Perfect Simulation and Deployment Strategies for Detection

Aya Wallwater

Submitted in partial fulfillment of the
requirements for the degree
of Doctor of Philosophy
in the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

2015

©2015
Aya Wallwater
All Rights Reserved

ABSTRACT

Perfect Simulation and Deployment Strategies for Detection

Aya Wallwater

This dissertation contains two parts. The first part provides the first algorithm that, under minimal assumptions, allows to simulate the stationary waiting-time sequence of a single-server queue backwards in time, jointly with the input processes of the queue (inter-arrival and service times). The single-server queue is useful in applications of DCFTP (Dominated Coupling From The Past), which is a well known protocol for simulation without bias from steady-state distributions. Our algorithm terminates in finite time assuming only finite mean of the inter-arrival and service times. In order to simulate the single-server queue in stationarity until the first idle period in finite expected termination time we require the existence of finite variance. This requirement is also necessary for such idle time (which is a natural coalescence time in DCFTP applications) to have finite mean. Thus, in this sense, our algorithm is applicable under minimal assumptions.

The second part studies the behavior of diffusion processes in a random environment. We consider an adversary that moves in a given domain and our goal is to produce an optimal strategy to detect and neutralize him by a given deadline. We assume that the target's dynamics follows a diffusion process whose parameters are informed by available intelligence information. We will dedicate one chapter to the rigorous formulation of the detection problem, an introduction of several frameworks that can be considered when applying our methods, and a discussion on the challenges of finding the analytical optimal solution. Then, in the following chapter, we will present our main result, a large deviation behavior of the adversary's survival probability under a given strategy. This result will be later give rise to asymptotically efficient Monte Carlo algorithms.

Table of Contents

List of Figures **iv**

List of Tables **v**

List of Algorithms **vi**

1 Introduction **1**

I Exact Sampling of Stationary and Time-Reversed Queues **3**

2 Perfect Sampling **7**

2.1 Introduction 7

2.2 Construction of $(S_n(\mu), M_n : n \geq 0)$ via "milestone events" 8

2.3 Sampling M_0 jointly with $(S_1(\mu), ..., S_\Delta(\mu))$ 14

2.3.1 Sampling $Ber(P_0(T_m < \infty))$ and $(S_1(\mu), ..., S_{T_m}(\mu))$ given $T_m < \infty$ 14

2.3.2 Building M_0 and $(S_1(\mu), ..., S_\Delta(\mu))$ from downward and upward patches 25

2.4 From M_0 to $(S_k(\mu), M_k : k \geq 0)$: Implementation of Procedure 1 26

2.4.1 Implementing Step 1 in Procedure 1 26

2.4.2 Implementing Step 2 in Procedure 1 27

2.4.3 Our algorithm to sample $(S_k(\mu), M_k : 0 \leq k \leq n)$ and Proof of Theorem 2 28

2.5 Additional considerations: increments with infinite variance and computing truncated tilted distributions . . . 30
2.5.1 The case where $E\left|X\right|^{\beta}<\infty$ for $\beta\in(1,2]$ . . . 30
2.5.2 The issue of evaluating $\psi_{k}\left(\theta_{k}\right)$ . . . 31
2.6 Numerical Example . . . 33
2.6.1 Choice of Parameters . . . 34
2.6.2 Simulation Results . . . 35
2.7 Conclusions . . . 37

II Diffusion Processes in Random Environment 38

3 Detection of Adversarial Agents: Modeling and Problem Formulation 44
3.1 One-Dimensional Brownian Case . . . 45
3.1.1 Deterministic Case . . . 45
3.1.2 The Stochastic Approach . . . 47
3.2 Disscusion . . . 54
3.3 The General Scenario . . . 55
3.3.1 Player 1 . . . 56
3.3.2 Player 2 . . . 57
3.4 Optimal Policy: Two-Dimensional Brownian Motion . . . 59
3.4.1 Placing K Traps in G . . . 60
3.4.2 Partitioning G to K Subsets . . . 62

4 Detection of Adversarial Agents: Main Results and Algorithms 64
4.1 Main Result . . . 65
4.2 Upper bound . . . 66
4.2.1 Initialization . . . 67
4.2.2 Step k, $k\geq 1$ . . . 67
4.2.3 Proof of upper bound . . . 68
4.3 Lower bound . . . 70
4.4 Asymptotically Optimal Monte Carlo for Detection Policies . . . 71

4.4.1 Estimating $P\left(\sigma_R > T\right)$ 72
4.4.2 Sampling form P_* the law of $\left\{X\left(T\right) \mid \sigma_{\mathcal{R}} > T\right\}$ 74
4.4.3 Sampling $\left\{X\left(T\right) \mid \sigma_{\mathcal{R}} > T\right\}$ when the rate function I is computable . 76
4.5 Numerical Results 78
4.5.1 One-dimensional Brownian Motion 78
4.5.2 Two-dimensional diffusion process 79
4.6 Conclusions 83

III Bibliography **84**

Bibliography **85**

Bibliography **85**

IV Appendices **89**

A Appendix for Chapter 2 **90**
A.1 Discussion on the generality of the assumptions imposed and selection of parameters 90
A.2 Proof of Lemma 3 92
A.3 Proof of Lemma 4 92
A.4 Proof of Lemma 5 96

B Appendix for Chapter 4 **97**
B.1 Proof of Lemma 15 97
B.1.1 Proof of Lemma 15 98
B.1.2 Proof of Lemma 21 99

List of Figures

2.2.1 Construction of Milestone Stopping Times 9

2.2.2 High-level description of the algorithm 12

2.6.1 Exact sampler mean $E\left(M_0\right)$ VS. batches mean of $\{W_n : 0 \leq n \leq l\}$, along with the corresponding 95% confidence intervals. 37

2.7.1 Deterministic deployment strategy 41

3.1.1 Optimal Policy for one-dimensional Brownian motion confined to $(-m, m)$. 46

3.1.2 G_ϵ 50

3.1.3 G_ϵ 51

3.3.1 Poisson Cloud Vs. Wiener Sausage 59

3.4.1 Minimal traps allocation in a ball 61

3.4.2 Tessellation of trap allocation in a ball 61

3.4.3 Level contours to determine fence location 63

4.2.1 Construction of the set G_k 66

4.3.1 G^ϵ and G_a^ϵ 70

4.5.1 A_0 (on the left) and A_1 (on the right) for $L = 2$. 81

4.5.2 -Log of the estimator's standard deviation. 82

List of Tables

2.1 Parameters Choice for Simulation 36
2.2 Simulation Results Perfect Sampling 36

4.1 Survival probability: Importance Sampling (IS) vs. naïve Simulation: One-dimensional Brownian motion. 79
4.2 Survival probability: Importance Sampling (IS) vs. naïve Simulation: Two-dimensional Brownian motion. 80

