algorithm with multiple vehicles. We propose an algorithm that partitions the set of regions into m subsets, allocates one vehicle to each subset, and implements our single vehicle efficient stationary policy in each subset. This procedure is formally defined in Algorithm 4.2.

Let the subset of regions allocated to vehicle $\mathcal{U}_r$ be $\mathcal{S}^r, r \in \{1, \ldots, m\}$. We will denote the elements of subset $\mathcal{S}^r$ by $\mathcal{S}_i^r, i \in \{1, \ldots, n_r\}$. Let $\vec{\boldsymbol{q}}_{\text{part}}^{\dagger} \in \Delta_{n-1}^m$ be a stationary routing policy under the partitioning algorithm that implements single vehicle efficient stationary policy in each partition. We define the weights in equation (4.1) by $w_k = \pi_k^1 / \sum_{j=1}^{n} \pi_j^1$, where π_k^1 is the prior probability of anomaly at region $\mathcal{R}_k$. Let $w_{\min} = \min\{w_1, \ldots, w_n\}$ and $w_{\max} = \max\{w_1, \ldots, w_n\}$. We

Algorithm 4.2: *Partitioning Algorithm*

Input : vehicles $\{\mathcal{U}_1, \ldots, \mathcal{U}_m\}$, regions $\mathcal{R} = \{\mathcal{R}_1, \ldots, \mathcal{R}_n\}$, a single vehicle routing policy;

Require : $n > m$;

Output : a m-partition of the regions ;

1 partition $\mathcal{R}$ into m arbitrary subsets $\{\mathcal{S}^r\}_{r\in\{1,\ldots,m\}}$ with cardinalities $n_r \leq \lceil n/m \rceil, r \in \{1, \ldots, m\}$;

2 allocate vehicle $\mathcal{U}_r$ to subset $\mathcal{S}^r$, for each $r \in \{1, \ldots, m\}$;

3 implement the single vehicle efficient stationary policy in each subset.

now analyze the performance of the partitioning algorithm and show that it is within a factor of optimal.

Theorem 4.6 (***Performance of the partitioning policy***)**.** For the partitioning algorithm with m vehicles and n regions that implements the single vehicle efficient stationary policy in each partition, the following statements hold:

(i). the average detection delay under partitioning policy satisfies the following upper bound

$$\delta_{\text{avg}}(\vec{\boldsymbol{q}}^{\dagger}_{\text{part}}) \leq m \left\lceil \frac{n}{m} \right\rceil^2 \frac{w_{\max}\bar{\eta}(\bar{T}_{\max} + d_{\max})}{\mathcal{D}_{\min}};$$

(ii). the average detection delay satisfies the following lower bound

$$\delta_{\text{avg}}(\vec{\boldsymbol{q}}_m) \geq \Big(\sum_{k=1}^{n} \sqrt{\frac{w_k}{\mathcal{D}_k}} \Big)^2 \frac{\bar{\eta}\bar{T}_{\text{one}}}{m},$$

for any $\vec{\boldsymbol{q}}_m \in \Delta_{n-1}^m$;

(iii). the stationary policy $\vec{\boldsymbol{q}}_{\text{part}}^{\dagger}$ is within a factor of optimal, and

$$\frac{\delta_{\text{avg}}(\vec{\boldsymbol{q}}_{\text{part}}^{\dagger})}{\delta_{\text{avg}}(\vec{\boldsymbol{q}}_m^*)} \leq \frac{4w_{\max}}{w_{\min}} \frac{(\bar{T}_{\max} + d_{\max})}{\bar{T}_{\text{one}}} \frac{\mathcal{D}_{\max}}{\mathcal{D}_{\min}}, \text{ and}$$
$$\frac{\delta_{\text{avg}}(\vec{\boldsymbol{q}}_{\text{part}}^{\dagger})}{\delta_{\text{avg}}^{m\text{-min}}} \leq m^2 \left\lceil \frac{n}{m} \right\rceil \frac{(\bar{T}_{\max} + d_{\max})}{\bar{T}_{\min}^{m\text{-smlst}}} \frac{\mathcal{D}_{\max}}{\mathcal{D}_{\min}},$$

where $\vec{\boldsymbol{q}}_m^*$ is optimal stationary policy;

(iv). the expected detection delay at region $\mathcal{R}_k$ under the stationary policy $\vec{\boldsymbol{q}}_{\text{part}}^{\dagger}$ satisfies

$$\frac{\mathbb{E}[\delta_k(\vec{\boldsymbol{q}}_{\text{part}}^{\dagger})]}{\delta_k^{m\text{-min}}} \leq \frac{m(\bar{T}_{\max} + d_{\max})}{\bar{T}_k^{m\text{-smlst}}} \sqrt{\left\lceil \frac{n}{m} \right\rceil \frac{\mathcal{D}_k}{w_k \mathcal{D}_{\min}}}.$$

Proof. We start by establishing the first statement. We note that under the partitioning policy, the maximum number of regions a vehicle serves is $\lceil n/m \rceil$. It follows from Theorem 4.4 that for vehicle $\mathcal{U}_r$ and the associated partition $\mathcal{S}^r$, the average detection delay is upper bounded by

$$\begin{aligned}\delta_{\text{avg}}(\boldsymbol{q}_{\text{part}}^r) &\leq \Big(\sum_{i=1}^{n_r} \sqrt{\frac{w_i}{\mathcal{D}_i}} \Big)^2 \bar{\eta}(\bar{T}_{\max} + d_{\max}) \\ &\leq \left\lceil \frac{n}{m} \right\rceil^2 \frac{\bar{\eta} w_{\max}(\bar{T}_{\max} + d_{\max})}{\mathcal{D}_{\min}}.\end{aligned}$$

Therefore, the overall average detection delay satisfies $\delta_{\text{avg}}(\vec{\boldsymbol{q}}_{\text{part}}^{\dagger}) \leq m\delta_{\text{avg}}(\boldsymbol{q}_{\text{part}}^r)$. This establishes the first statement.

To prove the second statement, we utilize the lower bounds obtained in Theorem 4.5 and construct a lower bound to the average detection delay $\delta^m_{\text{lower}} : \Delta^m_{n-1} \to \mathbb{R}_{>0} \cup \{+\infty\}$ defined by $\delta^m_{\text{lower}}(\vec{\boldsymbol{q}}_m) = \sum_{k=1}^n (v_k \bar{T}_{\text{one}} / \sum_{r=1}^m q_k^r)$. It can be verified that

$$\min_{\vec{\boldsymbol{q}}_m \in \Delta^m_{n-1}} \delta^m_{\text{lower}}(\vec{\boldsymbol{q}}_m) = \Big(\sum_{k=1}^n \sqrt{\frac{w_k}{\mathcal{D}_k}} \Big)^2 \frac{\bar{\eta} \bar{T}_{\text{one}}}{m}.$$

We now establish the first part of the third statement. Note that

$$\begin{aligned} \frac{\delta_{\text{avg}}(\vec{\boldsymbol{q}}^{\dagger}_{\text{part}})}{\delta_{\text{avg}}(\vec{\boldsymbol{q}}^{*}_{m})} &\leq \frac{\lceil n/m \rceil^2}{(n/m)^2} \frac{w_{\max}}{w_{\min}} \frac{(\bar{T}_{\max} + d_{\max})}{\bar{T}_{\text{one}}} \frac{\mathcal{D}_{\max}}{\mathcal{D}_{\min}} \\ &\leq \frac{4 w_{\max}}{w_{\min}} \frac{(\bar{T}_{\max} + d_{\max})}{\bar{T}_{\text{one}}} \frac{\mathcal{D}_{\max}}{\mathcal{D}_{\min}}, \end{aligned}$$

where the last inequality follows from the fact that $(\lceil n/m \rceil)/(n/m) \leq 2$.

The remainder of the proof follows similar to the proof of Theorem 4.4. □

4.3 Adaptive ensemble CUSUM Algorithm

The stationary vehicle routing policy does not utilize the real-time information regarding the likelihood of anomalies at the regions. We now develop an adaptive policy that incorporates the anomaly likelihood information provided by the anomaly detection algorithm. We consider the CUSUM statistic at a region as a measure of the likelihood of an anomaly at that region, and utilize it at each iteration to design new prior probability of an anomaly for each region. At each iteration, we adapt the efficient stationary policy using this new prior probability.

This procedure results in higher probability of visiting an anomalous region and, consequently, it improves the performance of our efficient stationary policy. In Section 4.4 we provide numerical evidence showing that the adaptive ensemble CUSUM algorithm improves the performance of randomized ensemble CUSUM algorithm.

Our adaptive ensemble CUSUM algorithm is formally presented in Algorithm 4.3 for the single vehicle case. For the case of multiple vehicles we resort to the partitioning Algorithm 4.2 that implements the single vehicle adaptive ensemble CUSUM Algorithm 4.3 in each partition. Let us denote the adaptive routing policy for a single vehicle by $\boldsymbol{a}$ and the policy obtained from the partitioning algorithm that implements single vehicle adaptive routing policy in each partition by $\boldsymbol{a}_{\text{part}}$. We now analyze the performance of the adaptive ensemble CUSUM algorithm. Since, the probability to visit any region varies with time in the adaptive ensemble CUSUM algorithm, we need to determine the number of iterations between two consecutive visit to a region, i.e., the number of iterations for the recurrence of the region. We first derive a bound on the expected number of samples to be drawn from a time-varying probability vector for the recurrence of a particular state.

Lemma 4.7 (***Mean observations for region recurrence***)**.** Consider a sequence $\{x_\tau\}_{\tau\in\mathbb{N}}$, where x_τ is sampled from a probability vector $\boldsymbol{p}^\tau \in \Delta_{n-1}$. If the

k-th entry of $\boldsymbol{p}^\tau$ satisfy $p_k^\tau \in (\alpha_k, \beta_k)$, for each $\tau \in \mathbb{N}$ and some $\alpha_k, \beta_k \in (0,1)$, then the number of iterations I_k for the recurrence of state k satisfy $\mathbb{E}[I_k] \leq \beta_k/\alpha_k^2$.

Proof. The terms of the sequence $\{x_\tau\}_{\tau\in\mathbb{N}}$ are statistically independent. Further, the probability mass function p^τ is arbitrary. Therefore, the bound on the expected iterations for the first occurrence of state k is also a bound on the subsequent recurrence of state k. The expected number of iterations for first occurrence of region k are

$$\mathbb{E}[I_k] = \sum_{i\in\mathbb{N}} i p_k^i \prod_{j=1}^{i-1}(1-p_k^j) \leq \beta_k \sum_{i\in\mathbb{N}} i(1-\alpha_k)^{i-1} = \beta_k/\alpha_k^2.$$

This establishes the statement. □

We utilize this upper bound on the expected number of iterations for recurrence of a region to derive performance metrics for the adaptive ensemble CUSUM algorithm. We now derive an upper bound on the expected detection delay at each region for adaptive ensemble CUSUM algorithm. We derive these bounds for the expected evolution of the CUSUM statistic at each region.

Theorem 4.8 (***Adaptive ensemble CUSUM algorithm***)**.** Consider the expected evolution of the CUSUM statistic at each region. For the partitioning algorithm that implements single vehicle adaptive ensemble CUSUM algorithm

(Algorithm 4.3) in each subset of the partition, the following statement holds:

$$\mathbb{E}[\delta_k(\boldsymbol{a}_{\text{part}})] \leq \Big(\frac{\bar{\eta}}{\mathcal{D}_k} + \frac{2(\lceil n/m \rceil - 1)e^{\eta/2}\sqrt{\mathcal{D}_k}(1-e^{-\bar{\eta}/2})}{\sqrt{\mathcal{D}_{\min}}(1-e^{-\mathcal{D}_k/2})} + \frac{(\lceil n/m \rceil - 1)^2 e^{\eta}\mathcal{D}_k(1-e^{-\bar{\eta}})}{\mathcal{D}_{\min}(1-e^{-\mathcal{D}_k})}\Big)(\bar{T}_{\max} + d_{\max}).$$

Proof. We start by deriving expression for a single vehicle. Let the number of iterations between the $(j-1)$th and jth visit to region $\mathcal{R}_k$ be I_j^k.

Let the observation during the jth visit to region $\mathcal{R}_k$ be y_j and the CUSUM statistic at region $\mathcal{R}_k$ after the visit be C_j^k. It follows that the probability to visit region $\mathcal{R}_k$ between $(j-1)$th and jth visit is greater than

$$p_k^{j-1} = \frac{e^{C_{j-1}^k/2}/\sqrt{\mathcal{D}_k}}{e^{C_{j-1}^k/2}/\sqrt{\mathcal{D}_k} + (n-1)e^{\eta/2}/\sqrt{\mathcal{D}_{\min}}}.$$

Therefore, it follows from Lemma 4.7 that

$$\mathbb{E}[I_j^k] \leq (1 + (n-1)e^{(\eta - C_{j-1}^k)/2}\sqrt{\mathcal{D}_k/\mathcal{D}_{\min}})^2.$$

Note that $C_j^k = \max\{0, C_{j-1}^k + \log(f_k^1(y_j)/f_k^0(y_j))\}$. Since, maximum of two convex function is a convex function, it follows from Jensen inequality, see [76], that

$$\mathbb{E}[C_j^k] \geq \max\{0, \mathbb{E}[C_{j-1}^k] + \mathcal{D}_k\} \geq \mathbb{E}[C_{j-1}^k] + \mathcal{D}_k.$$

Therefore, $\mathbb{E}[C_j^k] \geq j\mathcal{D}_k$ and for expected evolution of the CUSUM statistics

$$\mathbb{E}[I_j^k] \leq (1 + (n-1)e^{(\eta - (j-1)\mathcal{D}_k)/2}\sqrt{\mathcal{D}_k/\mathcal{D}_{\min}})^2.$$

Therefore, the total number of iterations N_k required to collect N_k^{obs} observations at region $\mathcal{R}_k$ satisfy

$$\begin{aligned}\mathbb{E}[N_k(\boldsymbol{a})|N_k^{\text{obs}}] &= \sum_{j=1}^{N_k^{\text{obs}}}(1+(n-1)e^{(\eta-(j-1)\mathcal{D}_k)/2}\sqrt{\mathcal{D}_k/\mathcal{D}_{\min}})^2 \\ &= N_k^{\text{obs}} + \frac{2(n-1)e^{\eta/2}\sqrt{\mathcal{D}_k}(1-e^{-\mathcal{D}_k N_k^{\text{obs}}/2})}{\sqrt{\mathcal{D}_{\min}}(1-e^{-\mathcal{D}_k/2})} \\ &\quad + \frac{(n-1)^2 e^{\eta}\mathcal{D}_k(1-e^{-\mathcal{D}_k N_k^{\text{obs}}})}{\mathcal{D}_{\min}(1-e^{-\mathcal{D}_k})}.\end{aligned}$$

Note that the number of observations N_k^{obs} required at region $\mathcal{R}_k$ satisfy $\mathbb{E}[N_k^{\text{obs}}] = \bar{\eta}/\mathcal{D}_k$. It follows from Jensen's inequality that

$$\mathbb{E}[N_k(\boldsymbol{a})] \leq \frac{\bar{\eta}}{\mathcal{D}_k} + \frac{2(n-1)e^{\eta/2}\sqrt{\mathcal{D}_k}(1-e^{-\bar{\eta}/2})}{\sqrt{\mathcal{D}_{\min}}(1-e^{-\mathcal{D}_k/2})}\frac{(n-1)^2 e^{\eta}\mathcal{D}_k(1-e^{-\bar{\eta}})}{\mathcal{D}_{\min}(1-e^{-\mathcal{D}_k})}.$$

Since the expected time required to collect each evidence is smaller $\bar{T}_{\max} + d_{\max}$, it follows that

$$\mathbb{E}[\delta_k(\boldsymbol{a})] \leq (\bar{T}_{\max} + d_{\max})\mathbb{E}[N_k(\boldsymbol{a})].$$

The expression for the partitioning policy that implement single vehicle adaptive routing policy in each partition follow by substituting $\lceil n/m \rceil$ in the above expressions. This completes the proof of the theorem. □

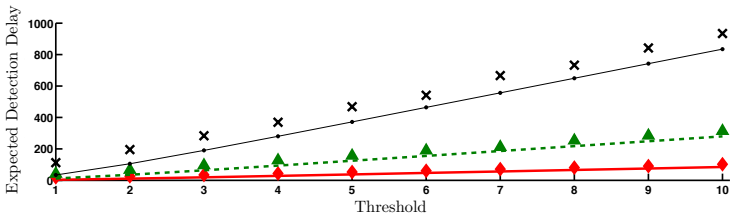
Remark 4.6 (***Performance bound***)**.** The bound derived in Theorem 4.8 is very conservative. Indeed, it assumes the CUSUM statistic at each region to be fixed at its maximum value η, except for the region in consideration. This is practically never the case. In fact, if at some iteration the CUSUM statistic is

close to η, then it is highly likely that the vehicle visits that region at the next iteration, so that the updated statistic crosses the threshold η and resets to zero. □

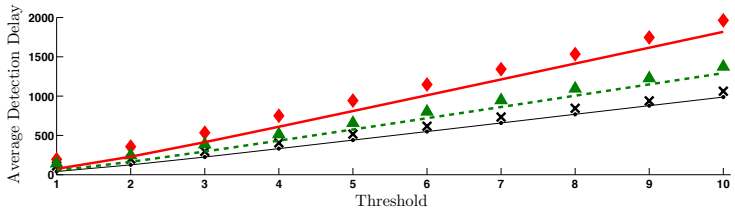
4.4 Numerical Results

We now elucidate on the concepts developed in this chapter through some numerical examples. We first validate the expressions for expected detection delay obtained in Section 4.2.

Example 4.1 (***Expected detection delay***)**.** Consider a set of 4 regions surveyed by a single vehicle. Let the location of the regions be $(10, 0), (5, 0), (0, 5)$, and $(0, 10)$, respectively. The vector of processing times at each region is $(1, 2, 3, 4)$. Under the nominal conditions, the observations at each region are sampled from normal distributions $\mathcal{N}(0, 1), \mathcal{N}(0, 1.33), \mathcal{N}(0, 1.67)$ and $\mathcal{N}(0, 2)$, respectively, while under anomalous conditions, the observations are sampled from normal distributions with unit mean and same variance as in nominal case. Let the prior probability of anomaly at each region be 0.5. An anomaly appears at each region at time $50, 200, 350$, and 500, respectively. Assuming that the vehicle is holonomic and moves at unit speed, the expected detection delay at region $\mathcal{R}_1$ and the average detection delay are shown in Fig. 4.3. It can be seen that the theoret-



(a) Expected detection delay at region $\mathcal{R}_1$



(b) Average detection delay

Figure 4.3: Expected and average detection delay. Solid black line with dots and black $\times$, respectively, represent the theoretical expression and the value obtained by Monte-Carlo simulations under stationary policy $\boldsymbol{q} = [\ 0.2\ 0.25\ 0.25\ 0.3\ ]$. Dashed green line and green triangles, respectively, represent the theoretical expression and the value obtained by Monte-Carlo simulations under stationary policy $\boldsymbol{q} = [\ 0.5\ 0.2\ 0.2\ 0.1\ ]$. Solid red line and red diamonds, respectively, represent the theoretical expression and the value obtained by Monte-Carlo simulations under stationary policy $\boldsymbol{q} = [\ 0.85\ 0.05\ 0.05\ 0.05\ ]$.

ical expressions provide a lower bound to the expected detection delay obtained through Monte-Carlo simulations. This phenomenon is attributed to the Wald's approximation. □

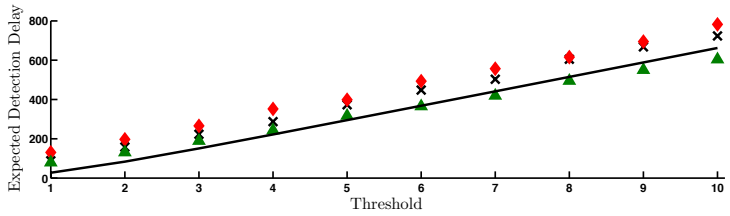
We remarked earlier that if each region cannot be reached from another region in a single hop, then a fastest mixing Markov chain (FMMC) with the desired stationary distribution can be constructed. Consider a set of regions modeled by the graph $\mathcal{G} = (V, \mathcal{E})$, where V is the set of nodes (each node corresponds to a region) and $\mathcal{E}$ is the set of edges representing the connectivity of the regions. The

transition matrix of the FMMC $P \in \mathbb{R}^{n\times n}$ with a desired stationary distribution $\boldsymbol{q} \in \Delta_{n-1}$ can be determined by solving the following convex minimization problem proposed in [14]:

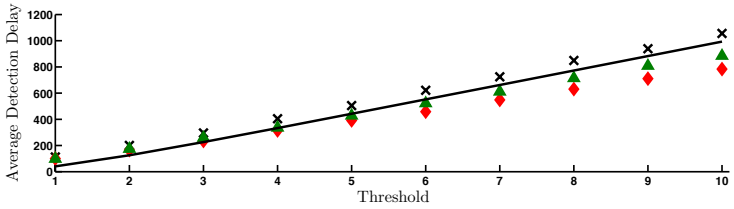
$$\begin{aligned} \text{minimize} \quad & \|Q^{1/2}PQ^{1/2} - \boldsymbol{q}_{\text{root}}\boldsymbol{q}_{\text{root}}^T\|_2 \\ \text{subject to} \quad & P\mathbf{1} = \mathbf{1} \\ & QP = P^TQ \\ & P_{ij} \geq 0, \text{ for each } (i,j) \in \mathcal{E} \\ & P_{ij} = 0, \text{ for each } (i,j) \notin \mathcal{E}, \end{aligned}$$

where Q is a diagonal matrix with diagonal $\boldsymbol{q}$, $\boldsymbol{q}_{\text{root}} = (\sqrt{q_1}, \ldots, \sqrt{q_n})$, and $\mathbf{1}$ is the vector of all ones. We now demonstrate the effectiveness of FMMC in our setup.