List of Algorithms

- Procedure 1 13
1 Sampling $Ber\left(P_0\left(T_m < \infty\right)\right)$ and $\left(S_1\left(\mu\right), ..., S_{T_m}\left(\mu\right)\right)$ given $T_m < \infty$ 20
2 Sampling M_0 and $\left(S_1\left(\mu\right), ..., S_\Delta\left(\mu\right)\right)$ 26
3 An Efficient Implementation of Procedure 1 29
4 Strategy for simulating $\left(S_k\left(\mu\right), M_k : 0 \leq k \leq n\right)$ 33

5 Generating a Poisson Cloud 58

6 Sampling $P\left(\sigma_R > T\right)$ 73
7 Acceptance-Rejection Algorithm 76

Acknowledgments

I would like to thank my advisor, Professor Jose Blanchet , for his unwavering support. I would like to thank the members of my committee for offering their invaluable insight and expertise. I would like to thank my fellow doctoral students - those who have moved on, those still deep in research, and those just beginning- for their support, friendship, and all around good humor. I would like to thank all the staff in the IEOR department, for making sure I progressed smoothly in my studies. I would like to thank all of my professors, for sharing their knowledge and love of learning with me. I would like to thank all my friends, for their support and good cheer. Last, but certainly not least, I would like to thank my family for their unconditional love and endless support - it is on all your shoulders that I now stand.

To my beloved husband, Aaron Miller,
Sabrina (Yuanuan) Wang, and Valia Skiani
for their unconditional love.

Chapter 1

Introduction

The work that is covered in this dissertation is driven by the intention of overcoming some of the current challenges in operations research and enhancing the bridge between mathematical methods and real-world applications. My desire to learn a wide range of tools and techniques in applied probability and operations research, such as steady-state analysis, perfect simulation, large deviation theory, spectral theory of operators, game theory and others, resulted in a dissertation that is divided into two parts.

Part I studies the topic of steady-state analysis and perfect simulation. We address the problem of producing the first general perfect (also known as exact) sampling algorithm for what is the most fundamental queuing system in operations research, the single-server queue. We will discuss the relevant background and the literature on perfect sampling in Chapter 2. We will also describe how our methods connect to other areas in Monte Carlo, such as rare-event simulation, via the use of importance sampling. Thus, this contribution is not only relevant from the perfect sampling perspective, but also because it bridges areas in simulation that are seldom connected.

The work that will be presented in Part II is motivated by our wish to solve a current problem with defense applications. In this problem, we consider an adversary that moves in a given domain and our goal is to produce an optimal strategy to detect and neutralize him by a given deadline. For the moment, we can think of a strategy as a placement of detection units in the region. We assume that the target's dynamics follows a diffusion process whose parameters are informed by available intelligence information, of which a simple example for such dynamic is a Brownian motion.

It becomes evident that as the deadline gets large and the agent has not been caught, we can show that its evasion strategy can be represented as choosing an optimal "confinement" set, which maximizes his survival probability. As we shall see, the problem is extremely challenging to solve. However, we will give a characterization to the optimal solution in terms of a set-valued function, which depends on the spectrum of the operator generating the target's dynamics.

In Chapter 3 we will formulate the problem rigorously, present several frameworks that can be considered when applying our methods, and discuss the challenges of finding the analytical optimal solution. Furthermore, in Chapter 4, we will present our main result, a large deviation behavior of the adversary's survival probability under a given strategy. This result will later give rise to efficient Monte Carlo algorithms that can numerically evaluate the performance of a given deployment policy and sample the target's location conditioned on him avoiding detection in an asymptotically optimal manner (as the deadline increases).

Part I

Exact Sampling of Stationary and Time-Reversed Queues

Introduction to Part I

The theory of exact simulation has attracted substantial attention, particularly since the ground breaking paper [Propp and Wilson, 1996]. In their paper, the authors introduced the most popular sampling protocol for exact simulation to date; namely, Coupling From The Past (CFTP). CFTP is a simulation technique which results in samples from the steady-state distribution of a Markov chain under certain compactness assumptions. The paper [Kendall, 1998] describes a useful variation of CFTP, called Dominated CFTP (DCFTP). Like CFTP, DCFTP aims to sample from the steady-state distribution of a Markov chain, but this technique can also be applied to cases in which the state-space is unbounded.

The idea in the DCFTP method is to simulate a dominating stationary process backwards in time until the detection of a so-called coalescence time, in which the target and dominating processes coincide. The sample path of the target process can then be reconstructed forward in time from coalescence up to time zero. The state of the target process at time zero is a sample from the associated stationary distribution.

Our contribution in this paper is to provide, under nearly minimal assumptions (finite-mean service and inter-arrival times), an exact simulation algorithm for the stationary workload of a single-server queue backwards in time. This is a fundamental queueing system which can be used in many applications as a natural dominating process when applying DCFTP. Usually additional assumptions, beyond the ones we consider here, have been imposed to enable the simulation of the stationary single-server queue backwards in time.

For example, in [Sigman, 2011] the problem of sampling from the distribution of an

$M/G/c$ queue is considered. It is assumed that the ratio between the arrival rate, λ, and the service rate, μ, namely the traffic intensity parameter $\rho = \lambda/\mu$ is strictly less than unity. This is a strong assumption because stability is known to hold if $\rho < c$. Nevertheless, this assumption is imposed because one can clearly use a stable $M/G/1$ queue (only one server) in order to dominate the workload. The challenge is then to detect a coalescence time, that is, a time at which the state of the target system (in this case the $M/G/c$ system) is known. A natural coalescence time in this case occurs when the upper bound process, namely the workload of the $M/G/1$ queue, simulated in stationarity and backwards in time is empty. Then, from this time the $M/G/c$ queue can be reconstructed forward up to time zero using the traffic underlying the simulation of the upper bound. At this point, the difficulty consists in precisely simulating the workload of a stationary $M/G/1$ queue backwards in time. The author in [Sigman, 2011] overcomes this problem by noting that a processor sharing queue, which can easily be simulated backwards in time because it is quasi-reversible, shares the same workload as the corresponding $M/G/1$ queue, and thus it is possible to detect coalescence. An immediate extension of our contribution here is the ability to handle $GI/G/c$ queues with $\rho < 1$.