Example 4.2 (***Effectiveness of FMMC***)**.** Consider the same set of data as in Example 4.1. We study the expected and average detection delay for randomized ensemble CUSUM algorithm when the regions to visit are sampled from the FMMC. The expected and average detection delay for all-to-all connection topology, line connection topology and ring connection topology are shown in Fig. 4.4. It can be seen that the performance under all three topologies is remarkably close to each other. □



(a) Expected detection delay at region $\mathcal{R}_1$



(b) Average detection delay

Figure 4.4: Expected and average detection delay for uniform stationary policy. The solid black line represents the theoretical expression. The black $\times$, red diamonds, and green triangles, respectively, represent the values obtained by Monte-Carlo simulations for all-to-all, line, and ring connection topology. For the line and ring topologies, the region to visit at each iteration is sampled from the fastest mixing Markov chain with the desired stationary distribution.

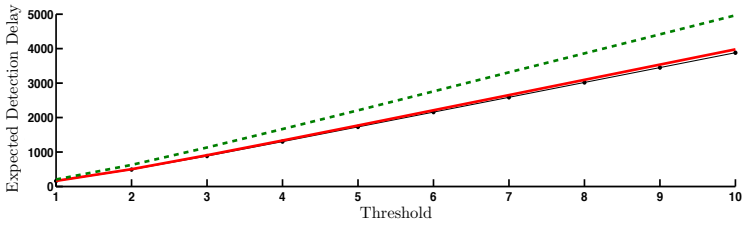


Figure 4.5: Average detection delay for a single vehicle. The solid red line, the dashed green line, and the solid black line with dots represent efficient, uniform, and optimal stationary policies, respectively.

We now study the performance of the (numerically computed) optimal and our efficient stationary policies for the single vehicle randomized ensemble CUSUM algorithm.

Example 4.3 (***Single vehicle optimal stationary policy***)**.** For the same set of data as in Example 4.1, we now study the performance of the uniform, the (numerically computed) optimal and our efficient stationary routing policies. A comparison is shown in Fig. 4.5. Notice that the performance of the optimal and efficient stationary policy is extremely close to each other. □

We now study the performance of the optimal, partitioning and uniform stationary policies for randomized ensemble CUSUM algorithm with multiple vehicles.

Example 4.4 (***Multiple-vehicle optimal stationary policy***)**.** Consider a set of 6 regions surveyed by 3 vehicles. Let the regions be located at $(10, 0), (5, 0), (0, 5), (0, 10), (0, 0)$ and $(5, 5)$. Let the processing time at each region be unitary. Under

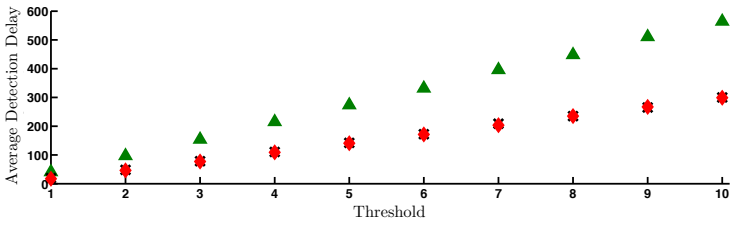


Figure 4.6: Average detection delay for 3 vehicles surveying 6 regions. The green triangles represent the policy in which each vehicle surveys each region uniformly. The red diamonds and black $\times$ represent the partitioning policy in which each vehicle implements the single vehicle efficient stationary policy and the single vehicle optimal stationary policy, respectively.

nominal conditions, the observations at each region are sampled from normal distributions $\mathcal{N}(0,1)$, $\mathcal{N}(0,1.4)$, $\mathcal{N}(0,1.8)$, $\mathcal{N}(0,2.2)$, $\mathcal{N}(0,2.6)$ and $\mathcal{N}(0,3)$, respectively. Under anomalous conditions, the observations are sampled from normal distributions with unit mean and same variance as in the nominal case. Let the prior probability of anomaly at each region be 0.5. An anomaly appears at each region at time $25, 35, 45, 55, 65$ and 75, respectively. Assuming that the vehicles are holonomic and moves at unitary speed, the average detection delay for the uniform stationary policy for each vehicle, the partitioning policy in which each vehicle implements single vehicle efficient stationary policy in each subset of the partition, and the partitioning policy in which each vehicle implements single vehicle optimal stationary policy in each subset of the partition is shown in Fig. 4.6.

□

We now study the performance of the adaptive ensemble CUSUM algorithm, and we numerically show that it improves the performance of our stationary policy.

Example 4.5 (***Adaptive ensemble CUSUM algorithm***)**.** Consider the same set of regions as in Example 4.1. Let the processing time at each region be unitary. The observations at each region are sampled from normal distributions $\mathcal{N}(0, \sigma^2)$ and $\mathcal{N}(1, \sigma^2)$, in nominal and anomalous conditions, respectively. Under the nominal conditions at each region and $\sigma^2 = 1$, a sample evolution of the adaptive ensemble CUSUM algorithm is shown in Fig. 4.7(a). The anomaly appears at regions $\mathcal{R}_2$, $\mathcal{R}_3$, and $\mathcal{R}_4$ at time 100, 300, and 500, respectively. Under these anomalous conditions and $\sigma^2 = 1$, a sample evolution of the adaptive ensemble CUSUM algorithm is shown in Fig. 4.7(b). It can be seen that the adaptive ensemble algorithm samples a region with high likelihood of anomaly with high probability, and, hence, it improves upon the performance of the stationary policy.

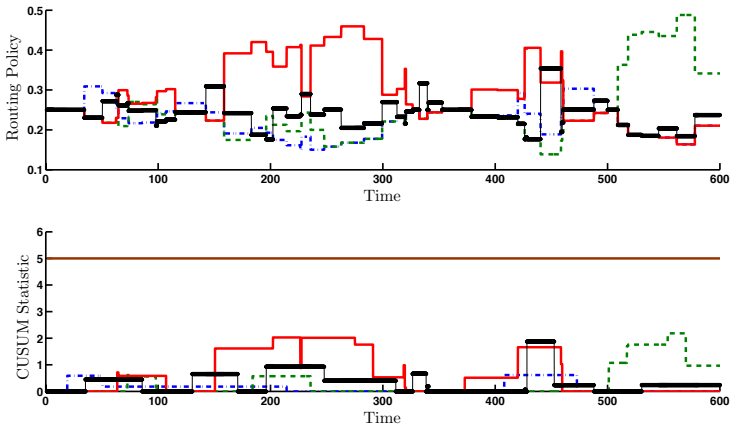
We now study the expected detection delay under adaptive ensemble CUSUM algorithm and compare it with the efficient stationary policy. The anomaly at each region appears at time $50, 200, 350$ and 500, respectively. The expected detection delay obtained by Monte-Carlo simulations for $\sigma^2 = 1$ and different thresholds is shown in Fig. 4.8(a). It can be seen that the adaptive policy improves the detection delay significantly over the efficient stationary policy for large thresholds. It should be noted that the detection delay minimization is most needed at large thresholds because the detection delay is already low at small thresholds. Furthermore, frequent false alarms are encountered at low thresholds and hence, low thresholds

are not typically chosen. The expected detection delay obtained by Monte-Carlo simulations for different value of σ^2 and threshold $\eta = 5$ is shown in Fig. 4.8(b). Note that for a given value of σ^2, the Kullback-Leibler divergence between $\mathcal{N}(1, \sigma^2)$ and $\mathcal{N}(0, \sigma^2)$ is $1/2\sigma^2$. It can be seen that the adaptive policy improves the performance of the stationary policy for each value of noise. □

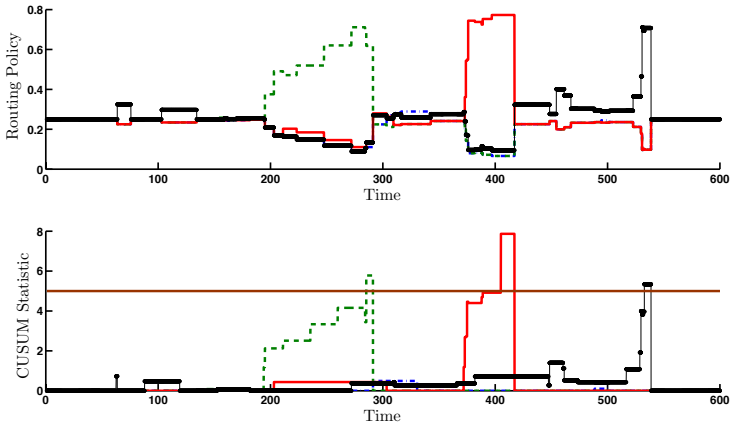
We now apply the adaptive ensemble CUSUM algorithm to a more general scenario where the anomalous distribution is not completely known. As remarked earlier, in this case, the CUSUM algorithm should be replaced with the GLR algorithm. Given the nominal probability density function f_k^0 and the anomalous probability density function $f_k^1(\cdot|\theta)$ parameterized by $\theta \in \Theta \subseteq \mathbb{R}^\ell$, for some $\ell \in \mathbb{N}$, the GLR algorithm, see [6], works identically to the CUSUM algorithm, except that the CUSUM statistic is replaced by the statistic

$$\Lambda_\tau^k = \max_{t\in\{1,\dots,\tau\}} \sup_{\theta\in\Theta} \sum_{i=t}^{\tau} \log \frac{f_k^1(y_i|\theta)}{f_k^0(y_i)}.$$

Example 4.6 (***Generalized Likelihood Ratio***)**.** For the same set of data as in Example 4.5, assume that there are three types of potential anomalies at each region. Since any combination of these anomalies can occur simultaneously, there are 7 potential distributions under anomalous conditions. We characterize these distributions as different hypothesis and assume that the observations under each hypothesis $h \in \{1, \dots, 8\}$ are sampled from a normal distribution with mean μ_h

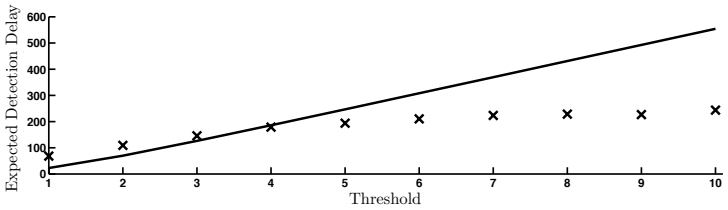


(a) CUSUM statistic and vehicle routing probabilities under nominal conditions

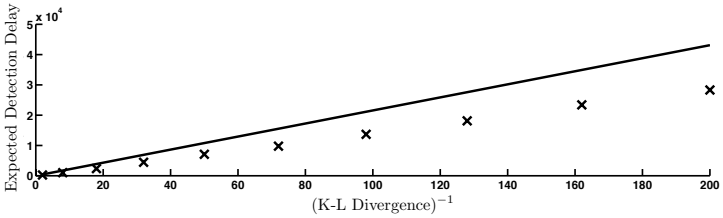


(b) CUSUM statistic and vehicle routing probabilities under anomalous conditions

Figure 4.7: Sample evolution of the adaptive ensemble CUSUM algorithm. The dashed-dotted blue line, dashed green line, solid red line and solid black line with dots represent data from regions $\mathcal{R}_1, \mathcal{R}_2, \mathcal{R}_3$ and $\mathcal{R}_4$, respectively. The solid brown horizontal line represents the threshold. The vehicle routing probability is a function of the likelihood of anomaly at each region. As the likelihood of an anomaly being present at a region increases, also the probability to survey that region increases. Anomalies appear at region $\mathcal{R}_2$, $\mathcal{R}_3$ and $\mathcal{R}_4$ at times 100, 300 and 500, respectively. Once an anomaly is detected, it is removed and the statistic is reset to zero.



(a) Expected detection delay as a function of threshold



(b) Expected detection delay as a function of KL divergence

Figure 4.8: Performance of the adaptive ensemble CUSUM algorithm. The solid black line represents the theoretical expected detection delay for the efficient stationary policy and the black $\times$ represent the expected detection delay for the adaptive ensemble CUSUM algorithm.

and covariances Σ_h. Let

$$\mu_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \mu_2 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \mu_3 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \mu_4 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix},$$
$$\mu_5 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, \mu_6 = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}, \mu_7 = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}, \mu_8 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \text{ and}$$

$$\Sigma_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \Sigma_2 = \begin{bmatrix} 2 & 1 & 0 \\ 1 & \frac{3}{2} & 0 \\ 0 & 0 & 1 \end{bmatrix}, \Sigma_3 = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & \frac{3}{2} \end{bmatrix}, \Sigma_4 = \begin{bmatrix} \frac{3}{2} & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 2 \end{bmatrix},$$
$$\Sigma_5 = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 1 \end{bmatrix}, \Sigma_6 = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 2 \end{bmatrix}, \Sigma_7 = \begin{bmatrix} 2 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 2 \end{bmatrix}, \Sigma_8 = \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix}.$$

We picked region $\mathcal{R}_1$ as non-anomalous, while hypothesis 4, 6, and 8 were true at regions $\mathcal{R}_2, \mathcal{R}_3$, and $\mathcal{R}_4$, respectively. The Kullback-Leibler divergence at a region was chosen as the minimum of all possible Kullback-Leibler divergences at that region. A sample evolution of the adaptive ensemble CUSUM algorithm with

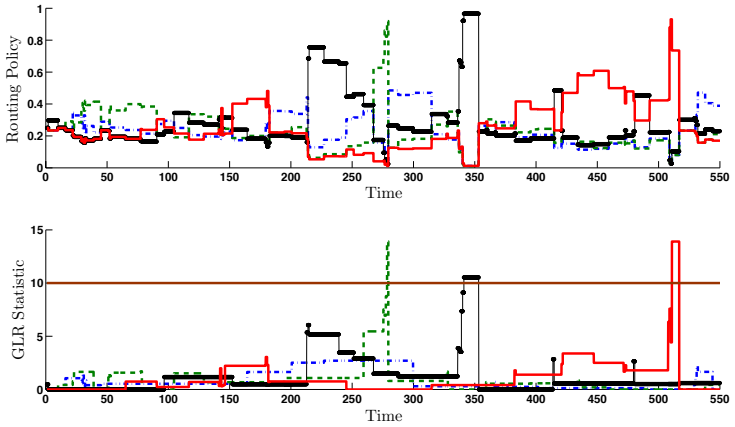
GLR statistic replacing the CUSUM statistic is shown in Fig 4.9(a). It can be seen the performance is similar to the performance in Example 4.5. As an additional ramification of this algorithm, we also get the likelihood of each hypothesis at each region. It can be seen in Fig 4.9(b) that the true hypothesis at each region corresponds to the hypothesis with maximum likelihood. □

4.5 Experimental Results

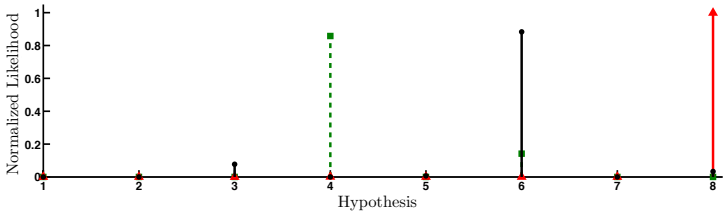
We first detail our implementation of the algorithms using the Player/Stage robot control software package and the specifics of our robot hardware. We then present the results of the experiment.

Robot hardware

We use Erratic mobile robots from Videre Design shown in Fig. 4.10. The robot platform has a roughly square footprint (40cm × 37cm), with two differential drive wheels and a single rear caster. Each robot carries an on-board computer with a 1.8Ghz Core 2 Duo processor, 1 GB of memory, and 802.11g wireless communication. For navigation and localization, each robot is equipped with a Hokuyo URG- 04LX laser rangefinder. The rangefinder scans 683 points over 240at 10Hz with a range of 5.6 meters.



(a) GLR statistic under anomalous conditions



(b) Normalized likelihood of each hypothesis

Figure 4.9: Sample evolution of the adaptive ensemble CUSUM algorithm with GLR statistic. The dashed-dotted blue line, dashed green line, solid red line and solid black line with dots represent data from regions $\mathcal{R}_1, \mathcal{R}_2, \mathcal{R}_3$ and $\mathcal{R}_4$, respectively. The solid brown horizontal line represents the threshold. The vehicle routing probability is a function of the likelihood of anomaly at each region. As the likelihood of an anomaly being present at a region increases, also the probability to survey that region increases. Anomalies appear at region $\mathcal{R}_2$, $\mathcal{R}_3$ and $\mathcal{R}_4$ at times 100, 300 and 500, respectively. Once an anomaly is detected, it is removed and the statistic is reset to zero. The true hypothesis at each region corresponds to the hypothesis with maximum likelihood

Figure 4.10: Erratic mobile robot with URG-04LX laser rangefinder.

Localization

We use the *amcl* driver in Player which implements Adaptive Monte-Carlo Localization, see [100]. The physical robots are provided with a map of our lab with a 15cm resolution and told their starting pose within the map (Fig. 4.11). We set an initial pose standard deviation of 0.9m in position and 12 in orientation, and request updated localization based on 50 of the sensors range measurements for each change of 2cm in robot position or 2 in orientation. We use the most likely pose estimate by amcl as the location of the robot.

Navigation

Each robot uses the *snd* driver in Player for the Smooth Nearness Diagram navigation, see [32]. For the hardware, we set the robot radius parameter to 22cm, obstacle avoidance distance to 0.5m, and maximum speed to 0.2m/s. We let a robot achieve its target when it is within 10cm of the target.

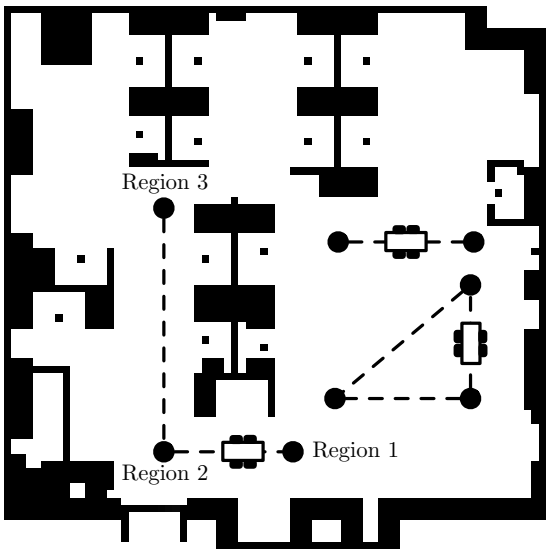


Figure 4.11: This figure shows a map of our lab together with our surveillance configuration. Three erratic robots survey the selected 8 regions (black dots), which have been partitioned among the robots. Regions 1, 2, and 3 are also considered in Fig. 4.12, where we report the statistics of our detection algorithm.

Experiment setup

For our experiment we employed our team of 3 Erratic robots to survey our laboratory. As in Fig. 4.11, a set of 8 important regions have been chosen and partitioned among the robots. Each robot surveys its assigned regions. In particular, each robot implements the single robot adaptive ensemble CUSUM algorithm in its regions. Notice that Robot 1 cannot travel from region 1 to region 3 in a single hop. Therefore, Robot 1 selects the regions according to a Markov chain with desired stationary distribution. This Markov chain was constructed using the Metropolis-Hastings algorithm. In particular, for a set of regions modeled as a graph $\mathcal{G} = (V, \mathcal{E})$, to achieve a desired stationary routing policy $\boldsymbol{q}$, the Metropolis-

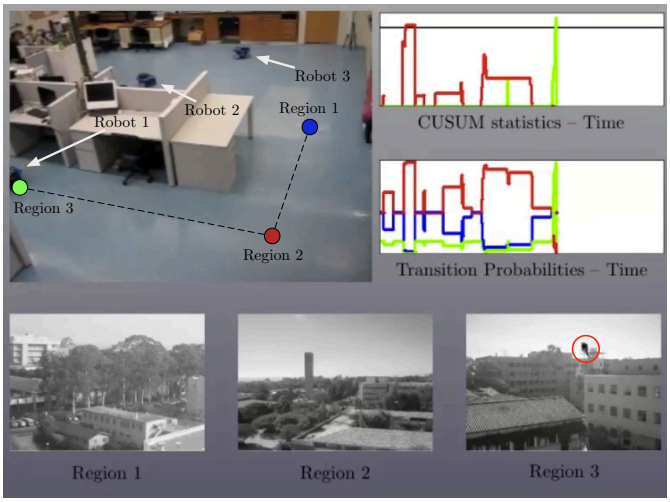


Figure 4.12: A snapshot of our surveillance experiment, where three robots survey six locations in our lab (Fig. 4.11). In this figure we show the three regions assigned to the first robot. Each region correspond to a part of our campus, and observations are taken accordingly. Notice that Region 3 contains an anomaly (black smoke), and that the CUSUM statistics, which are updated upon collection of observations, reveal the anomaly (green peak). The transition probabilities are updated according to our adaptive ensemble CUSUM algorithm.

Hastings algorithm, see [105], picks the transition matrix P with entries:

$$P_{ij} = \begin{cases} 0, & \text{if } (i,j) \notin \mathcal{E}, \\ \min\left\{\frac{1}{d_i}, \frac{q_j}{q_i d_j}\right\} & \text{if } (i,j) \in \mathcal{E} \text{ and } i \neq j, \\ 1 - \sum_{k=1, k\neq i}^{n} P_{ik} & \text{if } (i,j) \in \mathcal{E} \text{ and } i = j, \end{cases}$$

where d_i is the number of regions that can be visited from region $\mathcal{R}_i$.