Later [Sigman, 2012] was able to simulate a stationary $M/G/c$ queue assuming $\rho < c$. The algorithm [Sigman, 2012] avoids the use of DCFTP, but it as a price an infinite expected termination time. The strategy in [Sigman, 2012] was first to show that one can dominate the number in system of an $M/G/c$ queue using c independent $M/G/1$ queues. The coupling explained in [Sigman, 2012] consists in taking the arriving customers into the $M/G/c$ queue and using the Poisson thinning theorem to split these customers into c i.i.d. Poisson processes. The service times, however, must preserve the order in which they start to be processed in order to ensure domination. Our contribution here would allow to use the domination result in [Sigman, 2012] to produce an exact sampling algorithm for the $M/G/c$ queue which runs in finite expected termination time. In particular, we now can simulate each of the independent $M/G/1$ queues in stationarity and backwards in time. Then, we note that a coalescence time occurs when all of the queues are empty. At that point we simulate the $M/G/c$ queue forward in time up until time zero, but one must be careful to make sure that the service times are used in the $M/G/c$ according to the times in which they start to be processed in the $M/G/1$ queues. An additional extension to appear in [?] uses a different dominating process, but also exploits our work here to deal with $GI/G/c$ queues.

We note other applications. For example, the paper [Blanchet and Dong, 2012] uses the single server queue backwards in time to sample the state descriptor of the infinite server queue in stationarity – in turn, the infinite server queue is used to simulate loss networks in stationarity. In the paper [Blanchet and Sigman, 2011] the single-server queue backwards in time is used to simulate from a general class of heavy-tailed perpetuities. Both in [Blanchet and Dong, 2012] and [Blanchet and Sigman, 2011] the underlying distributions are assumed to have a finite moment generating function in a neighborhood of the origin. Applications to multidimensional stochastic-fluid networks are discussed in [Blanchet and Chen, 2012]. Our contribution here directly extends the applicability of all of these instances in which the single-server queue has been used as a dominated process under stronger assumptions. A short section at the end of this paper on a direct application to multi-server queues has been added in response to a request from one of the referees.

The first idle period (backwards in time starting from stationarity) is a natural coalescence time when applying DCFTP. Therefore, we are specially interested in an algorithm that has finite expected termination time to simulate such first idle period. Moreover, it is well known that finite-variance service times are necessary if the first idle period (starting from stationarity) has finite expected time (this follows from Wald's identity, [Durrett, 2005] p. 178, and from Theorem 2.1 in [Asmussen, 2003], p. 270). While our algorithm terminates with probability one imposing only the existence of finite mean of service times and inter-arrival times, when we assume finite variances we obtain an algorithm that has finite expected running time (see Theorem 2 in Section 2.4).

Chapter 2

Perfect Sampling

2.1 Introduction

Let us now provide the mathematical description of the problem we want to solve. Consider a random walk $S_n = X_1 + \ldots + X_n$ for $n \geq 1$, and $S_0 = 0$. We assume that $(X_k : k \geq 1)$ is a sequence of independent and identically distributed (IID) random variables with

$$EX_k = 0 \quad \text{and} \quad E\left|X_k\right|^{\beta} < \infty \quad \text{for some } \beta > 1. \tag{2.1.1}$$

As we indicated earlier, of special interest is the case $E\left|X_k\right|^{\beta} < \infty$ for some $\beta > 2$. Now, for $\mu > 0$ and $n \geq 0$ we define the negative-drift random walk and its associated running (forward) maximum by

$$S_n\left(\mu\right) = S_n - n\mu \ \text{ and } \ M_n = \max_{m \geq n}\{S_m\left(\mu\right) - S_n\left(\mu\right)\}, \tag{2.1.2}$$

respectively. Note that the maximum is taken over an infinite time-horizon, so the process $(M_n : n \geq 0)$ is not adapted to the random walk $(S_n\left(\mu\right) : n \geq 0)$. Our aim in this paper is to design an algorithm that samples jointly from the sequence $(S_n\left(\mu\right), M_n : 0 \leq n \leq N)$ for any finite N (potentially a stopping time adapted to $(S_n\left(\mu\right), M_n : n \geq 0)$). Of particular interest is the first idle time, $N = \min\{n \geq 0 : M_n = 0\}$, which can often be used as a coalescence time.

Note that if we define $W_m = M_{-m}$ for $m \leq 0$, then we can easily verify the so-called Lindlay's recursion (see [Asmussen, 2003], p. 92) namely

$$M_{-m} = (M_{-m+1} + X_{-m} - \mu)^{+} = (W_{m-1} + X_{-m} - \mu)^{+} = W_m,$$

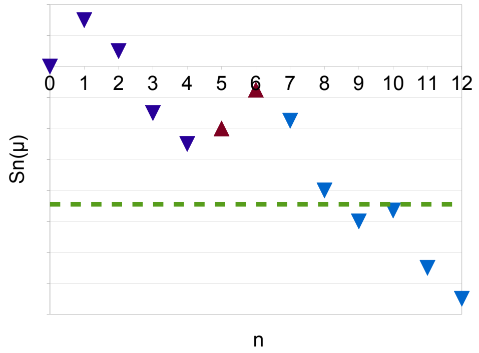
and therefore $(W_m : m \leq 0)$ corresponds to a single-server queue waiting time sequence backwards in time; the sequence is clearly stationary since the M_n's are all equal in distribution. Simulating $(S_n(\mu), M_n : n \geq 0)$ jointly allows to couple the single-server queue backwards in time with the driving sequence (i.e. the X_n's). Such coupling is required in the applications of the DCFTP method.

The algorithm that we propose here extends previous work in [Ensor and Glynn, 2000], which shows how to simulate M_0 assuming the existence of the so-called Cramer root (i.e. $\theta > 0$ such that $E(\exp(\theta X_1)) = 1$). The paper [Blanchet and Sigman, 2011] explains how to simulate $(S_n(\mu), M_n : n \geq 0)$ assuming a finite moment generating function in a neighborhood of the origin. Multidimensional extensions, also under the assumption of a finite moment generating function around the origin, are discussed in [Blanchet and Chen, 2012].

Our strategy for simulating the sequence $(S_n(\mu), M_n : n \geq 0)$ relies on certain "upward events" and "downward events" that occur at random times. These "milestone events" will be discussed in Section 2.2. In Section 2.2 we will also present the high-level description of our proposed algorithm, which will be elaborated in subsequent sections. Section 2.3 explains how to simulate M_0 under the assumption that $E|X_k|^{\beta} < \infty$ for $\beta > 2$. In Section 2.4 we built on our construction for the sampling of M_0 to simulate the sequence $(S_k(\mu), M_k : k \leq n)$. Section 2.5 will explain how to extend our algorithm to the case $E|X_k|^{\beta} < \infty$ for $\beta > 1$ and also discuss additional considerations involved in evaluating certain normalizing constants. Finally, in Section 2.6 we will present a numerical example that tests the empirical performance of our proposed algorithm.