Observations (in the form of pictures) are collected by a robot each time a region is visited. In order to have a more realistic experiment, we map each location in our lab to a region in our campus. Then, each time a robot visit

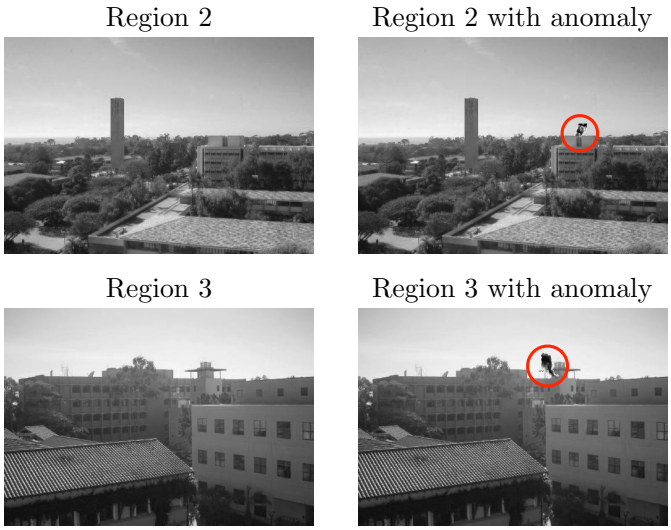


Figure 4.13: This figure shows sample pictures from Region 2 and Region 3, both with and without the anomaly to be detected.

a region in our lab, a picture of a certain region in our campus is selected as observation (see Fig. 4.12). Pictures have been collected prior to the experiment.

Finally, in order to demonstrate the effectiveness of our anomaly detection algorithm, some pictures from regions 2 and 3 have been manually modified to contain an anomalous pattern; see Fig. 4.13. Anomalous pictures are collected by Robot 1 at some pre-specified time instants (the detection algorithm, however, does not make use of this information).

Probability density function estimation

In order to implement our adaptive ensemble CUSUM algorithm, the probability density functions of the observations at the regions in presence and absence

of an anomaly need to be estimated. For this task, we first collect sample images, and we register them in order to align their coordinates, see[74]. We then select a reference image, and compute the difference between the sample pictures and the reference image. Then, we obtain a coarse representation of each difference image by dividing the image into blocks. For each difference image, we create a vector containing the mean value of each block, and we compute the mean and standard deviation of these vectors. Finally, we fit a normal distribution to represent the collected nominal data. In order to obtain a probability density distribution of the images with anomalies, we manually modify the nominal images, and we repeat the same procedure as in the nominal case.

Experiment results

The results of our experiment are illustrated in Fig. 4.12, Fig. 4.13, and in the multimedia extension available at `http://www.ijrr.org`. From the CUSUM statistics we note that the anomalies in Region 2 and Region 3 are both detected: indeed both the red curve and the green curve pass the decision threshold. We also note that few observations are necessary to detect the anomaly. Since the robots successfully survey the given environment despite sensor and modeling uncertainties due to real hardware, we conclude that our modeling assumptions are not restrictive.

4.6 Conclusions and Future Directions

In this chapter we studied a spatial quickest detection problem in which multiple vehicles surveil a set of regions to detect anomalies in minimum time. We developed a novel ensemble CUSUM algorithm to detect an anomaly in any of the regions. A stochastic vehicle routing policy was adopted in which the vehicle samples the next region to visit from a probability vector. In particular, we studied (i) stationary policy: the probability vector is a constant function of time; and (ii) adaptive policy: the probability vector is adapted with time based on the collected observations. We designed an efficient stationary policy that depends on the travel time of the vehicles, the processing time required to collect information at each region, and the anomaly detection difficulty at each region. In adaptive policy, we modified the efficient stationary policy at each iteration to ensure that the regions with high likelihood of anomaly are visited with high probability, and thus, improved upon the performance of the stationary policy. We also mentioned the methods that extend the ideas in this chapter immediately to the scenario in which the distributions of the observations in presence and absence of anomaly are not completely known, but belong to some parametrized family, or to the scenario in which the observations collected from each region are not independent (e.g., in the case of dynamic anomalies).

There are several possible extensions of the ideas considered here. First, in the case of dependent observations at each region, the current method assumes known distributions in the presence and absence of anomalies. An interesting direction is to design quickest detection strategies that are robust to the uncertainties in these distributions. Second, the anomalies considered in this chapter are always contained in the same region. It would be of interest to consider anomalies that can move from one region to another. Third, the policy presented in this chapter considers an arbitrary partition that satisfy some cardinality constraints. It is of interest to come up with *smarter* partitioning policies that take into consideration the travel times, and the difficulty of detection at each region. Last, to construct the fastest mixing Markov chain with desired stationary distribution, we relied on time-homogeneous Markov chains. A time varying Markov chain may achieve a faster convergence to the desired stationary distribution, see [37]. This is also an interesting direction to be pursued.

4.7 Appendix: Probabilistic guarantee to the uniqueness of critical point

We now provide probabilistic guarantee for Conjecture 4.3. The average detection delay for a single vehicle under a stationary policy $\boldsymbol{q}$ is

$$\delta_{\text{avg}}(\boldsymbol{q}) = \Big(\sum_{i=1}^{n} \frac{v_i}{q_i}\Big)\Big(\sum_{i=1}^{n} q_i \bar{T}_i + \sum_{i=1}^{n}\sum_{j=1}^{n} q_i q_j d_{ij}\Big),$$

where $v_i = w_i\bar{\eta}/\mathcal{D}_i$ for each $i \in \{1,\dots,n\}$. A local minimum of δ_{avg} can be can be found by substituting $q_n = 1 - \sum_{j=1}^{n-1} q_j$, and then running the gradient descent algorithm from some initial point $\boldsymbol{q}_0 \in \Delta_{n-1}$ on the resulting objective function.

Let $\boldsymbol{v} = (v_1,\dots,v_n)$ and $\boldsymbol{T} = (\bar{T}_1,\dots,\bar{T}_n)$. We assume that the parameters $\{\boldsymbol{v}, \boldsymbol{T}, D, n\}$ in a given instance of optimization problem (4.3) and the chosen initial point $\boldsymbol{q}_0$ are realizations of random variables sampled from some space $\mathcal{K}$. For a given realization $\kappa \in \mathcal{K}$, let the realized value of the parameters be $\{\boldsymbol{v}(\kappa), \boldsymbol{T}(\kappa), D(\kappa), n(\kappa)\}$, and the chosen initial point be $\boldsymbol{q}_0(\kappa)$. The associated optimization problem is:

$$\underset{\boldsymbol{q}\in\Delta_{n(\kappa)-1}}{\text{minimize}} \quad \delta_{\text{avg}}(\boldsymbol{q} \,|\, \kappa), \tag{4.4}$$

where, for a given realization $\kappa \in \mathcal{K}$, $\delta_{\text{avg}}(\cdot \,|\, \kappa) : \Delta_{n(\kappa)-1} \to \mathbb{R}_{>0} \cup \{+\infty\}$ is defined by

$$\delta_{\text{avg}}(\boldsymbol{q} \,|\, \kappa) = \Big(\sum_{i=1}^{n(\kappa)} \frac{v_i(\kappa)}{q_i}\Big)\Big(\sum_{i=1}^{n(\kappa)} q_i \bar{T}_i(\kappa) + \sum_{i=1}^{n(\kappa)}\sum_{j=1}^{n(\kappa)} q_i q_j d_{ij}(\kappa)\Big).$$

For a given realization κ, let $gd(\cdot \,|\, \kappa) : \Delta_{n(\kappa)-1} \to \Delta_{n(\kappa)-1}$ be the function that determines the outcome of the gradient descent algorithm applied to the function obtained by substituting $q_{n(\kappa)} = 1 - \sum_{j=1}^{n(\kappa)-1} q_j$ in $\delta_{\text{avg}}(\boldsymbol{q} \,|\, \kappa)$. In other words, the gradient descent algorithm starting from point $q_0(\kappa)$ converges to the point $gd(q_0(\kappa) \,|\, \kappa)$. Consider N_1 realizations $\{\kappa_1, \ldots, \kappa_{N_1}\} \in \mathcal{K}^{N_1}$. Let $\boldsymbol{q}_{\text{optimal}}(\kappa) = gd(\frac{1}{n(\kappa)}\mathbf{1}_{n(\kappa)} \,|\, \kappa)$, and define

$$\hat{\gamma} = \max\{\|gd(\boldsymbol{q}_0(\kappa_s) \,|\, \kappa_s) - \boldsymbol{q}_{\text{optimal}}(\kappa_s)\| \mid s \in \{1, \ldots, N_1\}\}.$$

It is known from [19] that if $N_1 \geq -(\log \nu_1)/\mu_1$, for some $\mu_1, \nu_1 \in]0, 1[$, then, with at least confidence $1 - \nu_1$, it holds

$$\mathbb{P}(\{\boldsymbol{q}_0(\kappa) \in \Delta_{n(\kappa)-1} \mid \|gd(\boldsymbol{q}_0(\kappa) \,|\, \kappa) - \boldsymbol{q}_{\text{optimal}}(\kappa)\| \leq \hat{\gamma}\}) \geq 1 - \mu_1,$$

for any realization $\kappa \in \mathcal{K}$.

We sample the following quantities: the value n as uniformly distributed in $\{3, \ldots, 12\}$; each coordinate of the n regions in two dimensional space from the normal distribution with mean 0 and variance 100; the value T_i, for each $i \in \{1, \ldots, n\}$, from the half normal distribution with mean 0 and variance 100; and the value v_i, for each $i \in \{1, \ldots, n\}$, uniformly from $]0, 1[$. For a realized value of n, we chose $\boldsymbol{q}_0$ uniformly in Δ_{n-1}. Let the matrix D be the Euclidean distance matrix between the n sampled regions.

We considered $N_1 = 1000$ realizations of the parameters $\{\boldsymbol{v}, \boldsymbol{T}, D, n\}$ and initial value $\boldsymbol{q}_0$. The sample sizes were determined for $\mu_1 = 0.01$ and $\nu_1 = 10^{-4}$. The value of $\hat{\gamma}$ obtained was 10^{-4}. Consequently, the gradient descent algorithm for the optimization problem (4.3) starting from any feasible point yields the same solution with high probability. In other words, with at least confidence level 99.99% and probability at least 99%, the optimization problem (4.3) has a unique critical point at which the minimum is achieved.

Algorithm 4.3: *Single Vehicle Adaptive Ensemble CUSUM*

Input : parameters η, $\mathcal{D}_k$, pdfs f_k^0, f_k^1, for each $k \in \{1, \ldots, n\}$;

Output : decision on anomaly at each region ;

1 set $\Lambda_0^j = 0$, for all $j \in \{1, \ldots, n\}$, and $\tau = 1$;

while true **do**

2 set new prior $\pi_k^1 = e^{\Lambda_\tau^k}/(1 + e^{\Lambda_\tau^k})$, for each $k \in \{1, \ldots, n\}$

3 set $q_k = \frac{\sqrt{\pi_k^1/\mathcal{D}_k}}{\sum_{j=1}^n \sqrt{\pi_j^1/\mathcal{D}_j}}$, for each $k \in \{1, \ldots, n\}$;

4 sample a region from probability distribution $(q_1, \ldots, q_n)$;

5 collect sample y_τ from region k;

6 update the CUSUM statistic at each region

$$\Lambda_\tau^j = \begin{cases} \left(\Lambda_{\tau-1}^k + \log \frac{f_k^1(y_\tau)}{f_k^0(y_\tau)}\right)^+, & \text{if } j = k; \\ \Lambda_{\tau-1}^j, & \text{if } j \in \{1, \ldots, n\} \setminus \{k\}; \end{cases}$$

if $\Lambda_\tau^k > \eta$ **then**

7 anomaly detected at region $\mathcal{R}_k$;

8 set $\Lambda_\tau^k = 0$;

9 set $\tau = \tau + 1$;

Chapter 5

Operator Attention Allocation via Knapsack Problems

In this chapter we study knapsack problems with sigmoid utility. The time evolution of the performance of a human operator is well modeled by a sigmoid function (see Section 2.3) and consequently, the knapsack problems with sigmoid utility well model the resource (attention) allocation problem of a time-constrained human operator. We study three particular knapsack problems with sigmoid utility, namely, the knapsack problem with sigmoid utility, the generalized assignment problem with sigmoid utility, and the bin-packing problem with sigmoid utility. These problems model situations where human operators are looking at the feeds from a camera network and deciding on the presence of some malicious activity. The first problem determines the optimal fraction of work-hours an operator should allocate to each feed such that their overall performance is optimal. The second problem determines the allocations of the tasks to identical and inde-

pendently working operators as well as the optimal fraction of work-hours each operator should allocate to each feed such that the overall performance of the team is optimal. Assuming that the operators work in an optimal fashion, the third problem determines the minimum number of operators and an allocation of each feed to some operator such that each operator allocates non-zero fraction of work-hours to each feed assigned to them. We study these knapsack problems under following assumptions on the sigmoid functions:

(A1). *Smooth sigmoid functions:* For the ease of the presentation, we focus on smooth sigmoid functions. The analysis presented here extends immediately to non-smooth functions by using the sub-derivative instead of the derivative in the following analysis.

(A2). *Monotonic sigmoid functions:* We assume that the sigmoid functions associated with the problem are non-decreasing. In several interesting budget allocation problems, e.g. [75], the sigmoid utility is not a non-decreasing function. The approach in this chapter extends to the case of a general sigmoid utility. The algorithms proposed in this chapter also involve certain heuristics that improve the constant-factor solution. These heuristics exploit the monotonicity of the sigmoid function and will not hold for a general sig-

moid function. We note that even without the performance improvement heuristics the solution is within a constant factor of optimal.

Note that both these assumptions hold true for sigmoid functions associated with human performance.

5.1 Sigmoid Function and Linear Penalty

In order to gain insight into the behavior of sigmoid functions, we start with a simple problem with a very interesting result. We study the maximization of a sigmoid function subject to a linear penalty. Consider a sigmoid function f and a penalty rate $c \in \mathbb{R}_{>0}$, and the following optimization problem:

$$\underset{t \geq 0}{\text{maximize}} \quad f(t) - ct. \tag{5.1}$$

The derivative of a sigmoid function is not a one to one mapping and hence, not invertible. We define the pseudo-inverse of the derivative of a sigmoid function f with inflection point t^{inf}, $f^{\dagger} : \mathbb{R}_{>0} \to \mathbb{R}_{\geq 0}$ by

$$f^{\dagger}(y) = \begin{cases} \max\{t \in \mathbb{R}_{\geq 0} \mid f'(t) = y\}, & \text{if } y \in (0, f'(t^{\text{inf}})], \\ 0, & \text{otherwise.} \end{cases} \tag{5.2}$$

Notice that the definition of the pseudo-inverse is consistent with Figure 2.1. We now present the solution to the problem (5.1).

Lemma 5.1 (***Sigmoid function with linear penalty***)**.** For the optimization problem (5.1), the optimal allocation t^* is

$$t^* := \text{argmax}\{f(\beta) - c\beta \mid \beta \in \{0, f^\dagger(c)\}\}.$$

Proof. The global maximum lies at the point where first derivative is zero or at the boundary. The first derivative of the objective function is $f'(t) - c$. If $f'(t^{\text{inf}}) < c$, then the objective function is a decreasing function of time and the maximum is achieved at $t^* = 0$. Otherwise, a critical point is obtained by setting first derivative zero. We note that $f'(t) = c$ has at most two roots. If there exist two roots, then only the larger root lies in the region where the objective is concave and hence corresponds to a maximum. Otherwise, the only root lies in the region where the objective is concave and hence corresponds to a local maximum. The global maximum is determined by comparing the local maximum with the value of the objective function at the boundary $t = 0$. This completes the proof. □

The optimal solution to problem (5.1) for different values of penalty rate c is shown in Figure 5.1. One may notice the optimal allocation jumps down to zero at a critical penalty rate. This jump in the optimal allocation gives rise to combinatorial effects in the problems involving multiple sigmoid functions.

Definition 5.1 (***Critical penalty rate***)**.** Given a sigmoid function f and a linear penalty, we refer to the maximum penalty rate at which problem (5.1) has

a non-zero solution by critical penalty rate. Formally, for a given sigmoid function f and penalty rate $c \in \mathbb{R}_{>0}$, let the solution of the problem (5.1) be $t^*_{f,c}$, the critical penalty rate ψ_f is defined by

$$\psi_f = \max\{c \in \mathbb{R}_{>0} \mid t^*_{f,c} \in \mathbb{R}_{>0}\}.$$

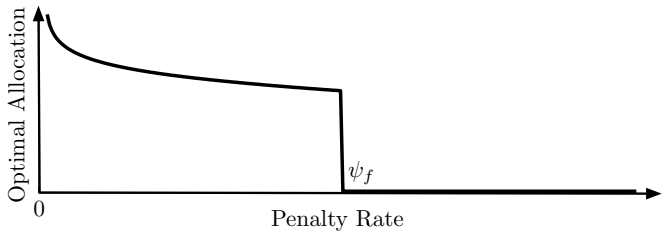


Figure 5.1: Optimal allocation to a sigmoid function as a function of linear penalty

5.2 Knapsack Problem with Sigmoid Utility

In this section, we consider the knapsack problem (KP) with sigmoid utility. We first define the problem and then develop an approximation algorithm for it.

5.2.1 KP with Sigmoid Utility: Problem Description

Consider a single knapsack and N items. Let the utility of item $\ell \in \{1, \ldots, N\}$ be a sigmoid function $f_\ell : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$. Given total available resource $T \in \mathbb{R}_{>0}$, the objective of the *KP with sigmoid utility* is to determine the resource allocation to each item such that the total utility of the knapsack is maximized. Formally,

the KP with sigmoid utility is posed as:

$$\begin{aligned} \underset{\boldsymbol{t}\succeq 0}{\text{maximize}} \quad & \sum_{\ell=1}^{N} f_\ell(t_\ell) \\ \text{subject to} \quad & \sum_{\ell=1}^{N} t_\ell \leq T. \end{aligned} \tag{5.3}$$

The KP with sigmoid utility models the situation where a human operator has to perform N decision making tasks within time T. If the performance of the human operator on task ℓ is given by sigmoid function f_ℓ, then the optimal duration allocation to each task is determined by the solution of problem (5.3). We remark that the objective function in problem (5.3) can be a weighted sum of performance functions as well. Such weights can be absorbed into the performance functions yielding a new sigmoid performance function. We now state the following proposition from [1]:

Proposition 5.2 (***Hardness of KP with sigmoid utility***)**.** The KP with sigmoid utility is NP-hard, unless P = NP.

5.2.2 KP with Sigmoid Utility: Approximation Algorithm

We now develop an approximation algorithm for the KP with sigmoid utility. We define the Lagrangian $L : \mathbb{R}^N_{>0} \times \mathbb{R}_{\geq 0} \times \mathbb{R}^N_{\geq 0} \to \mathbb{R}$ for the knapsack problem with sigmoid utility (5.3) by

$$L(\boldsymbol{t}, \alpha, \boldsymbol{\mu}) = \sum_{\ell=1}^{N} f_\ell(t_\ell) + \alpha(T - \sum_{\ell=1}^{N} t_\ell) + \boldsymbol{\mu}^{\mathrm{T}} \boldsymbol{t}.$$

Let t_ℓ^{inf} be the inflection point of sigmoid function f_ℓ and $f_\ell^\dagger$ be the pseudo-inverse of its derivative as defined in equation (5.2). We define the maximum value of the derivative of sigmoid function f_ℓ by $\alpha_\ell = f_\ell'(t_\ell^{\text{inf}})$. We also define $\alpha_{\max} = \max\{\alpha_\ell \mid \ell \in \{1, \ldots, N\}\}$. We will later show that $\alpha_{\max}$ is the maximum possible value of the Lagrange multiplier α.

We define the set of *inconsistent* sigmoid functions by $\mathcal{I} = \{\ell \in \{1, \ldots, N\} \mid t_\ell^{\text{inf}} > T\}$. Similarly and accordingly, we define the set of *consistent* sigmoid functions as $\{1, \ldots, N\} \setminus \mathcal{I}$. We denote the j-th element of the standard basis of $\mathbb{R}^N$ by $\boldsymbol{e}_j$.

Define $F : (0, \alpha_{\max}] \to \mathbb{R}_{\geq 0}$ as the optimal value of the objective function in the following α-parametrized KP:

$$\begin{aligned} \text{maximize} \quad & \sum_{\ell=1}^{N} x_\ell f_\ell(f_\ell^\dagger(\alpha)) \\ \text{subject to} \quad & \sum_{\ell=1}^{N} x_\ell f_\ell^\dagger(\alpha) \leq T \\ & x_\ell \in \{0, 1\}, \quad \forall \ell \in \{1, \ldots, N\}. \end{aligned} \tag{5.4}$$

Define $F_{\text{LP}} : (0, \alpha_{\max}] \to \mathbb{R}_{\geq 0}$ as the optimal value of the objective function in the following α-parametrized fractional KP:

$$\begin{aligned} \text{maximize} \quad & \sum_{\ell=1}^{N} x_\ell f_\ell(f_\ell^\dagger(\alpha)) \\ \text{subject to} \quad & \sum_{\ell=1}^{N} x_\ell f_\ell^\dagger(\alpha) \leq T \\ & x_\ell \in [0, 1], \quad \forall \ell \in \{1, \ldots, N\}. \end{aligned} \tag{5.5}$$

For a given α, the solution to problem (5.5) is obtained in the following way:

(i). sort tasks such that

$$\frac{f_1(f_1^\dagger(\alpha))}{f_1^\dagger(\alpha)} \geq \frac{f_2(f_2^\dagger(\alpha))}{f_2^\dagger(\alpha)} \geq \ldots \geq \frac{f_N(f_N^\dagger(\alpha))}{f_N^\dagger(\alpha)};$$

(ii). find $k := \min\{j \in \{1, \ldots, N\} \mid \sum_{i=1}^{j} f_i^\dagger(\alpha) \geq T\}$;

(iii). the solution is $x_1^{\text{LP}} = x_2^{\text{LP}} = \ldots = x_{k-1}^{\text{LP}} = 1$, $x_k^{\text{LP}} = (T - \sum_{i=1}^{k-1} f_i^\dagger(\alpha))/f_k^\dagger(\alpha)$, and $x_{k+1}^{\text{LP}} = x_{k+2}^{\text{LP}} = \ldots = x_N^{\text{LP}} = 0$.

A 2-factor solution to the binary KP (5.4) is obtained by picking the better of the sets of tasks $\{1, \ldots, k-1\}$ and $\{k\}$ (see [54, 57] for details). Let $F_{\text{approx}} : (0, \alpha_{\max}] \to \mathbb{R}_{\geq 0}$ be the value of the objective function in the α-parametrized knapsack problem under such 2-factor solution.

We now state the following important property of the function F_{LP}:

Lemma 5.3 (***Discontinuity of*** F_{LP})**.** The maximal set of points of discontinuity of the function F_{LP} is $\{\alpha_1, \ldots, \alpha_N\}$.

Proof. For each $\alpha \in [0, \alpha_{\max}]$, the α-parametrized fractional KP is a linear program, and the solution lies at one of the vertex of the feasible simplex. Note that if $f_\ell^\dagger(\alpha)$ is a continuous function for each $\ell \in \{1, \ldots, N\}$, then the vertices of the feasible simplex are continuous functions of α. Further, the objective function is also continuous if $f_\ell^\dagger(\alpha)$ is a continuous function for each $\ell \in \{1, \ldots, N\}$.

Therefore, the function F_{LP} may be discontinuous only if $f_\ell^\dagger(\alpha)$ is discontinuous for some ℓ, i.e., $\alpha \in \{\alpha_1, \ldots, \alpha_N\}$. □

We will show that if each sigmoid function is consistent, then the allocation to each sigmoid function can be written in terms of the Lagrange multiplier α, and the KP with sigmoid utility (5.3) reduces to α-parametrized KP (5.4). Further, the optimal Lagrange multiplier α^* can be searched in the interval $]0, \alpha_{\max}]$, and the α^*-parametrized KP can be solved using standard approximation algorithms to determine a solution within a constant factor of optimality. The search of the optimal Lagrange multiplier is a univariate continuous optimization problem and a typical optimization algorithm will determine the exact optimal Lagrange multiplier only asymptotically, but it will converge to an arbitrarily small neighborhood of the optimal Lagrange multiplier in finite number of iterations. Thus, a factor of optimality within an ϵ neighborhood of the desired factor of optimality, for any $\epsilon > 0$, can be achieved in a finite number of iterations. We utilize these ideas to develop a $(2 + \epsilon)$-factor approximation algorithm for KP with sigmoid utility in Algorithm 5.1. This algorithm searches for the Lagrange multiplier that maximizes the optimal value function F_{LP} of the α-parametrized fractional KP and truncates the associated solution to determine a constant-factor solution to the knapsack problem with sigmoid utility. This algorithm also involves a performance improvement heuristic in which the unemployed resource is allocated to

the most beneficial task amongst those with zero resource. We note that if the sigmoid utilities are non-smooth, then the standard KKT conditions in the following analysis are replaced with the KKT conditions for non-smooth optimization problems [44].

We define an ϵ-approximate maximizer of a function as a point in the domain of the function at which the function attains a value within ϵ of its maximum value. We now analyze Algorithm 5.1.

Theorem 5.4 (***KP with sigmoid utility***)**.** The following statements hold for the KP with sigmoid utility (5.3) and the solution obtained via Algorithm 5.1:

(i). the solution is within factor of optimality $(2+\epsilon)$, for any $\epsilon > 0$;

(ii). if ϵ-approximate maximizer over each continuous piece of F_{LP} can be searched using a constant number of function evaluations, then the Algorithm 5.1 runs in $O(N^2)$ time.

Proof. We apply the Karush-Kuhn-Tucker necessary conditions [60] for an optimal solution:

Linear dependence of gradients

$$\frac{\partial L}{\partial t_\ell^*}(\boldsymbol{t}^*, \alpha^*, \boldsymbol{\mu}^*) = f'_\ell(t_\ell^*) - \alpha^* + \mu_\ell^* = 0, \text{ for each } \ell \in \{1, \ldots, N\}. \tag{5.6}$$

Algorithm 5.1: *KP with Sigmoid Utility: Approximation Algorithm*

Input : $f_\ell, \ell \in \{1, \ldots, N\}, T \in \mathbb{R}_{>0}$;

Output : optimal allocations $\boldsymbol{t}^* \in \mathbb{R}^N_{\geq 0}$;

% *search for optimal Lagrange multiplier*

1 $\alpha^*_{\mathrm{LP}} \leftarrow \mathrm{argmax}\{F_{\mathrm{LP}}(\alpha) \mid \alpha \in [0, \alpha_{\max}]\}$;

2 determine the 2-factor solution $\boldsymbol{x}^*$ of α^*_{LP}-parametrized knapsack problem ;

% *determine best inconsistent sigmoid function*

3 find $\ell^* \leftarrow \mathrm{argmax}\{f_\ell(T) \mid \ell \in \mathcal{I}\}$;

% *pick the best among consistent and inconsistent tasks*

4 **if** $f_{\ell^*}(T) > F_{\mathrm{approx}}(\alpha^*_{\mathrm{LP}})$ **then**

  $\boldsymbol{t}^* = T\boldsymbol{e}_{\ell^*}$;

5 **else**

  $t^\dagger_\ell \leftarrow x^*_\ell f^\dagger_\ell(\alpha^*_{\mathrm{LP}}), \forall \ell \in \{1, \ldots, N\}$;

  % *heuristic to improve performance*

  % *pick the best sigmoid function with zero resource*

6 $t^*_\ell \leftarrow \begin{cases} t^\dagger_\ell, & \text{if } \ell \in \{1, \ldots, N\} \setminus \bar{\ell} \\ T - \sum_{\ell=1}^N t^\dagger_\ell, & \text{if } \ell = \bar{\ell}; \end{cases}$

Feasibility of the solution

$$T - \mathbf{1}_N^{\mathrm{T}} \boldsymbol{t}^* \geq 0 \quad \text{and } \boldsymbol{t}^* \succeq 0. \tag{5.7}$$

Complementarity conditions

$$\alpha^*(T - \mathbf{1}_N^{\mathrm{T}}\boldsymbol{t}^*) = 0. \tag{5.8}$$

$$\mu_\ell^* t_\ell^* = 0, \quad \text{for each } \ell \in \{1, \ldots, N\}. \tag{5.9}$$

Non-negativity of the multipliers

$$\alpha^* \geq 0, \quad \boldsymbol{\mu}^* \succeq 0. \tag{5.10}$$

Since f_ℓ is a non-decreasing function, for each $\ell \in \{1, \ldots, N\}$, the constraint (5.7) should be active, and thus, from complementarity condition (5.8) $\alpha^* > 0$. Further, from equation (5.9), if $t_\ell^* \neq 0$, then $\mu_\ell^* = 0$. Therefore, if a non-zero resource is allocated to sigmoid function $f_\eta, \eta \in \{1, \ldots, N\}$, then it follows from equation (5.6)

$$f_\eta'(t_\eta^*) = \alpha^*. \tag{5.11}$$

Assuming each f_ℓ is consistent, i.e., $t_\ell^{\inf} \leq T$, for each $\ell \in \{1, \ldots, N\}$, the second order condition [60] yields that a local maxima exists at $\boldsymbol{t}^*$ only if

$$f_\eta''(t_\eta^*) \leq 0 \iff t_\eta^* \geq t_\eta^{\inf}. \tag{5.12}$$

The equations (5.11) and (5.12) yield that optimal non-zero allocation to sigmoid function f_η is

$$t_\eta^* = f_\eta^\dagger(\alpha^*). \tag{5.13}$$

Given the optimal Lagrange multiplier α^*, the optimal non-zero allocation to the sigmoid function f_η is given by equation (5.13). Further, the optimal set of

sigmoid functions with non-zero allocations is the solution to the α^*-parametrized knapsack problem (5.4). We now show that α^* is maximizer of F. Since, at least one task is processed, $w_\ell f'_\ell(t^*_\ell) = \alpha$, for some $\ell \in \{1, \ldots, N\}$. Thus, $\alpha \in [0, \alpha_{\max}]$. By contradiction assume that $\bar{\alpha}$ is the maximizer of F, and $F(\bar{\alpha}) > F(\alpha^*)$. This means that the allocation corresponding to $\bar{\alpha}$ yields higher reward than the allocation corresponding to α^*. This contradictions equation (5.13).

If $t^{\inf}_\ell > T$, for some $\ell \in \{1, \ldots, N\}$, then equation (5.12) does not hold for any $t_\ell \in [0, T]$. Since, f_ℓ is convex in the interval $[0, T]$, the optimal allocation for maximum is at the boundary, i.e., $t_\ell \in \{0, T\}$. Therefore, as exemplified in Figure 5.2, the optimal allocation is either $T\boldsymbol{e}_\ell$ or lies at the projection of the simplex on the hyperplane $t_\ell = 0$. The projection of the simplex on the hyperplane $t_\ell = 0$ is again a simplex and the argument holds recursively.

To establish the first statement we note that α^*_{LP} is maximizer of F_{LP}, and the α-parametrized fractional knapsack problem is relaxation of α-parametrized knapsack problem, hence

$$F_{\text{LP}}(\alpha^*_{\text{LP}}) \geq F_{\text{LP}}(\alpha^*) \geq F(\alpha^*). \tag{5.14}$$

We further note that α^* is maximizer of F and F_{approx} is sub-optimal value of the objective function, hence

$$F(\alpha^*) \geq F(\alpha^*_{\text{LP}}) \geq F_{\text{approx}}(\alpha^*_{\text{LP}}) \geq \frac{1}{2} F_{\text{LP}}(\alpha^*_{\text{LP}}), \tag{5.15}$$

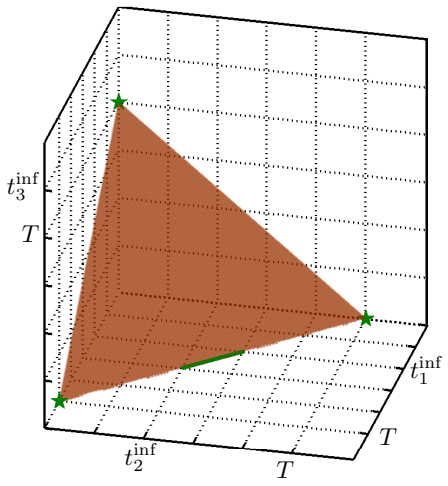


Figure 5.2: Possible locations of the maximum are shown in green stars and solid green line. The maximum possible allocation T is smaller than the inflection point of the third sigmoid function. For any allocation to the third sigmoid function, the corresponding entry in the Hessian matrix is positive, and the optimal allocation to third sigmoid function is 0 or T. Optimal allocation to the first and second sigmoid function may lie at the vertex of simplex, or at a location where Jacobian is zero and Hessian matrix is negative definite.

where the last inequality follows from the construction of F_{approx} (see 2-factor policy for binary knapsack problem [57]). The value of objective function at allocation $\boldsymbol{t}^{\dagger}$ in Algorithm 5.1 is equal to $F_{\text{approx}}(\alpha^*_{\text{LP}})$. The allocation $\boldsymbol{t}^{\dagger}$ may not saturate the total available resource T. Since, the sigmoid functions are non-decreasing function of the allocated resource, the total resource must be utilized, and it is heuristically done in step 6 of the Algorithm 5.1. This improves the value of the objective function and the factor of optimality remains at most 2. Finally, since a numerical method will only compute ϵ-approximate maximizer of F_{LP} in finite time, the factor of optimality increases to $(2 + \epsilon)$.

To establish the last statement, we note that each evaluation of F_{LP} requires the solution of fractional KP and has $O(N)$ computational complexity. According to Lemma 5.3 that the maximum number of points of discontinuity of F_{LP} is $N+1$. Therefore, if ϵ-approximate maximizer over each continuous piece of F_{LP} can be searched using a constant number of function evaluations, then $O(N)$ computations are needed over each continuous piece of F_{LP}. Consequently, the Algorithm 5.1 runs in $O(N^2)$ time. □

Corollary 5.5 (***Identical sigmoid functions***)**.** If the sigmoid functions in the KP with sigmoid utility (5.3) are identical and equal to f, then the optimal solution $\boldsymbol{t}^*$ is an N-tuple with m^* entries equal to T/m^* and all other entries zero, where

$$m^* = \underset{m\in\{1,\ldots,N\}}{\operatorname{argmax}} \quad m\, f(T/m). \tag{5.16}$$

Proof. It follows from Algorithm 5.1 that for identical sigmoid functions the optimal non-zero resource allocated is the same for each sigmoid function. The number of sigmoid functions with optimal non-zero resource is determined by equation (5.16), and the statement follows. □

Discussion 5.1 (***Search of the optimal Lagrange multiplier***)**.** The approximate solution to the KP with sigmoid utility in Algorithm 5.1 involves the search for α^*_{LP}, the maximizer of function F_{LP}. It follows from Lemma 5.3 that

this search corresponds to the global maximization of N univariate continuous functions. The global maximum over each continuous piece can be determined using the P-algorithm [58, 20]. If for a given instance of the KP with sigmoid utility, stronger properties of F_{LP} can be established, then better algorithms can be utilized, e.g., (i) if each continuous piece of F_{LP} is differentiable, then the modified P-algorithm [21] can be used for global optimization; (ii) if each continuous piece of F_{LP} is Lipschitz, then one of the algorithms in [41] can be used for the global optimization. □

Example 5.1. Given sigmoid functions $f_\ell(t) = w_\ell/(1 + \exp(-a_\ell t + b_\ell)), \ell \in \{1, \ldots, 10\}$ with parameters and associated weights

$$\boldsymbol{a} = (a_1, \ldots, a_{10}) = (1, 2, 1, 3, 2, 4, 1, 5, 3, 6),$$

$$\boldsymbol{b} = (b_1, \ldots, b_{10}) = (5, 10, 3, 9, 8, 16, 6, 30, 6, 12),$$

$$\text{and } \boldsymbol{w} = (w_1, \ldots, w_{10}) = (2, 5, 7, 4, 9, 3, 5, 10, 13, 6),$$

and total resource $T = 15$ units. The optimal solution and the approximate solution without the heuristic in step 6 of the Algorithm 5.1 are shown in Figure 5.3. The approximate solution with the performance improvement heuristic in step 6 of the Algorithm 5.1 gives the same solution as the optimal solution. The functions F, F_{approx}, and F_{LP} are shown in Figure 5.4. □

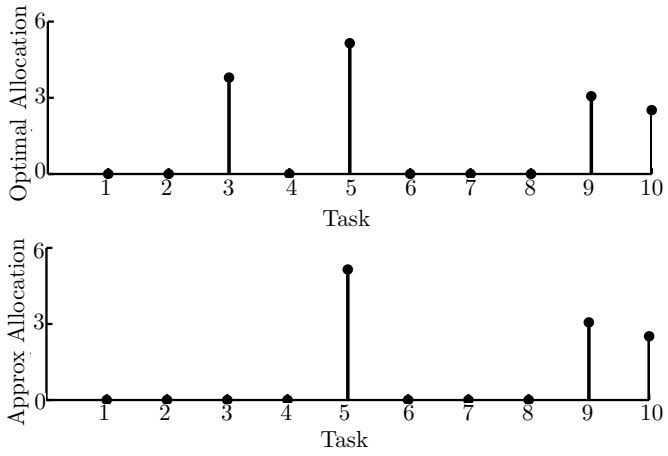


Figure 5.3: Optimal allocations and the approximate optimal allocations without performance improvement heuristic.

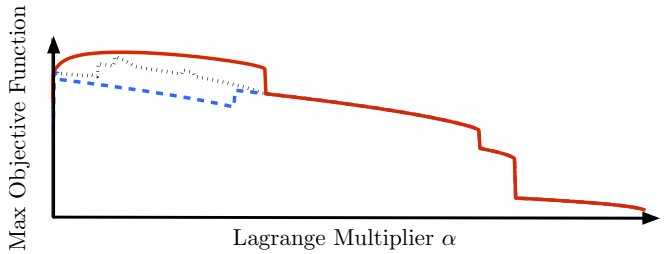


Figure 5.4: Exact and approximate maximum value of the objective function. The functions $F_{\text{LP}}, F, F_{\text{approx}}$ are shown by solid brown line, black dotted line, and blue dashed line, respectively. The points of discontinuity of function F_{LP} are at points where the Lagrange multiplier has value in the set $\{\alpha_1, \ldots, \alpha_N\}$.

Remark 5.1 (***Multiple-Choice KP with Sigmoid Utility***)**.** Consider m disjoint classes $\{N_1, \ldots, N_m\}$ of tasks and a single knapsack. The multiple-choice KP is to pick one task each from every class such that the total utility of the tasks is maximized for given total available resource. Let the total available resource be $T \in \mathbb{R}_{>0}$, and let the utility of allocating resource $t \in \mathbb{R}_{\geq 0}$ to task i in class N_j be a sigmoid function $f_{ij} : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$. The multiple-choice KP with sigmoid utility

is posed as:

$$\begin{aligned} \text{maximize} \quad & \sum_{i=1}^{m} \sum_{j \in N_i} f_{ij}(t_{ij}) x_{ij} \\ \text{subject to} \quad & \sum_{i=1}^{m} \sum_{j \in N_i} t_{ij} x_{ij} \leq T \\ & \sum_{j \in N_i} x_{ij} = 1, \\ & x_{ij} \in \{0, 1\}, \end{aligned} \tag{5.17}$$

for each $i \in \{1, \ldots, m\}$ and $j \in N_i$. The multiple-choice KP with sigmoid utility models a situation where a human operator has to process one task each from a given classes of tasks within time T. The performance of the operator on task i from class N_j is given by sigmoid function f_{ij}. The different tasks in a given class may be, e.g., observations collected from different sensors in a given region. The methodology developed in this section extends to a multiple-choice KP with sigmoid utility (5.17). In particular, the problem (5.17) can be reduced to an α-parameterized multiple-choice knapsack problem, and the LP relaxation based 2-factor approximation algorithm for the binary multiple choice knapsack problem [54] can be utilized to determine a 2-factor algorithm for the problem (5.17).

□

5.3 Generalized Assignment Problem with Sigmoid Utility

In this section, we consider the generalized assignment problem (GAP) with sigmoid utility. We first define the problem and then develop an approximation algorithm for it.

5.3.1 GAP with Sigmoid Utility: Problem Description

Consider M bins (knapsacks) and N items. Let T_j be the capacity of bin $j \in \{1, \ldots, M\}$. Let the utility of item $i \in \{1, \ldots, N\}$ when allocated to bin i be a sigmoid function $f_{ij} : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ of the allocated recourse t_{ij}. The GAP with sigmoid utility determines the optimal assignment of the items to bins such that the total utility of the bins is maximized. Formally, the GAP with sigmoid utility is posed as:

$$\begin{aligned} \text{maximize} \quad & \sum_{j=1}^{M} \sum_{i=1}^{N} f_{ij}(t_{ij}) x_{ij} \\ \text{subject to} \quad & \sum_{i=1}^{N} t_{ij} x_{ij} \leq T_j, \\ & \sum_{j=1}^{M} x_{ij} \leq 1, \\ & x_{ij} \in \{0, 1\}, \end{aligned} \tag{5.18}$$

for each $i \in \{1, \ldots, N\}$ and $j \in \{1, \ldots, M\}$. The GAP with sigmoid utility models a situation where M human operators have to independently serve N tasks. The operator j works for a duration T_j. The performance of human operator j on task i is given by sigmoid function f_{ij}. The solution to the GAP determines optimal assignments of the tasks to the operators and the optimal time-duration to be allocated to each processed task. We now state the following result about the hardness of the GAP with sigmoid utility:

Proposition 5.6 (***Hardness of GAP with sigmoid utility***)**.** The generalized assignment problem with sigmoid utility is NP-hard, unless P = NP.