2.2 Construction of $(S_n(\mu), M_n : n \geq 0)$ via "milestone events"

We will describe the construction of a pair of sequences of stopping times (with respect to the filtration generated by $(S_n(\mu) : n \geq 0)$), denoted by $(D_n : n \geq 0)$ and $(U_n : n \geq 1)$, which track certain downward and upward milestones in the evolution of $(S_n(\mu) : n \geq 0)$. We follow similar steps as described in [Blanchet and Sigman, 2011]. These "milestone events" will be used in the design of our proposed algorithm. The elements of the two



The figure illustrates a sample path $\{S_n(\mu) : 0 \le n \le 12\}$. If we set $m = 1$ and $L = 2$ then the corresponding stopping times are $D_1 = 4$, $U_1 = 6$, $D_2 = 9$. If in addition $U_2 = \infty$, then $S_n(\mu)$ stays below the bold dashed line for all $n \ge D_2$. Following Proposition 1 we can now evaluate M_n satisfying $\{M_n : n \le 9,\ S_n(\mu) \ge S_9(\mu) + 1\}$ In this example, at time $t = D_2 = 9$ the values of $\{M_n : 0 \le n \le 7\}$ can be calculated and we can update $C_{UB} \leftarrow S_{D_2}(\mu) + 1$. Notice that $S_8(\mu) \le S_9(\mu) + 1$ and therefore, in order to determine M_8 we need to keep on tracking the path until the next time we spot $U_n = \infty$.

Figure 2.2.1: Construction of Milestone Stopping Times

stopping times sequences interlace with each other (when finite) and their precise description follows next.

We start by fixing any $m > 0$, $L \ge 1$. Eventually we will choose m as small as possible subject to certain constraints described in Section 2.3, and then we can choose L as small as possible to satisfy

$$P(m < M_0 \le (L+1)m) > 0. \tag{2.2.1}$$

Typically, $L = 1$ is feasible. This constraint on L will be used in the proof of Proposition 1 and also in the implementation of Step 2 in Procedure 1 below.

Now set $D_0 = 0$. We observe the evolution of the process $(S_n(\mu) : n \ge 0)$ and detect the time D_1 (the first downward milestone),

$$D_1 = \inf\{n \ge D_0 : S_n(\mu) < -Lm\}.$$

Once D_1 is detected we check whether or not $\{S_n(\mu) : n \ge D_1\}$ ever goes above the height $S_{D_1}(\mu) + m$ (the first upward milestone); namely we define

$$U_1 = \inf\{n \ge D_1 : S_n(\mu) > m + S_{D_1}(\mu)\}$$

For now let us assume that we can check if $U_1 = \infty$ or $U_1 < \infty$ (how exactly to do so will be explained in Section 2.3). To continue simulating the rest of the path, namely $\{S_n(\mu) : n > D_1\}$, we potentially need to keep track of the conditional upper bound implied by the fact that $U_1 = \infty$. To this end, we introduce the conditional upper bound variable C_{UB} (initially $C_{UB} = \infty$). If at time D_1 we detect that $U_1 = \infty$, then we set $C_{UB} = S_{D_1}(\mu) + m$ and continue sampling the path of the random walk conditional on never crossing the upper bound $S_{D_1}(\mu) + m$, that is, conditional on $\{S_n(\mu) < C_{UB} : n > D_1\}$. Otherwise, if $U_1 < \infty$, we simulate the path conditional on $U_1 < \infty$, until we detect the time U_1. We continue on sequentially checking whenever a downward or an upward milestone is crossed as follows: For $j \geq 2$, define

$$\begin{aligned} D_j &= \inf\left\{n \geq U_{j-1} I\left(U_{j-1} < \infty\right) \vee D_{j-1} : S_n(\mu) < S_{D_{j-1}}(\mu) - Lm\right\} \\ U_j &= \inf\left\{n \geq D_j : S_n(\mu) - S_{D_j}(\mu) > m\right\}, \end{aligned} \tag{2.2.2}$$

with the convention that if $U_{j-1} = \infty$, then $U_{j-1} I\left(U_{j-1} < \infty\right) = 0$. Therefore, we have that $U_{j-1} I\left(U_{j-1} < \infty\right) > D_{j-1}$ if and only if $U_{j-1} < \infty$.

Let us define

$$\Delta = \inf\{D_n : U_n = \infty, n \geq 1\}. \tag{2.2.3}$$

So, for example, if $U_1 = \infty$ we have that $\Delta = D_1$ and the drifted random walk will never reach level $S_{D_1}(\mu) + m$ again. This allows us to evaluate M_0 by computing

$$M_0 = \max\left\{S_n(\mu) : 0 \leq n \leq \Delta\right\}. \tag{2.2.4}$$

Similarly, the event $U_j = \infty$, for some $j \geq 1$, implies that the level $S_{D_j}(\mu) + m$ is never crossed for all $n \geq D_j$, and we let $C_{UB} = S_{D_j}(\mu) + m$. The value of C_{UB} keeps updating as the random walk evolves, at times where $U_j = \infty$.

The advantage of considering these stopping times is the following: once we observed that some $U_j = \infty$, the values of $\left\{M_n : n \leq D_j, S_n(\mu) \geq S_{D_j}(\mu) + m\right\}$ are known without a need of further simulation. A detailed example is illustrated in Figure 2.2.

Before we summarize the properties of the stopping times D_n's and U_n's it will be useful to introduce the following. For any a and $b > 0$ let

$$\begin{aligned} T_b &= \inf\left\{n \geq 0 : S_n - \mu n > b\right\}, \\ T_{-b} &= \inf\left\{n \geq 0 : S_n - \mu n < -b\right\}, \\ P_a(\cdot) &= P\left(\cdot \mid S_0 = a\right). \end{aligned} \tag{2.2.5}$$

Proposition 1. *Set* $D_0 = 0$ *and let* $(D_n : n \geq 1)$ *and* $(U_n : n \geq 1)$ *be as (2.2.2). We have that*

$$P_0 \left(\lim_{n\to\infty} D_n = \infty\right) = 1 \quad \text{and} \quad P_0 \left(D_n < \infty\right) = 1, \quad \forall n \geq 1. \tag{2.2.6}$$