Proof. The statement follows from the fact that knapsack problem with sigmoid utility is a special case of the generalized assignment problem with sigmoid utility, and is NP-hard according to Proposition 5.2. □

5.3.2 GAP with Sigmoid Utility: Approximation Algorithm

We now propose an approximation algorithm for the GAP with sigmoid utility. This algorithm is an adaptation of the 3-factor algorithm for the binary GAP proposed in [27] and is presented in Algorithm 5.2. In Algorithm 5.2, F is the matrix of sigmoid functions f_{ij}, $F^{(j)}_{*1}$ represents first column of matrix $F^{(j)}$, and

$E^2_{*2:M-j+1}$ represents matrix E^2 with first column removed. Similarly, $F_{I_{\text{unproc}}j}$ represents the vector with entries $F_{ij}, i \in I_{\text{unproc}}$, and $t_{\bar{A}j}$ represents the vector with entries $t_{ij}, i \in \bar{A}$. The algorithm calls a recursive function, namely, $\texttt{GAP}(\cdot,\cdot)$ (defined in Algorithm 5.3) to compute the solution of GAP with sigmoid utility. The solution comprises of a set A describing assignment of tasks to bins and $\boldsymbol{t}$ describes associated duration allocations. The function `GAP` utilizes the solution to KP with sigmoid utility at each recursion to compute the solution to the GAP with sigmoid utility. In particular, the function $\texttt{KP}(\cdot,\cdot)$ determines the set of tasks to be processed $\bar{A}$, and allocations $\bar{\boldsymbol{t}}$ according to Algorithm 5.1. The key element of the algorithm is the decomposition of the performance matrix at each recursion. In particular, the performance matrix F is decomposed into E^1 and E^2 at each recursion, where E^1 corresponds to the current bin and allocations to it according to Algorithm 5.1, and E^2 corresponds to a GAP with smaller number of bins. The performance is efficient with respect to the component E^1 due to effectiveness of Algorithm 5.1, while component E^2 recursively reduces to the single bin case, i.e., to the KP with sigmoid utility. The algorithm also involves a performance improving heuristic. According to this heuristic, if the resource of a bin is not completely utilized and there are tasks that are not assigned to any bin, then a KP with sigmoid utility is solved using remaining resource and unassigned tasks. Likewise, if the resource of a bin is not completely utilized and each task has

been assigned to some bin, then the remaining resource is allocated to the most beneficial task in that bin. We now state some properties of this algorithm:

Theorem 5.7 (***GAP with sigmoid utility***)**.** The following statements hold for the GAP with sigmoid utility (5.18) and the solution obtained via Algorithm 5.2:

(i). the solution is within a factor $(3+\epsilon)$ of optimal, for any $\epsilon > 0$; and

(ii). the Algorithm 5.2 runs in $O(N^2M)$ time, provided the solution to KP with sigmoid utility can be computed in $O(N^2)$ time.

Proof. The proof is an adaptation of the inductive argument for the binary GAP in [27]. We note that for a single bin, the GAP reduces to the knapsack problem and Algorithm 5.1 provides a solution within $(2+\epsilon)$-factor of optimal. Consequently, Algorithm 5.2 provides a solution within $(2+\epsilon)$-factor of optimal, and hence, within $(3+\epsilon)$-factor of optimal. Assume by induction hypothesis that Algorithm 5.2 provides a solution with $(3+\epsilon)$-factor of optimal for L bins. We now consider the case with $(L+1)$ bins. The performance matrix F has two components, namely, E^1 and E^2. We note that first column of E^2 has each entry equal to zero, and thus, E^2 corresponds to a GAP with L bins. By the induction hypothesis, Algorithm 5.2 provides a solution within $(3+\epsilon)$-factor of optimal with respect to performance matrix E^2. We further note that the first column of E^1 is identical to the first column of F and Algorithm 5.1 provides a solution

Algorithm 5.2: *GAP with Sigmoid Utility: 3-factor Approximation*

Input : f_{ij}, T_j, $i \in \{1, \ldots, N\}, j \in \{1, \ldots, M\}$;

Output : assignment set $A = \{A_1, \ldots, A_M\}$ and allocations $\boldsymbol{t} \in \mathbb{R}_{\geq 0}^{N \times M}$;

% *Initialize*

1 $F^{(1)} \leftarrow F$;

% *Call function* `GAP`(Algorithm 5.3)

2 allocations $[A, \boldsymbol{t}] \leftarrow \texttt{GAP}(1, F^{(1)})$;

% *heuristic to improve performance*

% *assign unassigned tasks to unsaturated bins*

3 $I_{\text{unproc}} \leftarrow \{1, \ldots, N\} \setminus \cup_{i=1}^{m} A_i$;

4 **foreach** $j \in \{1, \ldots, M\}$ **do**

5 **if** $\sum_{i \in A_j} t_{ij} < T_j$ *and* $|I_{\text{unproc}}| > 0$ **then**

% *solve KP with unprocessed tasks*

6 $[\bar{A}, \bar{\boldsymbol{t}}] \leftarrow \texttt{KP}(F_{I_{\text{unproc}} j}, T_j - \sum_{i \in A_j} t_{ij})$;

7 $A_j \leftarrow A_j \cup \bar{A}; \quad t_{\bar{A} j} \leftarrow \bar{t}$;

8 **else if** $\sum_{i \in A_j} t_{ij} < T_j$ *and* $|I_{\text{unproc}}| = 0$ **then**

% *allocate remaining resource to the most rewarding task*

9 $\ell \leftarrow \text{argmax}\{f_{ij}(t_{ij} + T_j - \sum_{i \in A_j} t_{ij}) \mid i \in A_j\}$;

10 $t_{\ell j} \leftarrow t_{\ell j} + T_j - \sum_{i \in A_j} t_{ij}$;

11 $I_{\text{unproc}} \leftarrow \{1, \ldots, N\} \setminus (A_1 \cup \ldots \cup A_m)$;

within $(2+\epsilon)$-factor of optimal with respect to this column (bin). Moreover, the best possible allocation with respect to other entries can contribute at most $\sum_{i=1}^{N} f_{i1}(t^*_{i1})$ value to the objective function. Consequently, the solution obtained from Algorithm 5.2 is within $(3+\epsilon)$-factor with respect to performance matrix E^1. Since the solution is within $(3+\epsilon)$-factor of optimal with respect to both E^1 and E^2, it follows that the solution is within $(3+\epsilon)$-factor of optimal with respect to $E^1 + E^2$ (see Theorem 2.1 in [27]). The performance improvement heuristic further improves the value of the objective function and only improves the factor of optimality. This establishes the first statement.

The second statement follows immediately from the observation that Algorithm 5.2 solves $2M$ instances of knapsack problem with sigmoid utility using Algorithm 5.1. □

Example 5.2. Consider the GAP with $M = 4$ and $N = 10$. Let the associated sigmoid functions be $f_{ij}(t) = 1/(1+\exp(-t+b_{ij}))$, where the matrix of parameters b_{ij} is

$$\boldsymbol{b} = \begin{bmatrix} 1 & 7 & 2 & 3 & 8 & 7 & 5 & 1 & 3 & 6 \\ 7 & 9 & 8 & 8 & 6 & 1 & 7 & 4 & 5 & 4 \\ 6 & 10 & 1 & 2 & 3 & 1 & 9 & 7 & 9 & 5 \\ 9 & 2 & 4 & 8 & 1 & 2 & 5 & 8 & 6 & 8 \end{bmatrix}.$$

Let the vector of total resource at each bin be $\boldsymbol{T} = [5\ 10\ 15\ 20]$. The optimal allocations obtained through Algorithm 5.2 in Figure 5.5. The corresponding assignment sets are $A_1 = \{8\}, A_2 = \{10\}, A_3 = \{1, 3, 4, 5\}$, and $A_4 = \{2, 6, 7, 9\}$.

□

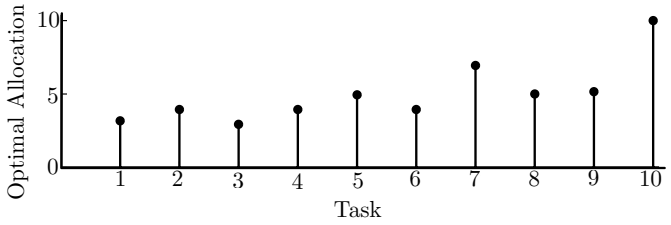


Figure 5.5: Allocations for the GAP obtained via Algorithm 5.2.

5.4 Bin-packing Problem with Sigmoid Utility

In this section, we consider the bin-packing problem (BPP) with sigmoid utility. We first define the problem and then develop an approximation algorithm for it.

5.4.1 BPP with Sigmoid Utility: Problem Description

Consider N sigmoid functions $f_\ell, \ell \in \{1, \ldots, N\}$, and a resource $T \in \mathbb{R}_{>0}$. Determine the minimum $K \in \mathbb{N}$ and a mapping $\Upsilon : \{1, \ldots, N\} \to \{1, \ldots, K\}$ such that, for each $i \in \{1, \ldots, K\}$, the optimal solution to the KP with sigmoid

utility

$$\begin{aligned} \text{maximize} \quad & \sum_{\ell \in \mathcal{A}_i} f_\ell(t_\ell) \\ \text{subject to} \quad & \sum_{\ell \in \mathcal{A}_i} t_\ell \leq T, \end{aligned} \tag{5.19}$$

where $\mathcal{A}_i = \{j \mid \Upsilon(j) = i\}$, allocates non-zero resource to each sigmoid function $f_\ell, \ell \in \mathcal{A}_i$.

The BPP with sigmoid utility models a situation where one needs to determine the minimum number of identical operators, each working for time T, required to optimally serve each of the N tasks characterized by functions $f_\ell, \ell \in \{1, \ldots, N\}$.

We denote the critical penalty rate for sigmoid function f_ℓ by $\psi_\ell, \ell \in \{1, \ldots, N\}$, and let $\psi_{\min} = \min\{\psi_\ell \mid \ell \in \{1, \ldots, N\}\}$. We now state a relevant property of the solution to the KP with sigmoid utility:

Lemma 5.8 (***Non-zero allocations***)**.** A solution to the optimization problem (5.19) allocates non-zero resource to each sigmoid function f_ℓ, $\ell \in \mathcal{A}_i, i \in \{1, \ldots, K\}$, if

$$T \geq \sum_{\ell \in \mathcal{A}_i} f_\ell^\dagger(\psi_{\min}).$$

Proof. It suffices to prove that if $T = \sum_{\ell \in \mathcal{A}_i} f_\ell^\dagger(\psi_{\min})$, then $\psi_{\min}$ is the optimal Lagrange multiplier α^* in Algorithm 5.1. Note that if a non-zero duration is allocated to each task, then the solution obtained from Algorithm 5.1 is the optimal solution. Since, $t_\ell^* = f_\ell^\dagger(\psi_{\min}), \ell \in \mathcal{A}_i$ are feasible non-zero allocations, $\psi_{\min}$ is a

Lagrange multiplier. We now prove that $\psi_{\min}$ is the optimal Lagrange multiplier. Let $\mathcal{A}_i = \{1, \ldots, a_i\}$. By contradiction, assume that $\boldsymbol{t}^*$ is not the globally optimal allocation. Without loss of generality, we assume that the global optimal policy allocates zero resource to sigmoid function f_{a_i}, and let $\bar{\boldsymbol{t}}$ be the globally optimal allocation. We observe that

$$\begin{aligned}
&\sum_{\ell=1}^{a_i-1} f_\ell(\bar{t}_\ell) + f_{a_i}(0) \\
&\quad \leq \sum_{\ell=1}^{a_i-1} f_\ell(\bar{t}_\ell) + f_{a_i}(t^*_{a_i}) - \psi_{\min} t^*_{a_i} && (5.20) \\
&\quad \leq \sum_{\ell=1}^{a_i} f_\ell(t^*_\ell) + \sum_{\ell=1}^{a_i-1} f'_\ell(t^*_\ell)(\bar{t}_\ell - t^*_\ell) - \psi_{\min} t^*_{a_i} && (5.21) \\
&\quad = \sum_{\ell=1}^{a_i} f_\ell(t^*_\ell) + \sum_{\ell=1}^{a_i} \psi_{\min}(\bar{t}_\ell - t^*_\ell) \\
&\quad = \sum_{\ell=1}^{a_i} f_\ell(t^*_\ell),
\end{aligned}$$

where inequalities (5.20) and (5.21) follow from the definition of critical penalty and the concavity to the sigmoid function at t^*_ℓ, respectively. This contradicts our assumption. Hence, $\boldsymbol{t}^*$ is the global optimal allocation and this completes the proof. □

We now state the following result about the hardness of the BPP with sigmoid utility:

Proposition 5.9 (***Hardness of BPP with sigmoid utility***)**.** The BPP with sigmoid utility is NP-hard, unless $P = NP$.

Proof. Consider an instance of standard BPP with items of size $a_i \leq T, i \in \{1, \ldots, N\}$ and bins of size T. It is well known [57] that BPP is NP-hard. Without loss of generality, we can pick N sigmoid functions $f_i, i \in \{1, \ldots, N\}$ such that $f_i^{\dagger}(\psi_{\min}) = a_i$, for each $i \in \{1, \ldots, N\}$ and some $\psi_{\min} \in \mathbb{R}_{>0}$. It follows from Lemma 5.8 that such an instance of BPP with sigmoid utility is in one-to-one correspondence with the aforementioned standard BPP. This establishes the statement. □

5.4.2 BPP with Sigmoid Utility: Approximation Algorithm

We now develop an approximation algorithm for BPP with sigmoid utility. The proposed algorithm is similar to the standard next-fit algorithm [57] for the binary bin-packing problem, and adds a sigmoid function to a bin if optimal policy for the associated KP with sigmoid utility allocates non-zero resource to each sigmoid function; otherwise, it opens a new bin. The approximation algorithm is presented in Algorithm 5.4. We now present a formal analysis of this algorithm. We introduce following notations. Let K^* be the optimal solution of the bin-packing problem with sigmoid utility, and $K_{\text{next-fit}}$ be the solution obtained through Algorithm 5.4. We first present the following important property of optimization problem (5.19).

Theorem 5.10 (***BPP with sigmoid utility***)**.** The following statements hold for the BPP with sigmoid utility (5.19), and its solution obtained via Algorithm 5.4:

(i). the optimal solution satisfies the following bounds

$$K_{\text{next-fit}} \geq K^* \geq \frac{1}{T}\sum_{\ell=1}^{N} \min\{T, t_\ell^{\text{inf}}\}.$$

(ii). the solution obtained through Algorithm 5.4 satisfies

$$K_{\text{next-fit}} \leq \frac{1}{T}\left(2\sum_{\ell=1}^{N} f_\ell^\dagger(\psi_{\min}) - 1\right).$$

(iii). the Algorithm 5.4 provides a solution to the BPP with sigmoid utility within a factor of optimality

$$\frac{\max\{2f_\ell^\dagger(\psi_{\min}) \mid \ell \in \{1,\ldots,N\}\}}{\max\{\min\{T, t_\ell^{\text{inf}}\} \mid \ell \in \{1,\ldots,N\}\}};$$

(iv). the Algorithm 5.4 runs in $O(N^3)$ time, provided the solution to KP with sigmoid utility can be computed in $O(N^2)$ time.

Proof. It follows from Algorithm 5.1 that if $t_\ell^{\text{inf}} < T$, then the optimal non-zero allocation to sigmoid function f_ℓ is greater than t_ℓ^{inf}. Otherwise, the optimal non-zero allocation is equal to T. Therefore, if each sigmoid function gets a non-zero allocation under the optimal policy, then at least $\sum_{\ell=1}^{N} \min\{T, t_\ell^{\text{inf}}\}$ resource is required, and the lower bound on the optimal K^* follows.

It follows from Lemma 5.8 that if resource $t_\ell = f_\ell^\dagger(\psi_{\min})$ is available for task $\ell \in \{1, \ldots, N\}$, then a non-zero resource is allocated to it. Therefore, the solution of the bin-packing problem with bin size T and items of size $\{f_\ell^\dagger(\psi_{\min}) \mid \ell \in \{1, \ldots, N\}\}$ provides an upper bound to the solution of the BPP with sigmoid utility. The upper bound to the solution of this bin-packing problem obtained through the standard next-fit algorithm is $(2\sum_{\ell=1}^{N} f_\ell^\dagger(\psi_{\min}) - 1)/T$, and this completes the proof of the second statement.

The third statement follows immediately from the first two statements, and the last statement follows immediately from the fact the Algorithm 5.1 is utilized at each iteration of the Algorithm. □

Example 5.3. For the same set of sigmoid functions as in Example 5.1 and $T = 20$ units, the solution to the BPP with sigmoid utility obtained through Algorithm 5.4 requires $K_{\text{next-fit}} = 3$ bins, and the optimal allocations to each task in these bins are shown in Figure 5.6. □

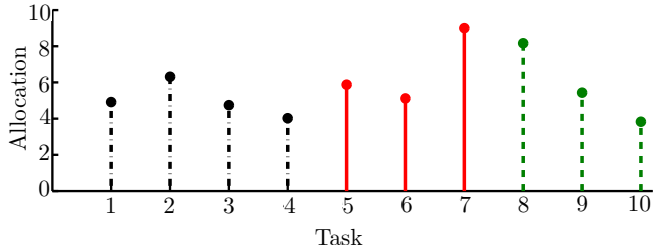


Figure 5.6: Allocations to sigmoid functions in each bin. The dot-dashed black lines represent tasks allocated to first bin, the solid red lines represent tasks allocated to second bin, and the dashed green line represent tasks allocated to third bin.

5.5 Conclusions and Future Directions

We studied non-convex optimization problems involving sigmoid functions. We considered the maximization of a sigmoid function subject to a linear penalty and showed that the optimal allocation jumps down to zero at a critical penalty rate. This jump in the allocation imparts combinatorial effects to the constrained optimization problems involving sigmoid functions. We studied three such problems, namely, the knapsack problem with sigmoid utility, the generalized assignment problem with sigmoid utility, and the bin-packing problem with sigmoid utility. We merged approximation algorithms from discrete optimization with algorithm from continuous optimization to develop hybrid approximation algorithms for these problems.

There are many possible extensions of this work. A similar strategy for approximate optimization could be adopted for other problems involving sigmoid functions, e.g., the network utility maximization problem, where the utility of each source is a sigmoid function. Other extensions include problems involving general non-convex functions and optimization in queues with sigmoid characteristics.

Algorithm 5.3: *Function* $\texttt{GAP}(j, F^{(j)})$

% *Function definition*

function $[A^{(j)}, \boldsymbol{t}^{(j)}] \leftarrow \texttt{GAP}(j, F^{(j)})$

% *Determine allocations for bin* j *using Algorithm 5.1*

$[\bar{A}, \bar{\boldsymbol{t}}] \leftarrow \texttt{KP}(F^{(j)}_{*1}, T_j)$;

foreach $i \in \{1, \ldots, N\}$ *and* $k \in \{1, \ldots, M - j + 1\}$ **do**

$$E^1_{ik}(t) \leftarrow \begin{cases} F^{(j)}_{i1}(\bar{t}_i), & \text{if } i \in \bar{A} \text{ and } k \neq 1, \\ F^{(j)}_{i1}(t), & \text{if } k = 1, \\ 0, & \text{otherwise;} \end{cases}$$

$E^2(t) \leftarrow F^{(j)}(t) - E^1(t)$;

if $j < M$ **then**

% *remove first column from* E^2 *and assign it to* $F^{(j+1)}$

$F^{(j+1)} \leftarrow E^2_{*2:M-j+1}$;

$[A^{(j+1)}, \boldsymbol{t}^{(j+1)}] \leftarrow \texttt{GAP}(j + 1, F^{(j+1)})$;

$A_j \leftarrow \bar{A} \setminus \cup_{i=j+1}^{M} A_i$;

$A^{(j)} \leftarrow A_j \cup A^{(j+1)}$;

foreach $i \in \bar{A} \cap \cup_{i=j+1}^{M} A_i$ **do**

$\bar{t}_i \leftarrow 0$;

$\boldsymbol{t}^{(j)} \leftarrow [\bar{\boldsymbol{t}} \quad \boldsymbol{t}^{(j+1)}]$;

else

$A_j \leftarrow \bar{A}$ and $\boldsymbol{t}^{(j)} \leftarrow \bar{\boldsymbol{t}}$;

Algorithm 5.4: *BPP with Sigmoid Utility: Approximation Algorithm*

Input : $f_\ell, \ell \in \{1, \ldots, N\}, T \in \mathbb{R}_{>0}$;

Output : number of required bins $K \in \mathbb{N}$ and assignments Υ;

$K \leftarrow 1; \mathcal{A}_K \leftarrow \{\}$;

foreach $\ell \in \{1, \ldots, N\}$ **do**

  $\mathcal{A}_K \leftarrow \mathcal{A}_K \cup \{\ell\}$;

  solve problem (5.19) for $i = K$, and find $\boldsymbol{t}^*$;

  % if optimal policy drops a task, open a new bin

  if $t_j^* = 0$, *for some* $j \in \mathcal{A}_K$ **then**

    $K \leftarrow K + 1$; $\mathcal{A}_K \leftarrow \{\ell\}$;

  $\Upsilon(\ell) \leftarrow K$;

Chapter 6

Attention Allocation in Decision Making Queues

In this chapter we consider the problem of optimal time duration allocation for a human operator. The decision making tasks arrive at a given rate and are stacked in a queue. A human operator processes these tasks on the *first-come first-serve* basis (see Figure 6.1.) The human operator receives a unit reward for the correct decision, while there is no penalty for a wrong decision. We assume that the tasks can be parametrized by some variable and the variable takes value in a finite set $\mathcal{D} \subseteq \mathbb{R}$. Let the performance of the operator on a task with parameter $d \in \mathcal{D}$ be a function $f_d : \mathbb{R}_{\geq 0} \to [0, 1)$ of the duration operator allocates to the task. As discussed in Section 2.3, the human performance function f_d is well modeled by a sigmoid function. In this chapter we only consider smooth sigmoid functions.

We study two particular problems. First, in Section 6.1, we consider a static queue with latency penalty, that is, the scenario where the human operator has to perform $N \in \mathbb{N}$ decision making tasks, but each task loses value at a constant rate per unit delay in its processing. Second, in Sections 6.2 and 6.3 we consider a dynamic queue of decision making tasks where each task loses value at a constant rate per unit delay in its processing. The loss in the value of a task may occur due to the processing deadline on the task. In other words, the latency penalty is a soft constraint that captures the processing deadline on the task. For such a decision making queue, we are interested in the optimal time-duration allocation to each task. Alternatively, we are interested in the arrival rate that will result in the desired accuracy for each task. We intend to design a decision support system that tells the human operator the optimal time-duration allocation to each task.