Furthermore,

$$P_0 \left(U_n = \infty, \, i.o.\right) = 1. \tag{2.2.7}$$

Proof. The statement in (2.2.6) follows easily from the Law of Large Numbers since $ES_1(\mu) = -\mu < 0$. Now we will verify that $P_0(U_n = \infty, \, i.o.) = 1$. Recall that U_1 was defined by $U_1 = \inf\{n \geq D_1 : S_n(\mu) - S_{D_1}(\mu) > m\}$. Therefore, since $ES_1(\mu) < 0$, for all $m \geq 0$ we have (see [Asmussen, 2003] p. 224),

$$P_0 \left(U_1 = \infty | S_1, ..., S_{D_1}\right) = P_0 \left(T_m = \infty\right) = P\left(M_0 \leq m\right) \geq P\left(M_0 = 0\right) > 0.$$

Our next goal is to show that for $j \geq 2$ we can find $\delta > 0$ such that

$$P_0 \left(U_j = \infty | S_1, ..., S_{D_j}, U_1, ..., U_{j-1}\right) \geq \delta > 0.$$

Suppose first that $U_l < \infty$ for each $l = 1, 2, ..., j-1$. Then, by the strong Markov property we have that

$$P_0 \left(U_j = \infty | S_1, ..., S_{D_j}, U_1, ..., U_{j-1}\right) = P_0 \left(T_m = \infty\right) \geq P\left(M_0 = 0\right) > 0.$$

Now suppose that $U_l = \infty$ for some $l \leq j-1$ and let $l^* = \max\{l \leq j-1 : U_l = \infty\}$. Define $r = S_{D_{l^*}}(\mu) + m - S_{D_j}(\mu) \geq (L+1)m$. Note that

$$P_0 \left(U_j = \infty | S_1, ..., S_{D_j}, U_1, ..., U_{j-1}\right) = P_0 \left(T_m = \infty | T_r = \infty\right). \tag{2.2.8}$$

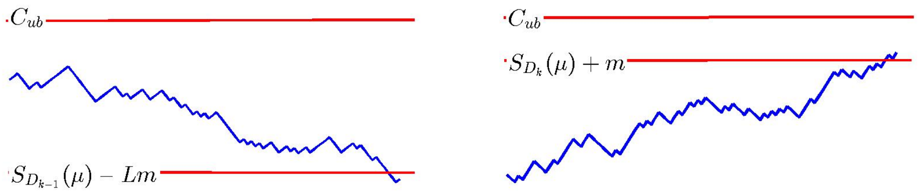
Keep in mind that the right hand side of (2.2.8) regards r as a deterministic constant and note that

$$P_0 \left(T_m = \infty | T_r = \infty\right) = P_0 \left(M_0 \leq m | M_0 \leq r\right) \geq \frac{P_0\left(M_0 = 0\right)}{P\left(M_0 \leq r\right)} \geq P_0\left(M_0 = 0\right) > 0 \tag{2.2.9}$$

Hence, we conclude that

$$P_0 \left(U_j = \infty | S_1, ..., S_{D_j}, U_1, ..., U_{j-1}\right) \geq P\left(M_0 = 0\right) := \delta > 0 \ \cdot$$

It then follows by the Borel-Cantelli lemma that $P_0(U_n = \infty, \text{ i.o.}) = 1$. □



a. Step 1.

b. Step 2.

Figure 2.2.2: High-level description of the algorithm

In the setting of Proposition 1, for each $k \geq 0$ we can define $N_0(k) = \inf\{n \geq 1 : D_n \geq k\}$ and $\mathcal{T}(k) = \inf\{j \geq N_0(k) + 1 : U_j = \infty\}$, both finite random variables such that

$$M_k = -S_k(\mu) + \max\{S_n(\mu) : k \leq n \leq D_{\mathcal{T}(k)}\} \tag{2.2.10}$$

In words, $D_{\mathcal{T}(k)}$ is the time, not earlier than k, at which we detect a second unsuccessful attempt at building an upward patch directly. The fact that the relation in (2.2.10) holds, follows easily by construction of the stopping times in (2.2.2). Note that it is important, however, to define $\mathcal{T}(k) \geq N_0(k) + 1$ so that $D_{N_0(k)+1}$ is computed first. That way we can make sure that the maximum of the sequence $(S_n(\mu) : n \geq k)$ is achieved between k and $D_{\mathcal{T}(k)}$ (see Figure 2.2).

Proposition 1 ensures that it suffices to sequentially simulate $(D_n : n \geq 0)$ and $(U_n : n \geq 1)$ jointly with the underlying random walk in order to sample from the sequence $(S_n(\mu), M_n : n \geq 0)$. This observation gives rise to our suggested scheme. The procedure sequentially constructs the random walk in the intervals $[D_{n-1}, D_n)$ for $n \geq 1$. Here is the high-level procedure to construct $(S_n(\mu), M_n : n \geq 0)$:

Procedure 1 Milestone Construction of $(S_n(\mu), M_n : n \geq 0)$ (see Figure 2.2.2)

At kth iteration, $k \geq 1$:

Step 1: "downward patch". Conditional on the path not crossing C_{UB} we simulate the path until we detect $D_k -$ the first time the path crosses the level $S_{D_{k-1}}(\mu) - Lm$ (see Figure 2.2.2a).

Step 2: "upward patch". Check whether or not the level $S_{D_k}(\mu) + m$ is ever crossed. That is, whether $U_k < \infty$ or not. If the answer is "Yes", then conditional on the path crossing the level $S_{D_k}(\mu) + m$ but not crossing the level C_{UB} we simulate the path until we detect U_k, the first time the level $S_{D_k}(\mu) + m$ is crossed (see Figure 2.2.2b). Otherwise $(U_j = \infty)$, and we can update C_{UB}: $C_{UB} \leftarrow S_{D_j}(\mu) + m$

The implementation of the steps in Procedure 1 will be discussed in detail in the next sections, culminating with the precise description given in Algorithm 3 at the end of Subsection 2.4.3. The following result summarizes the main contribution of this paper. The development in the next sections provides the proof of this result, which will ultimately be given after the description of Algorithm 3.

Throughout the rest of the paper a function evaluation is considered to be any of the following operations: evaluation of a sum, a product, the exponential of a number, the underlying increment distribution at a given point, the simulation of a uniform number, and the simulation of a single increment conditioned on lying on a given interval.

Theorem 2. *Suppose that* $E|X_k|^\beta < \infty$ *for some* $\beta > 1$. *If* $m > 0$ *is suitably chosen (see Subsection 2.3.1.1) then for each* $n \geq 0$ *deterministic it is possible to simulate exactly the sequence* $(D_j : 0 \leq j \leq n)$ *and* $(U_j : 0 \leq j \leq n)$ *jointly with* $(S_j(\mu) : j \leq n)$. *Therefore, (given our previous discussion on the evaluation of* M_k*) simulate exactly the sequence* $(S_k(\mu), M_k : 0 \leq k \leq n)$.