We remark that the processing deadlines on the tasks can be incorporated as hard constraints as well, but the resulting optimization problem is combinatorially hard. For instance, if the performance of the human operator is modeled by a step function with the jump at the inflection point and the deadlines are incorporated as hard constraints, then the resulting optimization problem is equivalent to the N-dimensional knapsack problem [54]. The N-dimensional knapsack problem is NP-hard and admits no fully polynomial time approximation algorithm for $N \geq 2$. The standard [54] approximation algorithm for this problem has factor

of optimality $N + 1$ and hence, for large N, may yield results very far from the optimal. The close connections between the knapsack problems with step functions and sigmoid functions (see Chapter 5) suggest that efficient approximation algorithms may not exist for the problem formulation where processing deadlines are modeled as hard constraints.

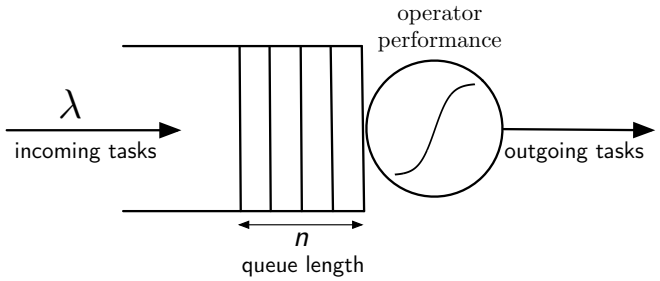


Figure 6.1: Problem setup. The decision making tasks arrive at a rate λ. These tasks are served by a human operator with sigmoid performance. Each task loses value while waiting in the queue.

6.1 Static queue with latency penalty

6.1.1 Problem description

Consider that the human operator has to perform $N \in \mathbb{N}$ decision making tasks in a prescribed order (task labeled "1" should be processed first, etc.) Let the human operator allocate duration t_ℓ to the task $\ell \in \{1, \ldots, N\}$. Let the difficulty of the task ℓ be $d_\ell \in \mathcal{D}$. According to the importance of the task, a weight $w_\ell \in \mathbb{R}_{\geq 0}$ is assigned to the task ℓ. The operator receives a reward $w_\ell f_{d_\ell}(t_\ell)$ for

allocating duration t_ℓ to the task ℓ, while they incur a latency penalty c_ℓ per unit time for the delay in its processing. The objective of the human operator is to maximize their average benefit and the associated optimization problem is:

$$\underset{\boldsymbol{t}\in\mathbb{R}^N_{\geq 0}}{\text{maximize}} \quad \frac{1}{N}\sum_{\ell=1}^{N}\big(w_\ell f_{d_\ell}(t_\ell) - (c_\ell + \cdots + c_N)t_\ell\big), \tag{6.1}$$

where $\boldsymbol{t} = \{t_1, \ldots, t_N\}$ is the duration allocation vector.

6.1.2 Optimal solution

We can now analyze optimization problem (6.1).

Theorem 6.1 (***Static queue with latency penalty***)**.** For the optimization problem (6.1), the optimal allocation to task $\ell \in \{1, \ldots, N\}$ is

$$t^*_\ell \in \operatorname{argmax}\big\{w_\ell f_{d_\ell}(\beta) - (c_\ell + \cdots + c_N)\beta \;\big|\; \beta \in \{0, f^\dagger_{d_\ell}((c_\ell + \ldots + c_N)/w_\ell)\}\big\}.$$

Proof. The proof is similar to the proof of Lemma 5.1. □

Remark 6.1 (***Comparison with a concave utility***)**.** The optimal duration allocation for the static queue with latency penalty decreases to a critical value with increasing penalty rate, then jumps down to zero. In contrast, if the performance function is concave instead of sigmoid, then the optimal duration allocation decreases continuously to zero with increasing penalty rate. □

6.1.3 Numerical Illustrations

We now present an example to elucidate on the ideas presented in this section.

Example 6.1 (***Static queue and heterogeneous tasks***)**.** The human operator has to serve $N = 10$ heterogeneous tasks and receives an expected reward $f_{d_\ell}(t) = 1/(1 + \exp(-a_\ell t + b_\ell))$ for an allocation of duration t secs to task ℓ, where d_ℓ is characterized by the pair (a_ℓ, b_ℓ). The following are the parameters and the weights associated with each task:

$$(a_1, \ldots, a_N) = (1, 2, 1, 3, 2, 4, 1, 5, 3, 6),$$

$$(b_1, \ldots, b_N) = (5, 10, 3, 9, 8, 16, 6, 30, 6, 12), \text{ and}$$

$$(w_1, \ldots, w_N) = (2, 5, 7, 4, 9, 3, 5, 10, 13, 6).$$

Let the vector of penalty rates be

$$\boldsymbol{c} = (0.09, 0.21, 0.21, 0.06, 0.03, 0.15, 0.3, 0.09, 0.18, 0.06)$$

per second. The optimal allocations are shown in Figure 6.1.3. The importance and difficulty level of a task are encoded in the associated weight and the inflection point of the associated sigmoid function, respectively. The optimal allocations depend on the difficulty level, the penalty rate, and the importance of the tasks. For instance, task 6 is a relatively simple but less important task and is dropped. On the contrary, task 8 is a relatively difficult but very important task and is processed. □

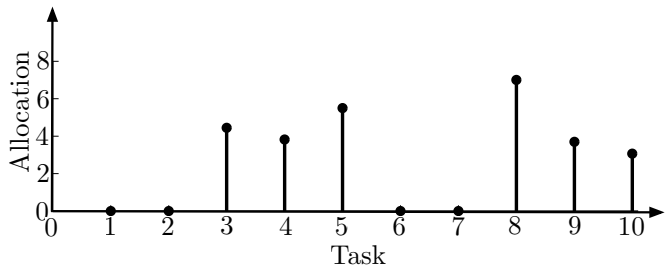


Figure 6.2: Static queue with latency penalty. The optimal allocations depends of the difficulty level, the penalty rate and the importance of the tasks.

6.2 Dynamic queue with latency penalty

In the previous section, we developed policies for static queue with latency penalty. We now consider dynamic queue with latency penalty, that is, the scenario where the tasks arrive according to a stochastic process and wait in a queue to get processed. We assume the tasks lose value while waiting in the queue. The operator's objective is to maximize their infinite horizon reward. In the following we pose the problem as an MDP and study its properties.

6.2.1 Problem description

We study the optimal policies for the human operator serving a queue of decision making tasks. We now define various components of the problem:

Description of Tasks: We make following assumptions on the decision making tasks: (i) tasks arrive according to Poisson process with rate $\lambda \in \mathbb{R}_{>0}$; (ii) each

task is parameterized by a variable $d \in \mathcal{D}$, where $\mathcal{D}$ is a finite set of parameters for the task; (iii) a task with parameter $d \in \mathcal{D}$ is characterized by a triplet of operator's performance function f_d, the latency penalty rate c_d, and the weight w_d assigned to the task; (iii) the parameter associated with each task is sampled from a probability distribution function $p : \mathcal{D} \to [0, 1]$. Let the realized parameter for task $\ell \in \mathbb{N}$ be d_ℓ. Thus, the operator receives a compensation $w_{d_\ell} f_{d_\ell}(t_\ell)$ for a duration allocation t_ℓ to task ℓ, while they incur a latency penalty c_{d_ℓ} per unit time for the delay in its processing. The objective of the operator is to maximize their infinite horizon expected reward. To this end, the support system suggests the optimal time duration that the human operator should allocate to a given task. We assume that such time-duration is suggested at the start of a stage and is not modified during the stage. We now formulate this optimization problem as an MDP, namely, Γ.

Description of MDP Γ: Let the stage $\ell \in \mathbb{N}$ of the MDP Γ be initiated with the processing of task ℓ. We now define the elements of the MDP Γ:

(i) *Action and State variables:* We choose the action variable at stage ℓ as the time-duration to be allocated to task ℓ, denoted by $t_\ell \in \mathbb{R}_{\geq 0}$. We choose the state variable at stage ℓ as the vector of parameters $\boldsymbol{d}_\ell \in \mathcal{D}^{n_\ell}$ associated with each task in the queue, where $n_\ell \in \mathbb{N}$ is the queue length at stage ℓ. Note that the definition of the stage and the state variable are consistent under the following assumption:

Assumption 6.1 (***Non-empty queue***)**.** *Without loss of generality, we assume that the queue is never empty. If queue is empty at some stage, then the operator waits for the next task to arrive, and there is no penalty for such waiting time.*□

(ii) *Reward Structure:* We define the reward $r : \mathcal{D}^{n_\ell} \times \mathbb{R}_{\geq 0} \to \mathbb{R}$ obtained by allocating duration t to the task ℓ by

$$r(\boldsymbol{d}_\ell, t) = w_{d_\ell} f_{d_\ell}(t) - \frac{1}{2}\Big(\sum_{i=\ell}^{\ell+n_\ell-1} c_{d_i} + \sum_{j=\ell}^{\ell+n'_\ell-1} c_{d_j} \Big) t,$$

where $\boldsymbol{d}_\ell \in \mathcal{D}^{n'_\ell}$ is the vector of penalty rates for the tasks in the queue and n'_ℓ is the queue length just before the end of stage ℓ.

Note that the queue length while a task is processed may not be constant, therefore, the latency penalty is computed as the average of the latency penalty for the tasks present at the start of processing the task and the latency penalty for the tasks present at the end of processing the task. Such averaging is consistent with the expected number of arrivals being a linear function of time for Poisson process.

(iii) *Value function:*

The MDP with finite horizon length $N \in \mathbb{N}$ maximizes the value function $V_N : \mathcal{D}^{n_1} \times \mathcal{B}(\{1, \ldots, N\} \times \mathcal{D}^\infty, \mathbb{R}_{\geq 0}) \to \mathbb{R}$ defined by

$$V_N(\boldsymbol{d}_1, \boldsymbol{t}^{\text{finite}}) = \sum_{\ell=1}^{N} \mathbb{E}[r(\boldsymbol{d}_\ell, t^{\text{finite}}(\ell, \boldsymbol{d}_\ell))],$$

where $n_1 \in \mathbb{N}$ is the initial queue length, $\mathcal{D}^\infty = \cup_{i\in\mathbb{N}} \mathcal{D}^i$, and $\mathcal{B}(\{1,\ldots,N\} \times \mathcal{D}^\infty, \mathbb{R}_{\geq 0})$ is the space of bounded below functions defined from $\{1,\ldots,N\} \times \mathcal{D}^\infty$ to $\mathbb{R}_{\geq 0}$. $\mathcal{B}(\{1,\ldots,N\} \times \mathcal{D}^\infty, \mathbb{R}_{\geq 0})$ represents the space of policies, that is, the duration allocation as a function of stage and state. We will focus on stationary policies and for stationary policies, the policy space reduces to $\mathcal{B}(\mathcal{D}^\infty, \mathbb{R}_{\geq 0})$

Under a stationary policy $\boldsymbol{t}^{\text{stat}}$, the infinite horizon average value function of the MDP $V_{\text{avg}} : \mathcal{D}^{n_1} \times \mathcal{B}(\mathcal{D}^\infty, \mathbb{R}_{\geq 0}) \to \mathbb{R}$ is defined by

$$V_{\text{avg}}(\boldsymbol{d}_1, \boldsymbol{t}^{\text{stat}}) = \lim_{N\to+\infty} \frac{1}{N} V_N(\boldsymbol{d}_1, \boldsymbol{t}^{\text{stat}}).$$

We also define the infinite horizon discounted value function $V_\alpha : \mathcal{D}^{n_1} \times \mathcal{B}(\mathcal{D}^\infty, \mathbb{R}_{\geq 0}) \to \mathbb{R}$ by

$$V_\alpha(\boldsymbol{d}_1, \boldsymbol{t}^{\text{stat}}) = \sum_{\ell=1}^{+\infty} \alpha^{\ell-1} \mathbb{E}[r(\boldsymbol{d}_\ell, t^{\text{stat}}(\boldsymbol{d}_\ell))],$$

where $\alpha \in (0,1)$ is the discount factor.

6.2.2 Properties of optimal solution

We now study some properties of the MDP Γ and its solution. Let $V_\alpha^* : \mathcal{D}^{n_1} \to \mathbb{R}_{\geq 0}$ denote the optimal infinite horizon α-discounted value function. We also define $N_{\max} = \lfloor \max\{w_d \psi_{f_d}/c_d \mid d \in \mathcal{D}\} \rfloor$.

Lemma 6.2 (***Properties of MDP*** Γ)**.** The following statements hold for the MDP Γ and its infinite horizon average value function:

(i). there exists a solution to the MDP Γ;

(ii). an optimal stationary policy allocates zero duration to the task ℓ if $n_\ell > N_{\max}$;

Proof. It can be verified that the conditions presented in Section 2.1 hold for MDP Γ and the optimal discounted value function exists. To prove the existence of a solution to the the average value formulation of the MDP Γ, we note that

$$V_\alpha(\boldsymbol{d}_1, \boldsymbol{t}) = \sum_{\ell=1}^{+\infty} \alpha^{\ell-1} \mathbb{E}[r(\boldsymbol{d}_\ell, t_\ell)] \leq \frac{w_{\max} - c_{\min}}{(1-\alpha)},$$

for each $\boldsymbol{d}_1 \in \mathcal{D}^{n_1}$, where n_1 is initial queue length, $w_{\max} = \max\{w_d \mid d \in \mathcal{D}\}$ and $c_{\min} = \min\{c_d \mid d \in \mathcal{D}\}$. Therefore, $V_\alpha^*(\boldsymbol{d}_1) \leq (w_{\max} - c_{\min})/(1-\alpha)$. Moreover, $V_\alpha^*(\boldsymbol{d}_1) \geq V_\alpha(\boldsymbol{d}_1, \boldsymbol{0}) = 0$. Hence, $|V_\alpha^*(\boldsymbol{d}_1) - V_\alpha^*(\boldsymbol{d}_0)| \leq 2|w_{\max} - c_{\min}|/(1-\alpha)$, for any $\boldsymbol{d}_0 \in \mathcal{D}^{n_0}, n_0 \in \mathbb{N}$. Thus, the conditions of Theorem 5.2 in [3] hold and this establishes the first statement.

We now establish the second statement. We note that for a state associated with queue length $n > N_{\max}$, the reward is non-positive and is zero only if the allocation at that stage is zero. Moreover, for a Poisson arrival process, the probability that the queue length is non-decreasing increases with the allocation at current stage. Thus a positive allocation increases the probability of non-positive reward at future stages. Therefore, a zero duration allocation for $n > N_{\max}$ maximizes the reward at current stage and maximizes the probability of getting

positive rewards at future stages. Consequently, the optimal stationary policy allocates zero duration for a queue length greater than $N_{\max}$. □

6.3 Receding Horizon Solution to dynamic queue with latency penalty

We rely on the certainty-equivalent receding horizon framework [9, 23, 62] to approximately solve the MDP Γ. In the certainty-equivalent approximation, the future uncertainties are replaced with their expected values [9]. For an allocation of duration t_ℓ at stage ℓ, the expected number of arrivals for a Poisson process with rate λ is λt_ℓ. Accordingly, the evolution of the queue length under certainty-equivalent approximation is

$$\bar{n}_{\ell+1} = \max\{1, \bar{n}_\ell - 1 + \lambda t_\ell\},$$

where $\bar{n}_\ell$ represents predicted queue length at stage ℓ under certainty-equivalent approximation, and $\bar{n}_1 = n_1$. The certainty-equivalent approximation also replaces the parameters of tasks that have not yet arrived by their expected values, and accordingly, assigns them the expected performance function $\bar{f} : \mathbb{R}_{\geq 0} \to [0, 1)$, the expected importance $\bar{w}$, and the expected latency penalty $\bar{c}$ defined by

$$\bar{f}(t) = \frac{1}{\bar{w}} \mathbb{E}_p[w_d f_d(t)],$$

$\bar{w} = \mathbb{E}_p[w_d]$, and $\bar{c} = \mathbb{E}_p[c_d]$, respectively, where $\mathbb{E}_p[\cdot]$ represents the expected value with respect to the measure p.

The receding horizon framework solves a finite horizon optimization problem at each iteration. We denote the receding horizon policy that solves a N-horizon certainty-equivalent problem at each stage by N-RH policy. We now study such certainty-equivalent finite horizon optimization problem.

6.3.1 Certainty-equivalent finite horizon optimization

We now study the finite horizon optimization problem with horizon length N that the certainty-equivalent receding horizon policy solves at each iteration. Given horizon length N, current queue length n_ℓ, the realization of the sigmoid functions $f_1, \ldots, f_{n_\ell}$, the associated latency penalties $c_1, \ldots, c_{n_\ell}$ and the importance levels $w_1, \ldots, w_{n_\ell}$. In certainty-equivalent problem, the true parameters of the tasks are used for the tasks that have already arrived, while the expected values of the parameters are used for the tasks that have not yet arrived. In particular, if current queue length is less than the horizon length, i.e., $n_\ell < N$, then we define the reward associated with task $j \in \{1, \ldots, N\}$ by

$$r_j = \begin{cases} r_j^{\text{rlzd}}, & \text{if } 1 \leq j \leq n_\ell, \\ r_j^{\text{exp}}, & \text{if } n_\ell + 1 \leq j \leq N, \end{cases} \tag{6.2}$$

where $r_j^{\text{rlzd}} = w_j f_j(t_j) - (\sum_{i=j}^{n_\ell} c_i + (\bar{n}_j - n_\ell - j + 1)\bar{c})t_j - \bar{c}\lambda t_j^2/2$ is the reward computed using the realized parameters, and $r_j^{\text{exp}} = \bar{w}\bar{f}(t_j) - \bar{c}(\bar{n}_\ell - j + 1)t_j - \bar{c}\lambda t_j^2/2$ is the reward computed using the expected values of the parameters. If the current queue length is greater than the horizon length, i.e., $n_\ell \geq N$, then we define all the reward using realized parameters, i.e., $r_j = r_j^{\text{rlzd}}$, for each $j \in \{1, \ldots, N\}$.

$$\begin{aligned} \underset{\boldsymbol{t} \succeq 0}{\text{maximize}} \quad & \frac{1}{N} \sum_{j=1}^{N} r_j \\ \text{subject to} \quad & \bar{n}_{j+1} = \max\{1, \bar{n}_j - 1 + \lambda t_j\},\ \bar{n}_1 = n_\ell, \end{aligned} \tag{6.3}$$

where $\boldsymbol{t} = \{t_1, \ldots, t_N\}$ is the duration allocation vector.

The optimization problem (6.3) is difficult to handle analytically. For the special case in which tasks are identical, we provide a procedure to determine the exact solution to problem (6.3) in the Appendix of this chapter. This procedure also provides insights into the implication of sigmoid performance function on the optimal policy. For the general case, we resort to the discretization of the action and the state space and utilize the backward induction algorithm to approximately solve the dynamic program (6.3). Let us define maximum allocation to any task $\tau^{\max} = \max\{f_d^\dagger(c_d/w_d) \mid d \in \mathcal{D}\}$. We now state the following results on the efficiency of discretization:

Lemma 6.3 (***Discretization of state and action space***)**.** For the optimization problem (6.3) and the discretization of the action and the state space with a uniform grid of width $\epsilon > 0$, the following statements hold:

(i). the state space and the action space can be restricted to compact spaces $[1, N_{\max} + 1]$, and $[0, \tau^{\max}]$, respectively;

(ii). the policy obtained through the discretized state and action space in within $O(\epsilon)$ of optimal;

(iii). the computational complexity of the solution is $O(N/\epsilon^2)$.

Proof. It follows from Lemma 5.1 that for $\bar{n}_j > \max\{w_d \psi_{f_d}/c_{\min} \mid d \in \mathcal{D}\}$, $r_j + \bar{c}\lambda t_j^2/2$ achieves its global maximum at $t_j = 0$. Hence, for $\bar{n}_j > \max\{w_d \psi_{f_d}/c_{\min} \mid d \in \mathcal{D}\}$, r_j achieves its global maximum at $t_j = 0$. Moreover, the certainty-equivalent queue length at a stage $k > j$ is a non-decreasing function of of the allocation at stage j. Thus, the reward at stage $k > j$ decreases with allocation t_j. Therefore, the optimal policy for the optimization problem (6.3) allocates zero duration at stage j if $\bar{n}_j > \max\{w_d \psi_{f_d}/c_{\min} \mid d \in \mathcal{D}\}$, and subsequently, the queue length decreases by unity at next stage. Thus, any certainty-equivalent queue length greater than $\max\{w_d \psi_{f_d}/c_{\min} \mid d \in \mathcal{D}\}$ can be mapped to $N_{\max} + \texttt{frac}$, where `frac` is the fractional part of the certainty-equivalent queue length. Consequently, the state space can be restricted to the compact set $[1, N_{\max} + 1]$. Similarly, for

$t_j > \tau^{\max}$, the reward at stage j in optimization problem (6.3) is a decreasing function of the allocation t_j, and the rewards at stages $k > j$ are decreasing function of allocation t_j. Therefore, the allocation to each task is less than $\tau^{\max}$. This completes the proof of the first statement.

Since the action variable and the state variable in problem (6.3) belong to compact sets and the reward function and the state evolution function is Lipschitz, it follows from Section 2.1 that the value function obtained using the discretized action and state space is within $O(\epsilon)$ of the optimal value function. This establishes the second statement.