Moreover, if $\beta > 2$, *the expected number of function evaluations required to simulate* $(S_k(\mu), M_k : 0 \leq k \leq n)$ *is finite. In particular, since* $EN < \infty$ *for*

$$N = \inf\{k \geq 0 : M_k = 0\},$$

the expected running time to simulate $(S_k(\mu), M_k : 0 \leq k \leq N)$ *is also finite.*

2.3 Sampling M_0 jointly with $(S_1(\mu), ..., S_\Delta(\mu))$

The goal of this section is to sample exactly from the steady-state distribution of the single-server queue, namely M_0. To this end we need to simulate the sample path up to the first U_j such that $U_j = \infty$ (recall that Δ was defined to be the corresponding D_j). This sample path will be used in the construction of further steps in Procedure 1 .

Throughout this section, in order to simplify the exposition, we will assume that $E|X_k|^{2+\varepsilon} < \infty$ (i.e. $\beta = 2 + \varepsilon$). This will allow us to conclude that our algorithm has finite expected termination time. We will discuss the case $E|X_k|^{1+\varepsilon} < \infty$ only (for $\varepsilon \in (0,1)$) in Section 2.5 for completeness, but in such case the algorithm may take infinite expected time to terminate.

Let us recall the definition of the crossing stopping times T_b and T_{-b}, for $b > 0$, introduced in (2.2.5). Since we concentrate on M_0, we have that $C_{UB} = \infty$. We first need to explain a procedure to generate a Bernoulli random variable with success parameter $P_0(T_m < \infty)$, for suitably chosen $m > 0$. Also, this procedure, as we shall see will allow us to simulate $(S_1(\mu), ..., S_{T_m}(\mu))$ given that $T_m < \infty$.

2.3.1 Sampling $Ber(P_0(T_m < \infty))$ and $(S_1(\mu), ..., S_{T_m}(\mu))$ given $T_m < \infty$

Let us denote by J a Bernoulli random variable with success parameter $P_0(T_m < \infty)$. The constant $m > 0$ will be selected below in Subsection 2.3.1.1. There are several ways of sampling J, we use a strategy similar to that considered in [Murthy *et al.*, 2013], in connection to a different sampling problem.

In order to sample J we first introduce a partition on the natural numbers (i.e. the positive time line on the lattice) as follows. Let

$$n_k = 2^{k-1}, \quad k = 1, 2, \ldots \ . \tag{2.3.1}$$

This sequence define a partition of the natural numbers via the sets $[n_{k-1}, n_k - 1]$ for $k = 2, 3, ...$ Now, for $k = 2, 3, \ldots$ we consider the sets

$$\begin{aligned}
A_k &= \bigcup_{j=n_{k-1}}^{n_k-1} \left\{ X_j > (\mu j + m)^{1-\delta} \right\} \\
B_k &= \bigcap_{j=1}^{n_k-1} \left\{ X_j \leq (\mu n_{k-1} + m)^{1-\delta} \right\} \qquad (2.3.2) \\
&A_k^c \cap B_k^c
\end{aligned}$$

for some $\delta \in (0, \frac{1}{2}]$, also to be selected.

First, the algorithm samples the random variable $K \geq 2$, which has probability mass function $g(\cdot)$ that will be specified later. The random variable K relates to the partition on the natural numbers that was induced by (2.3.1) and $K = k$ will eventually imply that $T_m \in [n_{k-1}, n_k - 1]$. Given $K = k$, the algorithm then proposes a walk $(S_1(\mu), \ldots, S_{n_k-1}(\mu))$ via conditioning on one of three possible events described in terms of A_k, $B_k \cap A_k^c$ and $A_k^c \cap B_k^c$ with equal probability (i.e. with probability $\frac{1}{3}$ each). Conditioning on A_k and $A_k^c \cap B_k^c$ will be handled using a mixtures based on individual large-jump-events of the form $\{X_j > (\mu j + m)^{1-\delta}\}$. Conditioning on B_k will be handled using an exponential tilting of the distribution of X_j given that $\{X_j < (\mu j + m)^{1-\delta}\}$. The tilting parameter will be selected via

$$\theta_k = \frac{\gamma}{n_{k-1}\mu + m}, \qquad (2.3.3)$$

for some $\gamma > 0$.

In order to describe all of these conditional sampling procedures we need to provide some definitions and state auxiliary lemmas which will be proved in the appendix.

We will start by specifying the probability mass function $\{g(k), k \geq 2\}$. Consider Y, a Pareto distributed random variable with some regularly varying index $\alpha > 0$, namely,

$$P(Y > y) = \frac{1}{(1+y)^{\alpha}},$$

for $y \geq 0$. Conditions on $\alpha > 0$ will be imposed below. Let

$$\bar{G}(t) = \int_t^{\infty} P(Y > s)\, ds$$

Then we set for $k = 2, 3, \ldots$

$$g(k) = P(K = k) = \frac{\bar{G}(m + \mu n_{k-1}) - \bar{G}(m + \mu n_k)}{\bar{G}(m + \mu n_1)}. \qquad (2.3.4)$$

Let us impose conditions on δ, α, m and γ that will be assumed for the implementation of the algorithm.

2.3.1.1 Assumptions imposed on the parameters δ, α, m and γ

In addition to $\delta \in \left(0, \frac{1}{2}\right]$, and (2.2.1), assume that $m \geq 1$ is selected large enough so that

$$\frac{E\left(X^2\right)}{m^{2(1-\delta)}} \leq \frac{1}{2}, \tag{2.3.5}$$

and that the following inequalities hold:

$$\sup_{z\in\mu\cdot\{2^k:k\geq 0\}} \frac{6\left(1+2z+m\right)^{\alpha} P\left(X > \left(z+m\right)^{1-\delta}\right)}{\left(\alpha-1\right)\left(m+1\right)^{\alpha-1}\mu} \leq 1, \tag{2.3.6}$$

$$\sup_{z\in\mu\cdot\{2^k:k\geq 0\}} \frac{\exp\left(-\gamma\left(m+z\right)^{\delta} + \frac{\gamma^2 e^{\gamma} E\left(X^2\right) z}{\left(m+z\right)^{2(1-\delta)}\mu} + 4\frac{z}{\mu} P\left(X > \left(z+m\right)^{1-\delta}\right)\right)}{3^{-1}\left(\alpha-1\right)\left(m+1\right)^{\alpha-1}\left(1+2z+m\right)^{-\alpha} z} \leq 1. \tag{2.3.7}$$

Inequalities (2.3.6) and (2.3.7) are used during the proofs of Lemmas 3 and 4 , respectively. Inequality (2.3.5) appears in a simple technical step leading to (2.3.7).

In Appendix A.1 we will discuss how equations (2.3.5)-(2.3.7) can always be satisfied under our assumptions on the increments X_k.

2.3.1.2 Some technical lemmas underlying the description of our algorithm

Using the previous assumptions we now are ready to discuss a series of technical lemmas that are the basis for our algorithm.

Lemma 3. *Under (A.1.2) (see Appendix A.1) we have that*

$$\frac{3P\left(A_k\right)}{g\left(k\right)} \leq 1, \ \forall k \geq 2. \tag{2.3.8}$$

Proof. See Appendix A.2 □

On the event B_k we sample the path $\left(S_1\left(\mu\right), ..., S_{n_k-1}\left(\mu\right)\right)$ using an exponential tilting. Specifically, we sample the increments, $\left(X_j : 1 \leq j \leq n_k - 1\right)$, conditional on the event B_k and tilted with parameter θ_k up to time $\min\left\{T_m, n_k - 1\right\}$, where

$$\theta_k = \frac{\gamma}{C_k^{1-\delta}}, \quad \text{and} \quad C_k := \left(n_{k-1}\mu + m\right).$$

Recall that $\gamma > 0$ has been implicitly constrained due to (2.3.7). The corresponding log-mgf is given by

$$\psi_k\left(\theta_k\right) := \log\left(\frac{E\left[\exp\left\{\theta_k X\right\} I\left(X \leq C_k^{1-\delta}\right)\right]}{P\left(X \leq C_k^{1-\delta}\right)}\right).$$

The likelihood ratio between $P\left(X_j \in \cdot | X_j \leq C_k^{1-\delta}\right)$ and the tilted distribution (to be used in an IID way for $1 \leq j \leq n_k - 1$) denoted via $P_{k,1}\left(\cdot\right)$ is given by

$$\frac{dP_{k,1}}{dP}\left(X\right) = \frac{I\left(X \leq C_k^{1-\delta}\right)\exp\left(\theta_k X - \psi_k\left(\theta_k\right)\right)}{P\left(X \leq C_k^{1-\delta}\right)}. \tag{2.3.9}$$

Now we summarize some bounds for this likelihood ratio.

Lemma 4. *Under conditions (2.3.5)-(2.3.7) we have that*

$$\frac{3\exp(-\theta_k S_{T_m} + T_m\psi_k\left(\theta_k\right))}{g\left(k\right)} \leq 1, \ \forall k \geq 2. \tag{2.3.10}$$

Proof. See Appendix A.3 □

As the final piece we will note the following.

Lemma 5. *Then, under (A.1.1), and (A.1.2) we have that*

$$\frac{3P\left(B_k^c\right)}{g\left(k\right)} \leq 1, \ \forall k \geq 2. \tag{2.3.11}$$

Proof. See Appendix A.4 □

2.3.1.3 Algorithm for sampling $Ber\left(P_0\left(T_m < \infty\right)\right)$ jointly with $\left(S_1\left(\mu\right), ..., S_{T_m}\left(\mu\right)\right)$ given $T_m < \infty$

Now we are ready to fully discuss our algorithm to sample J and $\omega = \left(S_1, ...S_{T_m}\right)$ given $T_m < \infty$. In addition to the random variable K following the probability mass function $g\left(\cdot\right)$, let us introduce a random variable Z uniformly distributed on $\{0, 1, 2\}$ and independent of K. Finally, we also introduce $V \sim U\left(0, 1\right)$ independent of everything else.

If $Z = 0$, then we sample the path $(S_1, ..., S_{n_k - 1})$ conditional on A_k (denote $P_{k,0}(\cdot) = P(\cdot | A_k)$). This will be explained in Subsection 2.3.1.4, the sample takes $O(n_k)$ function evaluations to be produced. Then we let

$$J = I(V \leq 3P(A_k) I(T_m \in [n_{k-1}, n_k - 1])/g(k)).$$

Owing to Lemma 3, we have that

$$\frac{3P(A_k)}{g(k)} \leq 1, \ \forall k \geq 2. \tag{2.3.12}$$

If $Z = 1$, we sample $(S_1(\mu), ..., S_{n_k - 1}(\mu))$ by applying each increment X_j conditional on $\{X_j \leq (\mu n_{k-1} + m)^{1-\delta}\}$ for $j \in \{1, ..., n_k - 1\}$ in an IID way each following the exponential tilting (2.3.9). This sampling distribution is denoted via $P_{k,1}(\cdot)$. The simulation of each increment is done using Acceptance/Rejection, as we shall explain, and the overall sampling $\{X_j : j \leq n_k - 1\}$ takes $O(n_k)$ function evaluations, see Subsections 2.3.1.5. Additional discussion on the evaluation $\psi_k(\theta_k)$ in $O(n_k)$ function evaluations is given in Subsection 2.5.2. We then set

$$J = I(V \leq 3 \cdot \exp\{-\theta_k S_{T_m} + T_m \psi_k(\theta_k)\} I(T_m \in [n_{k-1}, n_k - 1], A_k^c, B_k)/g(k)).$$

Observe that Lemma 4 guarantees the inequality

$$\frac{3\exp\{-\theta_k S_{T_m} + T_m \psi_k(\theta_k)\}}{g(k)} \leq 1, \ \forall k \geq 2. \tag{2.3.13}$$

Finally, if $Z = 2$, we sample the path $(S_1(\mu), ..., S_{n_k - 1}(\mu))$ conditional on the event B_k^c (denote $P_{k,2}(\cdot) = P(\cdot | B_k^c)$). This is done in a completely analogous manner as in Subsection 2.3.1.4, thus taking $O(n_k)$ function evaluations. We then let

$$J = I(V \leq 3P(B_k^c) I(T_m \in [n_{k-1}, n_k - 1], A_k^c, B_k^c)/g(k)).$$

Here the inequality

$$\frac{3P(B_k^c)}{g(k)} \leq 1, \ \forall k \geq 2, \tag{2.3.14}$$

is obtained thanks to Lemma 5.

Upon termination we will output the pair (J, ω). If $J = 1$, then we set $\omega = (S_1(\mu), ..., S_{T_m}(\mu))$. Otherwise ($J = 0$), we set $\omega = [\,]$, the empty vector. The precise description of the algorithm is given next.