The third statement is an immediate consequence of the fact that computational complexity of a finite horizon dynamic program is the sum over stages of the product of cardinalities of the state space and the action space in each stage. □

6.3.2 Performance of receding horizon algorithm

We now derive performance bounds on the receding horizon procedure. First, we determine a global upper bound on the performance of any policy for the MDP Γ. Then, we develop a lower bound on the performance of 1-RH policy. Without loss of generality, we assume that the initial queue length is unity. If the initial queue length is non-unity, then we drop tasks till queue length is unity. Note that this does not affect the infinite horizon average value function. We also assume

that the latency penalty is small enough to ensure an optimal non-zero duration allocation if only one task is present in the queue, that is, $c_d \leq w_d \psi_{f_d}$, for each $d \in \mathcal{D}$. We now introduce some notation. For a given λ and $d \in \mathcal{D}$, define

$$c_d^{\text{crit}} = \min \big\{ c \in \mathbb{R}_{>0} \mid \text{argmax}\{ t \in \mathbb{R}_{\geq 0} \mid w_d f_d(t) - ct - \bar{c}\lambda t^2/2 \} = 0 \big\}.$$

Moreover, let $c_d^{\max}$ be the optimal value of the following optimization problem:

$$\begin{aligned} \underset{x_{id} \in \{0,1\}}{\text{maximize}} \quad & \sum_{i=1}^{N_{\max}-1} \sum_{d \in \mathcal{D}} c_d x_{id} \\ \text{subject to} \quad & \sum_{i=1}^{N_{\max}-1} \sum_{d \in \mathcal{D}} c_d x_{id} < c_d^{\text{crit}} - c_d \\ & \sum_{d \in \mathcal{D}} x_{id} \leq 1, \text{ for each } i \in \{1, \ldots, N_{\max} - 1\}. \end{aligned}$$

Let $t_d^{\text{crit}} = \text{argmax}\{w_d f_d(t) - (c_d + c_d^{\max})t - \bar{c}\lambda t^2/2 \mid t \in \mathbb{R}_{\geq 0}\}$, and $\tau_d^{\max} = f_d^{\dagger}(c_d/w_d)$, for each $d \in \mathcal{D}$. Let $\boldsymbol{t}^{\text{unit}}$ be the 1-RH policy.

Theorem 6.4 (***Bounds on performance***)**.** For the MDP Γ and 1-RH policy the following statements hold:

(i). the average value function satisfy the following upper bound

$$V_{\text{avg}}(\boldsymbol{d}_1, \boldsymbol{t}) \leq \max_{d \in \mathcal{D}} \big\{ w_d f_d(\tau_d^{\max}) - c_d \tau_d^{\max} \big\},$$

for each $n_1 \in \mathbb{N}$ and any policy $\boldsymbol{t}$;

(ii). the average value function satisfy the following lower bound for 1-RH policy:

$$V_{\text{avg}}(\boldsymbol{d}_1, \boldsymbol{t}^{\text{unit}}) \geq \min_{d \in \mathcal{D}} \frac{w_d f_d(t_d^{\text{crit}}) - (c_d + c_d^{\max}) t_d^{\text{crit}} - \bar{c}\lambda {t_d^{\text{crit}}}^2/2}{\lambda \tau_{\max} + e^{-\lambda \tau_{\max}}},$$

for each $n_1 \in \mathbb{N}$.

Proof. We start by establishing the first statement. We note that the reward at stage ℓ is $r(\boldsymbol{d}_\ell, t_\ell) \leq w_{d_\ell} f_{d_\ell}(t_\ell) - c_{d_\ell} t_\ell$. It follows from Lemma 5.1 that the maximum value $w_{d_\ell} f_{d_\ell}(t_\ell) - c_{d_\ell} t_\ell$ is achieved at $t_\ell = \tau_{d_\ell}^{\max}$. Therefore, the reward at each stage is upper bounded by $\max\{w_d f_d(\tau_d^{\max}) - c_d \tau_d^{\max} \mid d \in \mathcal{D}\}$ and the first statement follows.

To establish the second statement, we note that if some new tasks arrive at a stage, then the optimal policy processes at least one of these tasks or processes at least one task already in the queue. Therefore, the optimal policy processes at least one task for each set of tasks arrived. The minimum reward obtained after processing task d is $w_d f_d(t_d^{\text{crit}}) - (c_d + c_d^{\max}) t_d^{\text{crit}} - \bar{c}\lambda {t_d^{\text{crit}}}^2/2$. If n_{arr} is the number of arrived tasks, the fraction of tasks processed is $1/n_{\text{arr}}$. Under Assumption 6.1, the expected number of arrivals at stage ℓ for a Poisson process is $\mathbb{E}[n_{\text{arr}}] = \lambda t_\ell + e^{-\lambda t_\ell}\mathbb{P}(n_\ell = 1)$, where the second term corresponds to the situation when the queue becomes empty after processing task ℓ and the operator waits for new task. Since $1/n_{\text{arr}}$ is a convex function and $\mathbb{E}[n_{\text{arr}}] \leq \lambda\tau_{\max} + e^{-\lambda\tau_{\max}}$, it follows from the Jensen's inequality that the expected fraction of tasks processed is greater than $1/(\lambda\tau_{\max} + e^{-\lambda\tau_{\max}})$. This completes the proof of the last statement. □

Similar bounds can be derived for the N-RH policy. Since the future queue length increases with increasing t_1, the total penalty increases in problem (6.3)

with increasing t_1. Thus, with increasing N, the receding horizon policy drops tasks at a smaller critical penalty c_d^{crit}. Consequently, the total penalty on each task decreases and the performance improves.

6.3.3 Numerical Illustrations

We now elucidate on the concepts discussed in this section with an example.

Example 6.2 (***RH policy***)**.** Suppose that the human operator has to serve a queue of tasks with Poisson arrival at the rate λ per sec. The set of the tasks is the same as in Example 6.1 and each task is sampled uniformly from this set. 1-RH and 10-RH policies for a sample evolution of the queue at an arrival rate $\lambda = 0.5$ per second are shown in Figure 6.3 and 6.4, respectively. The sequence of tasks arriving is the same for both the policies. The RH policy tends to drop the tasks that are difficult and unimportant. The difficulty of the tasks is characterized by the inflection point of the associated sigmoid functions. The queue length under the 1-RH policy is higher than the 10-RH policy. A comparison of the RH policies is shown in Figure 6.5. We obtained these performance curves through Monte-Carlo simulations. □

Remark 6.2 (***Comparison with a concave utility***)**.** With the increasing penalty rate as well as the increasing arrival rate, the time duration allocation decreases to a critical value and then jumps down to zero for the dynamic queue with

latency penalty. In contrast, if the performance function is concave instead of sigmoid, then the duration allocation decreases continuously to zero with increasing penalty rate as well as increasing arrival rate. □

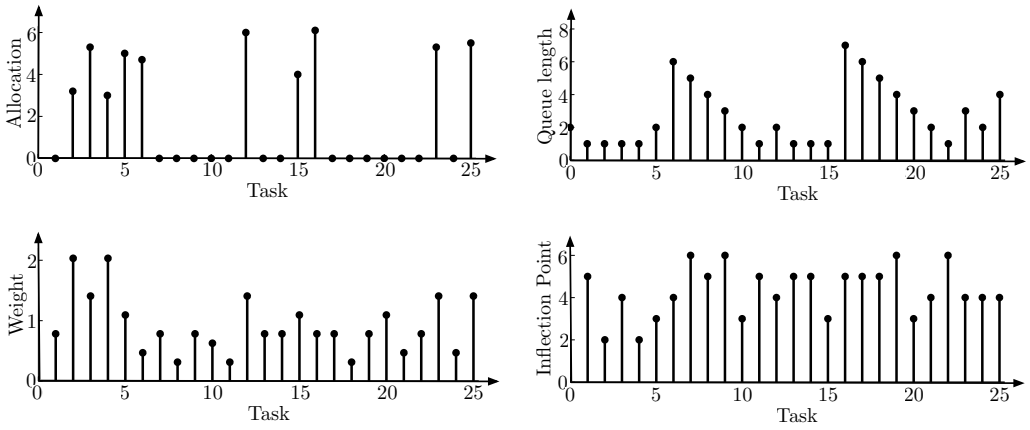


Figure 6.3: 10-RH policy for a *sample evolution* of the dynamic queue with latency penalty.

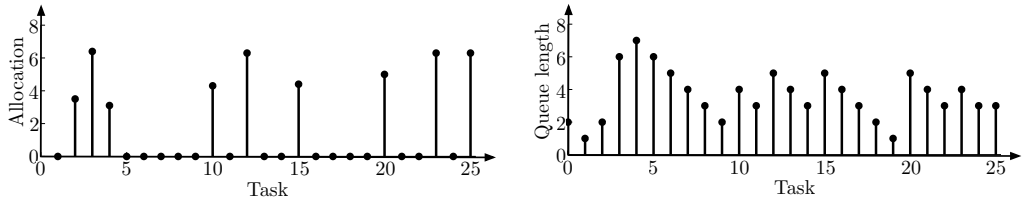


Figure 6.4: 1-RH policy for a *sample evolution* of the dynamic queue with latency penalty.

Discussion 6.1 (***Optimal arrival rate***)**.** The performance of the RH policy as a function of the arrival rate is shown in Figure 6.5. It can be seen that the expected benefit per unit task, that is, the value of the average value function under the RH policy, decreases slowly till a critical arrival rate and then starts decreasing quickly. This critical arrival rate corresponds to the situation where

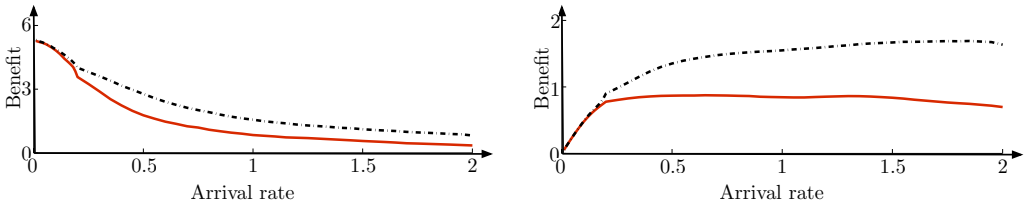


Figure 6.5: Empirical expected benefit per unit task and per unit time. The dashed-dotted black curve represents the 10-RH policy and the solid red curve represents the 1-RH policy, respectively.

a new task is expected to arrive as soon as the operator finishes processing the current task. The objective of the designer is to achieve a good performance on each task and therefore, the arrival rate should be picked close to this critical arrival rate. If each task is identical and is characterized by $d \in \mathcal{D}$, then it can be verified that the critical arrival rate is $\lambda_d^{\text{crit}} = 1/\tau_d^{\text{crit}}$, where $\tau_d^{\text{crit}} = f_d^{\dagger}(2c_d/w_d)$. In the context of heterogeneous tasks, if each task is sampled from p, then the critical arrival rate is $\sum_{d\in\mathcal{D}} p(d)\lambda_d^{\text{crit}}$. In general, designer may have other performance goals for the operator, and accordingly, may choose higher task arrival rate. □

6.4 Conclusions

We presented optimal servicing policies for the queues where the performance function of the server is a sigmoid function. First, we considered a queue with no arrival and a latency penalty. It was observed that the optimal policy may drop some tasks. Second, a dynamic queue with latency penalty was considered.

We posed the problem in an MDP framework and proposed an approximate solution in the certainty-equivalent receding horizon optimization framework. We derived performance bounds for the proposed solution and suggested guidelines for choosing the expected arrival rate for the queue.

The decision support system designed in this chapter assumes that the arrival rate of the tasks as well as the parameters in the performance function are known. An interesting open problem is to come up with policies which perform an online estimation of the arrival rate and the parameters of the performance function and simultaneously determine the optimal allocation policy. Another interesting problem is to incorporate more human factors into the optimal policy, for example, situational awareness, fatigue, etc. The policies designed in this chapter rely on first-come first-serve discipline to process tasks. It would be of interest to study problems with other processing disciples, for example, preemptive queues. We focused on open loop optimization of human performance interacting with automata. A significant future direction is to incorporate human feedback and study closed loop policies that are jointly optimal for the human operator as well as the automaton. Some preliminary results in this direction as presented in [98].

6.5 Appendix: Finite horizon optimization for identical tasks

In this section, we consider the special case of the finite horizon optimization problem (6.3) in which tasks are identical and propose a procedure to obtain the exact solution. We remark that even if the tasks are heterogeneous, many a times extensive experiments can not be done to determine operator's performance on each task. Under such circumstances, each task is treated as identical and a performance function associated with average data is used for each task. We also note that the optimal human attention allocation policy is needed to counter the information overload situations. The information overload situations correspond to the heavy traffic regime of the queue and we focus on this particular regime. In the following, we denote the sigmoid function and the latency penalty associated with each task by f and c, respectively. Let the inflection point associated with f be t^{inf}. We assume that the weight associated with each task is unity. We note that under the heavy-traffic regime the certainty-equivalent queue length is $\bar{n}_\ell = n_1 - \ell + 1 + \lambda \sum_{j=1}^{\ell-1} t_j$. Substituting the certainty-equivalent queue length into the objective function of the optimization problem (6.3), we obtain the function

$J : \mathbb{R}^N_{\geq 0} \to \mathbb{R}$ defined by

$$J(\boldsymbol{t}) := \frac{1}{N} \sum_{\ell=1}^{N} \Big(f(t_\ell) - c(n_1 - \ell + 1)t_\ell - c\lambda t_\ell \sum_{j=1, j\neq\ell}^{N} t_j - \frac{c\lambda t_\ell^2}{2} \Big),$$

where c is the penalty rate, λ is the arrival rate, and n_1 is the initial queue length. Thus, the optimization problem (6.3) is equivalent to

$$\underset{\boldsymbol{t}\succeq 0}{\text{maximize}} \quad J(\boldsymbol{t}). \tag{6.4}$$

Assume that the solution to the optimization problem (6.3) allocates a strictly positive time only to the tasks in the set $\mathcal{T}_{\text{proc}} \subseteq \{1, \ldots, N\}$, which we call the *set of processed tasks*. (Accordingly, the policy allocates zero time to the tasks in $\{1, \ldots, N\} \setminus \mathcal{T}_{\text{proc}}$). Without loss of generality, assume

$$\mathcal{T}_{\text{proc}} := \{\eta_1, \ldots, \eta_m\},$$

where $\eta_1 < \cdots < \eta_m$ and $m \leq N$. A duration allocation vector $\boldsymbol{t}$ is said to be consistent with $\mathcal{T}_{\text{proc}}$ if only the tasks in $\mathcal{T}_{\text{proc}}$ are allocated non-zero duration.

Lemma 6.5 (***Properties of maximum points***)**.** For the optimization problem (6.4), and a set of processed tasks $\mathcal{T}_{\text{proc}}$, the following statements hold:

(i). a global maximum point $\boldsymbol{t}^*$ satisfy $t^*_{\eta_1} \geq t^*_{\eta_2} \geq \ldots \geq t^*_{\eta_m}$;

(ii). a local maximum point $\boldsymbol{t}^\dagger$ consistent with $\mathcal{T}_{\text{proc}}$ satisfies

$$f'(t^\dagger_{\eta_k}) = c(n_1 - \eta_k + 1) + c\lambda \sum_{i=1}^{m} t^\dagger_{\eta_i}, \text{ for all } k \in \{1, \ldots, m\}; \tag{6.5}$$

(iii). the system of equations (6.5) can be reduced to

$$f'(t^\dagger_{\eta_1}) = \mathcal{P}(t^\dagger_{\eta_1}), \text{ and } t^\dagger_{\eta_k} = f^\dagger(f'(t^\dagger_{\eta_1}) - c(\eta_k - \eta_1)),$$

for each $k \in \{2, \dots, m\}$, where $\mathcal{P} : \mathbb{R}_{>0} \to \mathbb{R} \cup \{+\infty\}$ is defined by

$$\mathcal{P}(t) = \begin{cases} p(t), & \text{if } f'(t) \geq c(\eta_m - \eta_1), \\ +\infty, & \text{otherwise,} \end{cases}$$

where $p(t) = c(n_1 - \eta_1 + 1 + \lambda t + \lambda \sum_{k=2}^m f^\dagger(f'(t) - c(\eta_k - \eta_1)))$;

(iv). a local maximum point $\boldsymbol{t}^\dagger$ consistent with $\mathcal{T}_{\text{proc}}$ satisfies

$$f''(t_{\eta_k}) \leq c\lambda, \text{ for all } k \in \{1, \dots, m\}.$$

Proof. We start by proving the first statement. Assume $t^*_{\eta_j} < t^*_{\eta_k}$ and define the allocation vector $\bar{\boldsymbol{t}}$ consistent with $\mathcal{T}_{\text{proc}}$ by

$$\bar{t}_{\eta_i} = \begin{cases} t^*_{\eta_i}, & \text{if } i \in \{1, \dots, m\} \setminus \{j, k\}, \\ t^*_{\eta_j}, & \text{if } i = k, \\ t^*_{\eta_k}, & \text{if } i = j. \end{cases}$$

It is easy to see that

$$J(\boldsymbol{t}^*) - J(\bar{\boldsymbol{t}}) = (\eta_j - \eta_k)(t^*_{\eta_j} - t^*_{\eta_k}) < 0.$$

This inequality contradicts the assumption that $\boldsymbol{t}^*$ is a global maximum of J.

To prove the second statement, note that a local maximum is achieved at the boundary of the feasible region or at the set where the Jacobian of J is zero. At the boundary of the feasible region $\mathbb{R}^N_{\geq 0}$, some of the allocations are zero. Given the m non-zero allocations, the Jacobian of the function J projected on the space spanned by the non-zero allocations must be zero. The expressions in the theorem are obtained by setting the Jacobian to zero.

To prove the third statement, we subtract the expression in equation (6.5) for $k = j$ from the expression for $k = 1$ to get

$$f'(t_{\eta_j}) = f'(t_{\eta_1}) - c(\eta_j - \eta_1). \tag{6.6}$$

There exists a solution of equation (6.6) if and only if $f'(t_{\eta_1}) \geq c(\eta_j - \eta_1)$. If $f'(t_{\eta_1}) < c(\eta_j - \eta_1) + f'(0)$, then there exists only one solution. Otherwise, there exist two solutions. It can be seen that if there exist two solutions $t_j^{\pm}$, with $t_j^- < t_j^+$, then $t_j^- < t_{\eta_1} < t_j^+$. From the first statement, it follows that only possible allocation is t_j^+. Notice that $t_j^+ = f^\dagger(f'(t_{\eta_1}) - c(\eta_j - \eta_1))$. This choice yields feasible time allocation to each task $\eta_j, j \in \{2, \ldots, m\}$ parametrized by the time allocation to the task η_1. A typical allocation is shown in Figure 6.7(a). We further note that the effective penalty rate for the task η_1 is $c(n_1 - \eta_1 + 1) + c\lambda \sum_{j=1}^m t_{\eta_j}$. Using the expression of $t_{\eta_j}, j \in \{2, \ldots, m\}$, parametrized by t_{η_1}, we obtain the expression for $\mathcal{P}$.

To prove the last statement, we observe that the Hessian of the function J is

$$\frac{\partial^2 J}{\partial \boldsymbol{t}^2} = \text{diag}(f''(t_{\eta_1}), \ldots, f''(t_{\eta_m})) - c\lambda \mathbf{1}_m \mathbf{1}_m^T,$$

where $\text{diag}(\cdot)$ represents a diagonal matrix with the argument as diagonal entries. For a local maximum to exist at non-zero duration allocations $\{t_{\eta_1}, \ldots, t_{\eta_m}\}$, the Hessian must be negative semidefinite. A necessary condition for Hessian to be negative semidefinite is that diagonal entries are non-positive. □

We refer to the function $\mathcal{P}$ as the *effective penalty rate* for the first processed task. A typical graph of $\mathcal{P}$ is shown in Figure 6.7(b). Given $\mathcal{T}_{\text{proc}}$, a feasible allocation to the task η_1 is such that $f'(t_{\eta_1}) - c(\eta_j - \eta_1) > 0$, for each $j \in \{2, \ldots, m\}$. For a given $\mathcal{T}_{\text{proc}}$, we define the minimum feasible duration allocated to task η_1 (see Figure 6.7(a)) by

$$\tau_1 := \begin{cases} \min\{t \in \mathbb{R}_{\geq 0} \mid f'(t) = c(\eta_m - \eta_1)\}, & \text{if } f'(t^{\text{inf}}) \geq c(\eta_m - \eta_1), \\ 0, & \text{otherwise.} \end{cases}$$

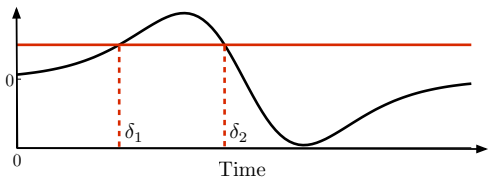
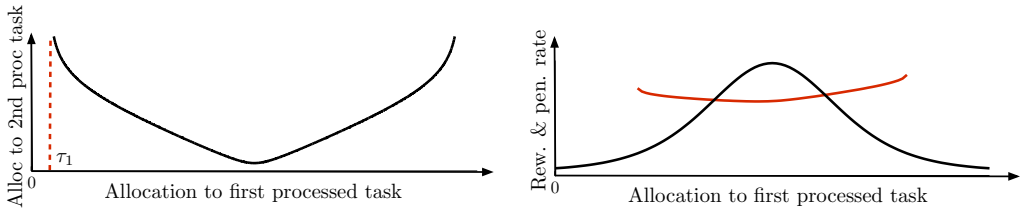


Figure 6.6: Second derivative of a sigmoid function.

Let $f''_{\max}$ be the maximum value of f''. We now define the points at which the function $f'' - c\lambda$ changes its sign (see Figure 6.6):

$$\delta_1 := \begin{cases} \min\{t \in \mathbb{R}_{\geq 0} \mid f''(t) = c\lambda\}, & \text{if } c\lambda \in [f''(0), f''_{\max}], \\ 0, & \text{otherwise,} \end{cases}$$

$$\delta_2 := \begin{cases} \max\{t \in \mathbb{R}_{\geq 0} \mid f''(t) = c\lambda\}, & \text{if } c\lambda \leq f''_{\max}, \\ 0, & \text{otherwise.} \end{cases}$$



(a) (b)

Figure 6.7: (a) Feasible allocations to the second processed task parametrized by the allocation to the first processed task. (b) The penalty rate and the sigmoid derivative as a function of the allocation to the first task.

Theorem 6.6 (***Finite horizon optimization***)**.** Given the optimization problem (6.4), and a set of processed tasks $\mathcal{T}_{\text{proc}}$. The following statements are equivalent:

(i). there exists a local maximum point consistent with $\mathcal{T}_{\text{proc}}$;

(ii). one of the following conditions hold

$$f'(\delta_2) \geq \mathcal{P}(\delta_2), \text{ or} \tag{6.7}$$

$$f'(\tau_1) \leq \mathcal{P}(\tau_1),\ f'(\delta_1) \geq \mathcal{P}(\delta_1), \text{ and } \delta_1 \geq \tau_1. \tag{6.8}$$

Proof. A critical allocation to task η_1 is located at the intersection of the graph of the reward rate $f'(t_{\eta_1})$ and the effective penalty rate $\mathcal{P}(t_{\eta_1})$. From Lemma 6.5, a necessary condition for the existence of a local maximum at a critical point is $f''(t_{\eta_1}) \leq c\lambda$, which holds for $t_{\eta_1} \in (0, \delta_1] \cup [\delta_2, \infty)$. It can be seen that if condition (6.7) holds, then the function $f'(t_{\eta_1})$ and the effective penalty function $\mathcal{P}(t_{\eta_1})$ intersect in the region $[\delta_2, \infty[$. Similarly, condition (6.8) ensures the intersection of the graph of the reward function $f'(t_{\eta_1})$ with the effective penalty function $\mathcal{P}(t_{\eta_1})$ in the region $(0, \delta_1]$. □

Bibliography

[1] S. Ağrali and J. Geunes. Solving knapsack problems with S-curve return functions. *European Journal of Operational Research*, 193(2):605–615, 2009.

[2] K. M. Adusumilli and J. J. Hasenbein. Dynamic admission and service rate control of a queue. *Queueing Systems*, 66(2):131–154, 2010.

[3] A. Arapostathis, V. S. Borkar, E. Fernández-Gaucherand, M. K. Ghosh, and S. I. Marcus. Discrete-time controlled Markov processes with average cost criterion: A survey. *SIAM Journal on Control and Optimization*, 31(2):282–344, 1993.

[4] C. Z. Bai, V. Gupta, and Y. F. Huang. Sequential hypothesis testing with off-line randomized sensor selection strategy. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pages 3269–3272, 2012.

[5] D. Bajović, B. Sinopoli, and J. Xavier. Sensor selection for hypothesis testing in wireless sensor networks: a Kullback-Leibler based approach. In *IEEE Conf. on Decision and Control*, pages 1659–1664, Shanghai, China, December 2009.

[6] M. Basseville and I. V. Nikiforov. *Detection of Abrupt Changes: Theory and Application*. Prentice Hall, 1993.

[7] C. W. Baum and V. V. Veeravalli. A sequential procedure for multihypothesis testing. *IEEE Transactions on Information Theory*, 40(6):1994–2007, 1994.

[8] H. P. Benson. On the global optimization of sums of linear fractional functions over a convex set. *Journal of Optimization Theory & Applications*, 121(1):19–39, 2004.

[9] D. Bertsekas. Dynamic programming and suboptimal control: A survey from ADP to MPC. *European Journal of Control*, 11(4-5):310–334, 2005.

[10] D. P. Bertsekas. Convergence of discretization procedures in dynamic programming. *IEEE Transactions on Automatic Control*, 20(6):415–419, 1975.

[11] L. F. Bertuccelli, N. W. M. Beckers, and M. L. Cummings. Developing operator models for UAV search scheduling. In *AIAA Conf. on Guidance, Navigation and Control*, Toronto, Canada, August 2010.

[12] L. F. Bertuccelli, N. Pellegrino, and M. L. Cummings. Choice modeling of relook tasks for UAV search missions. In *American Control Conference*, pages 2410–2415, Baltimore, MD, USA, June 2010.

[13] R. Bogacz, E. Brown, J. Moehlis, P. Holmes, and J. D. Cohen. The physics of optimal decision making: A formal analysis of performance in two-alternative forced choice tasks. *Psychological Review*, 113(4):700–765, 2006.

[14] S. Boyd, P. Diaconis, and L. Xiao. Fastest mixing Markov chain on a graph. *SIAM Review*, 46(4):667–689, 2004.

[15] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[16] K. M. Bretthauer and B. Shetty. The nonlinear knapsack problem–algorithms and applications. *European Journal of Operational Research*, 138(3):459–472, 2002.

[17] W. M. Bulkeley. Chicago's camera network is everywhere. The Wall Street Journal, November 17, 2009.

[18] G. J. Burke, J. Geunes, H. E. Romeijn, and A. Vakharia. Allocating procurement to capacitated suppliers with concave quantity discounts. *Operations Research Letters*, 36(1):103–109, 2008.

[19] G. C. Calafiore, F. Dabbene, and R. Tempo. Research on probabilistic methods for control system design. *Automatica*, 47(7):1279–1293, 2011.

[20] J. Calvin. Convergence rate of the P-algorithm for optimization of continuous functions. In P. M. Pardalos, editor, *Approximation and Complexity in Numerical Optimization: Continuous and Discrete Problems*. Kluwer Academic, 1999.

[21] J. Calvin and A. Žilinskas. On the convergence of the P-algorithm for one-dimensional global optimization of smooth functions. *Journal of Optimization Theory and Applications*, 102(3):479–495, 1999.

[22] D. A. Castañón. Optimal search strategies in dynamic hypothesis testing. *IEEE Transactions on Systems, Man & Cybernetics*, 25(7):1130–1138, 1995.

[23] H. S. Chang and S. I. Marcus. Approximate receding horizon approach for Markov decision processes: Average reward case. *Journal of Mathematical Analysis and Applications*, 286(2):636–651, 2003.

[24] B. Chen and P. Willett. Detection of hidden Markov model transient signals. *IEEE Transactions on Aerospace and Electronic Systems*, 36(4):1253–1268, 2000.

[25] Y. Chevaleyre. Theoretical analysis of the multi-agent patrolling problem. In *IEEE/WIC/ACM Int. Conf. on Intelligent Agent Technology*, pages 302–308, Beijing, China, September 2004.

[26] T. H. Chung and J. W. Burdick. Analysis of search decision making using probabilistic search strategies. *IEEE Transactions on Robotics*, 28(1):132–144, 2012.

[27] R. Cohen, L. Katzir, and D. Raz. An efficient approximation for the generalized assignment problem. *Information Processing Letters*, 100(4):162–166, 2006.

[28] T. M. Cover and J. A. Thomas. *Elements of Information Theory.* Wiley, 1991.

[29] J. W. Crandall, M. L. Cummings, M. Della Penna, and P. M. A. de Jong. Computing the effects of operator attention allocation in human control of multiple robots. *IEEE Transactions on Systems, Man & Cybernetics. Part A: Systems & Humans*, 41(3):385–397, 2011.

[30] R. Debouk, S. Lafortune, and D. Teneketzis. On an optimization problem in sensor selection. *Discrete Event Dynamic Systems*, 12(4):417–445, 2002.

[31] C. Drew. Military taps social networking skills. The New York Times, June 7, 2010.

[32] J. W. Durham and F. Bullo. Smooth nearness-diagram navigation. In *IEEE/RSJ Int. Conf. on Intelligent Robots & Systems*, pages 690–695, Nice, France, September 2008.

[33] Y. Elmaliach, A. Shiloni, and G. A. Kaminka. A realistic model of frequency-based multi-robot polyline patrolling. In *International Conference on Autonomous Agents*, pages 63–70, Estoril, Portugal, May 2008.

[34] J. R. Freeland and C. B. Weinberg. S-Shaped response functions: Implications for decision models. *Journal of the Operational Research Society*, 31(11):1001–1007, 1980.

[35] J. M. George and J. M. Harrison. Dynamic control of a queue with adjustable service rate. *Operations Research*, 49(5):720–731, 2001.

[36] W. Ginsberg. The multiplant firm with increasing returns to scale. *Journal of Economic Theory*, 9(3):283–292, 1974.

[37] J. Grace and J. Baillieul. Stochastic strategies for autonomous robotic surveillance. In *IEEE Conf. on Decision and Control and European Control Conference*, pages 2200–2205, Seville, Spain, December 2005.

[38] E. Guizzo. Obama commanding robot revolution announces major robotics initiative. IEEE Spectrum, June 2011.

[39] V. Gupta, T. H. Chung, B. Hassibi, and R. M. Murray. On a stochastic sensor selection algorithm with applications in sensor scheduling and sensor coverage. *Automatica*, 42(2):251–260, 2006.

[40] A. Gut. *Stopped Random Walks: Limit Theorems and Applications*. Springer, 2009.

[41] P. Hansen, B. Jaumard, and S. H. Lu. Global optimization of univariate Lipschitz functions: I. Survey and properties. *Mathematical Programming*, 55(1):251–272, 1992.

[42] O. Hernández-Lerma and S. I. Marcus. Adaptive control of service in queueing systems. *Systems & Control Letters*, 3(5):283–289, 1983.

[43] J. P. Hespanha, H. J. Kim, and S. S. Sastry. Multiple-agent probabilistic pursuit-evasion games. In *IEEE Conf. on Decision and Control*, pages 2432–2437, Phoenix, AZ, USA, December 1999.

[44] J. B. Hiriart-Urruty. On optimality conditions in nondifferentiable programming. *Mathematical Programming*, 14(1):73–86, 1978.

[45] G. A. Hollinger, U. Mitra, and G. S. Sukhatme. Active classification: Theory and application to underwater inspection. In *International Symposium on Robotics Research*, Flagstaff, AZ, USA, August 2011.

[46] G. A. Hollinger, U. Mitra, and G. S. Sukhatme. Autonomous data collection from underwater sensor networks using acoustic communication. In *IEEE/RSJ Int. Conf. on Intelligent Robots & Systems*, pages 3564–3570, San Francisco, CA, USA, September 2011.

[47] S. K. Hong and C. G. Drury. Sensitivity and validity of visual search models for multiple targets. *Theoretical Issues in Ergonomics Science*, 3(1):85–110, 2002.

[48] T. Ibaraki and N. Katoh. *Resource Allocation Problems: Algorithmic Approaches*. MIT Press, 1988.

[49] V. Isler and R. Bajcsy. The sensor selection problem for bounded uncertainty sensing models. *IEEE Transactions on Automation Sciences and Engineering*, 3(4):372–381, 2006.

[50] S. Joshi and S. Boyd. Sensor selection via convex optimization. *IEEE Transactions on Signal Processing*, 57(2):451–462, 2009.

[51] S. Stidham Jr. and R. R. Weber. Monotonic and insensitive optimal policies for control of queues with undiscounted costs. *Operations Research*, 37(4):611–625, 1989.

[52] S. Kameshwaran and Y. Narahari. Nonconvex piecewise linear knapsack problems. *European Journal of Operational Research*, 192(1):56–68, 2009.

[53] V. Katewa and V. Gupta. On the optimality of sequential test with multiple sensors. In *American Control Conference*, pages 282–287, Montréal, Canada, 2012.

[54] H. Kellerer, U. Pferschy, and D. Pisinger. *Knapsack Problems*. Springer, 2004.

[55] D. B. Kingston, R. W. Beard, and R. S. Holt. Decentralized perimeter surveillance using a team of UAVs. *IEEE Transactions on Robotics*, 24(6):1394–1404, 2008.

[56] D. J. Klein, J. Schweikl, J. T. Isaacs, and J. P. Hespanha. On UAV routing protocols for sparse sensor data exfiltration. In *American Control Conference*, pages 6494–6500, Baltimore, MD, USA, June 2010.

[57] B. Korte and J. Vygen. *Combinatorial Optimization: Theory and Algorithms*, volume 21 of *Algorithmics and Combinatorics*. Springer, 4 edition, 2007.

[58] H. J. Kushner. A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise. *Journal of Basic Engineering*, 86(1):97–106, 1964.

[59] S. M. LaValle. *Planning Algorithms*. Cambridge University Press, 2006. Available at `http://planning.cs.uiuc.edu`.

[60] D. G. Luenberger. *Linear and Nonlinear Programming*. Addison-Wesley, 2 edition, 1984.

[61] S. Martello and P. Toth. *Knapsack Problems: Algorithms and Computer Implementations*. Wiley, 1990.

[62] J. Mattingley, Y. Wang, and S. Boyd. Receding horizon control: Automatic generation of high-speed solvers. *IEEE Control Systems Magazine*, 31(3):52–65, 2011.

[63] J. J. Moré and S. A. Vavasis. On the solution of concave knapsack problems. *Mathematical Programming*, 49(1):397–411, 1990.

[64] G. V. Moustakides. Optimal stopping times for detecting changes in distributions. *The Annals of Statistics*, 14(4):1379–1387, 1986.

[65] C. Nehme, B. Mekdeci, J. W. Crandall, and M. L. Cummings. The impact of heterogeneity on operator performance in futuristic unmanned vehicle systems. *The International C2 Journal*, 2(2):1–30, 2008.

[66] C. E. Nehme. *Modeling Human Supervisory Control in Heterogeneous Unmanned Vehicle Systems*. PhD thesis, Department of Aeronautics and Astronautics, MIT, February 2009.

[67] A. Ortiz, D. Kingston, and C. Langbort. Multi-UAV Velocity and Trajectory Scheduling Strategies for Target Classification by a Single Human Operator. *Journal of Intelligent & Robotic Systems*, pages 1–20, 2012.

[68] E. S. Page. Continuous inspection schemes. *Biometrika*, 41(1/2):100–115, 1954.

[69] F. Pasqualetti, J. W. Durham, and F. Bullo. Cooperative patrolling via weighted tours: Performance analysis and distributed algorithms. *IEEE Transactions on Robotics*, 28(5):1181–1188, 2012.

[70] F. Pasqualetti, A. Franchi, and F. Bullo. On cooperative patrolling: Optimal trajectories, complexity analysis and approximation algorithms. *IEEE Transactions on Robotics*, 28(3):592–606, 2012.

[71] R. W. Pew. The speed-accuracy operating characteristic. *Acta Psychologica*, 30:16–26, 1969.

[72] H. V. Poor and O. Hadjiliadis. *Quickest Detection*. Cambridge University Press, 2008.

[73] N. D. Powel and K. A. Morgansen. Multiserver queueing for supervisory control of autonomous vehicles. In *American Control Conference*, pages 3179–3185, Montréal, Canada, June 2012.

[74] R. J. Radke, S. Andra, O. Al-Kofahi, and B. Roysam. Image change detection algorithms: A systematic survey. *IEEE Transactions on Image Processing*, 14(3):294–307, 2005.

[75] A. G. Rao and M. R. Rao. Optimal budget allocation when response is S-shaped. *Operations Research Letters*, 2(5):225–230, 1983.

[76] S. I. Resnick. *A Probability Path*. Birkhäuser, 1999.

[77] T. Sak, J. Wainer, and S. Goldenstein. Probabilistic multiagent patrolling. In *Brazilian Symposium on Artificial Intelligence*, Advances in Artificial Intelligence, pages 124–133, Salvador, Brazil, 2008. Springer.

[78] K. Savla and E. Frazzoli. Maximally stabilizing task release control policy for a dynamical queue. *IEEE Transactions on Automatic Control*, 55(11):2655–2660, 2010.

[79] K. Savla and E. Frazzoli. A dynamical queue approach to intelligent task management for human operators. *Proceedings of the IEEE*, 100(3):672–686, 2012.

[80] K. Savla, C. Nehme, T. Temple, and E. Frazzoli. On efficient cooperative strategies between UAVs and humans in a dynamic environment. In *AIAA Conf. on Guidance, Navigation and Control*, Honolulu, HI, USA, 2008.

[81] K. Savla, T. Temple, and E. Frazzoli. Human-in-the-loop vehicle routing policies for dynamic environments. In *IEEE Conf. on Decision and Control*, pages 1145–1150, Cancún, México, December 2008.

[82] D. K. Schmidt. A queuing analysis of the air traffic controller's work load. *IEEE Transactions on Systems, Man & Cybernetics*, 8(6):492–498, 1978.

[83] L. I. Sennott. *Stochastic Dynamic Programming and the Control of Queueing Systems*. Wiley, 1999.

[84] T. Shanker and M. Richtel. In new military, data overload can be deadly. The New York Times, January 16, 2011.

[85] D. Siegmund. *Sequential Analysis: Tests and Confidence Intervals*. Springer, 1985.

[86] S. L. Smith and D. Rus. Multi-robot monitoring in dynamic environments with guaranteed currency of observations. In *IEEE Conf. on Decision and Control*, pages 514–521, Atlanta, GA, USA, December 2010.

[87] S. L. Smith, M. Schwager, and D. Rus. Persistent robotic tasks: Monitoring and sweeping in changing environments. *IEEE Transactions on Robotics*, 28(2):410–426, 2012.

[88] D. N. Southern. Human-guided management of collaborating unmanned vehicles in degraded communication environments. Master's thesis, Electrical Engineering and Computer Science, Massachusetts Institute of Technology, May 2010.

[89] K. Srivastava, D. M. Stipanovìc, and M. W. Spong. On a stochastic robotic surveillance problem. In *IEEE Conf. on Decision and Control*, pages 8567–8574, Shanghai, China, December 2009.

[90] V. Srivastava and F. Bullo. Hybrid combinatorial optimization: Sample problems and algorithms. In *IEEE Conf. on Decision and Control and European Control Conference*, pages 7212–7217, Orlando, FL, USA, December 2011.

[91] V. Srivastava and F. Bullo. Stochastic surveillance strategies for spatial quickest detection. In *IEEE Conf. on Decision and Control and European Control Conference*, pages 83–88, Orlando, FL, USA, December 2011.

[92] V. Srivastava and F. Bullo. Knapsack problems with sigmoid utility: Approximation algorithms via hybrid optimization. *European Journal of Operational Research*, October 2012. Submitted.

[93] V. Srivastava, R. Carli, F. Bullo, and C. Langbort. Task release control for decision making queues. In *American Control Conference*, pages 1855–1860, San Francisco, CA, USA, June 2011.

[94] V. Srivastava, R. Carli, C. Langbort, and F. Bullo. Attention allocation for decision making queues. *Automatica*, February 2012. Submitted.

[95] V. Srivastava, F. Pasqualetti, and F. Bullo. Stochastic surveillance strategies for spatial quickest detection. *International Journal of Robotics Research*, April 2012. to appear.

[96] V. Srivastava, K. Plarre, and F. Bullo. Adaptive sensor selection in sequential hypothesis testing. In *IEEE Conf. on Decision and Control and European Control Conference*, pages 6284–6289, Orlando, FL, USA, December 2011.

[97] V. Srivastava, K. Plarre, and F. Bullo. Randomized sensor selection in sequential hypothesis testing. *IEEE Transactions on Signal Processing*, 59(5):2342–2354, 2011.

[98] V. Srivastava, A. Surana, and F. Bullo. Adaptive attention allocation in human-robot systems. In *American Control Conference*, pages 2767–2774, Montréal, Canada, June 2012.

[99] W. P. Tay, J. N. Tsitsiklis, and M. Z. Win. Asymptotic performance of a censoring sensor network. *IEEE Transactions on Information Theory*, 53(11):4191–4209, 2007.

[100] S. Thrun, D. Fox, W. Burgard, and F. Dellaert. Robust Monte Carlo localization for mobile robots. *Artificial Intelligence*, 128(1-2):99–141, 2001.

[101] J. Unnikrishnan, V. V. Veeravalli, and S. P. Meyn. Minimax robust quickest change detection. *IEEE Transactions on Information Theory*, 57(3):1604–1614, 2011.

[102] D. Vakratsas, F. M. Feinberg, F. M. Bass, and G. Kalyanaram. The shape of advertising response functions revisited: A model of dynamic probabilistic thresholds. *Marketing Science*, 23(1):109–119, 2004.

[103] A. Wald. Sequential tests of statistical hypotheses. *The Annals of Mathematical Statistics*, 16(2):117–186, 1945.

[104] H. Wang, K. Yao, G. Pottie, and D. Estrin. Entropy-based sensor selection heuristic for target localization. In *Symposium on Information Processing of Sensor Networks*, pages 36–45, Berkeley, CA, April 2004.

[105] L. Wasserman. *All of Statistics: A Concise Course in Statistical Inference*. Springer, 2004.

[106] C. D. Wickens and J. G. Hollands. *Engineering Psychology and Human Performance*. Prentice Hall, 3 edition, 2000.

[107] J. L. Williams, J. W. Fisher, and A. S. Willsky. Approximate dynamic programming for communication-constrained sensor network management. *IEEE Transactions on Signal Processing*, 55(8):4300–4311, 2007.

[108] D. Zhang, C. Colburn, and T. Bewley. Estimation and adaptive observation of environmental plumes. In *American Control Conference*, pages 4821–4286, San Francisco, CA, USA, June 2011.