Algorithm 1: Sampling $Ber\left(P_0\left(T_m<\infty\right)\right)$ and $\left(S_1\left(\mu\right),...,S_{T_m}\left(\mu\right)\right)$ given $T_m<\infty$

```
Input: $g\left(\cdot\right)$ as in (2.3.4), with $\alpha,\delta,m,\gamma$ satisfying the conditions in Section 2.3.1.1
       and $L$ as in (2.2.1).
Output: $J\sim Ber\left(P_0\left(T_m<\infty\right)\right)$ and $\omega$. If $J=1$, then $\omega=\left(S_1\left(\mu\right),\dots,S_{T_m}\left(\mu\right)\right)$.
    Otherwise ($J=0$), $\omega=[\,]$ // If $J=0$, then $\omega$ equals to the empty
    vector
Sample a time $K$ with probability mass function $g\left(k\right)=P\left(K=k\right)$
Sample $Z\sim Unif\left\{0,1,2\right\}$
Sample $V\sim U\left(0,1\right)$ independent of everything
Given $Z$ and $K=k$ sample $\left(S_1,\dots,S_{n_k}\right)$ as follows:
if $Z=0$ then
    Sample $\tilde{w}=\left(S_j:\,j\leq n_k-1\right)$ from $P_{k,0}\left(\cdot\right):=P\left(\cdot|\,A_k\right)$
    if $V\leq\frac{3P(A_k)}{g(k)}I\left(A_k,\,T_m\in\left[n_{k-1},n_k-1\right]\right)$ then
        $J=1$
    else
        $J=0$
if $Z=1$ then
    Sample $\tilde{w}=\left(S_j:\,j\leq T_m\wedge\left(n_k-1\right)\right)$ from $P_{k,1}\left(\cdot\right)$

        $dP_{k,1}\left(\tilde{w}\right)=\exp\left\{\theta_k S_{T_m\wedge\left(n_{k-1}\right)}-\left(T_m\wedge\left(n_k-1\right)\psi_k\left(\theta_k\right)\right)\right\}dP\left(\tilde{w}\right)$

    if $V\leq\frac{3\exp\{-\theta_k S_{T_m}+T_m\psi_k(\theta_k)\}}{g(k)}I\left(B_k,\,A_k^c,\,T_m\in\left[n_{k-1},n_k-1\right]\right)$ then
        $J=1$
    else
        $J=0$
if $Z=2$ then
    Sample $\tilde{w}=\left(S_j:\,j\leq n_k-1\right)$ from $P_{k,2}\left(\cdot\right):=P\left(\cdot|\,B_k^c\right)$
    if $V\leq\frac{3P\left(B_k^c\right)}{g(k)}I\left(B_k^c\,A_k^c,\,T_m\in\left[n_{k-1},n_k-1\right]\right)$ then
        $J=1$
    else
        $J=0$
if $J=1$ then
    Output $\left(J,\omega\right)$, where $\omega=\left(S_j\left(\mu\right):\,1\leq j\leq T_m\right)$ // Recall: $S_j\left(\mu\right)=S_j-\mu j$.
else
    Output $\left(J,\omega\right)$, where $\omega=[\,]$ and $J=0$.
```

We now provide the following result which justifies the validity of the algorithm.

Proposition 6. *The output* J *is Bernoulli with success parameter* $P_0(T_m < \infty)$ *and* ω *follows the required distribution of* $(S_1, \ldots, S_{T_m})$ *given* $T_m < \infty$*. Moreover, if* $E|X_1|^{2+\varepsilon} < \infty$*, then the expected number of function evaluations required to sample* J *and* ω *is finite.*

Proof. To verify that indeed $J \sim Ber(P_0(T_m < \infty))$, let $P'(\cdot)$ denote the joint probability distribution of K, Z, $(S_1, ..., S_{n_K - 1})$, and J induced by the algorithm. Note, of course, that $n_K - 1 \geq T_m$ under $P'(\cdot)$. In addition, observe that

$$\begin{aligned} P'(J = 1|\, Z = 0, K = k) &= \tfrac{3P(A_k)}{g(k)} \cdot P_0(T_m \in [n_{k-1}, n_k - 1] \,|\, A_k) \\ &= \tfrac{3}{g(k)} \cdot P_0(T_m \in [n_{k-1}, n_k - 1],\, A_k). \end{aligned} \tag{2.3.15}$$

Let $r_{k,1} := \exp(-\theta_k S_{T_m} + T_m \psi(\theta_k)) I(B_k, A_k^c, T_m \in [n_{k-1}, n_k - 1])$, and define $E_{k,1}(\cdot)$ to be the expectation operator associated to the exponential tilting distribution with parameter θ_k applied to the random variables $X_1, ..., X_{n_k - 1}$ (see (2.3.9)). Note that,

$$\begin{aligned} P'(J = 1|\, Z = 1, K = k) &= \tfrac{3}{g(k)} E_{k,1}[r_{k,1}] \\ &= \tfrac{3}{g(k)} P_0(B_k, A_k^c, T_m \in [n_{k-1}, n_k - 1]) \end{aligned} \tag{2.3.16}$$

Finally,

$$P'(J = 1|\, Z = 2, K = k) = \tfrac{3}{g(k)} P_0(B_k^c, A_k^c, T_m \in [n_{k-1}, n_k - 1]) \tag{2.3.17}$$

Combining (2.3.15)-(2.3.17) we have

$$\begin{aligned} &P'(J = 1) = \\ &= \sum_{k=2}^{\infty} \tfrac{1}{3} \left(P'(J = 1|\, Z = 0, K = k) + P'(J = 1|\, Z = 1, K = k) + P'(J = 1|\, Z = 2, K = k) \right) g(k) \\ &= \sum_{k=2}^{\infty} P_0(T_m \in [n_{k-1}, n_k - 1],\, A_k) + P_0(B_k,\, A_k^c,\, T_m \in [n_{k-1}, n_k - 1]) \\ &\qquad + P_0(B_k^c, A_k^c, T_m \in [n_{k-1}, n_k - 1]) \\ &= \sum_{k=2}^{\infty} P_0(T_m \in [n_{k-1}, n_k - 1]) = P_0(T_m < \infty). \end{aligned} \tag{2.3.18}$$

Similarly we can verify that if $J = 1$, $\omega = (S_1, ..., S_{T_m})$ follows the conditional law

$P\left(\omega \in \cdot | T_m < \infty\right)$. Just note that for any F,

$$\begin{aligned}
P'\left(\omega \in F, J = 1 | K = k, Z = 0\right) &= P_0\left(\omega \in F, T_m \in \left[n_{k-1}, n_k - 1\right] | A_k\right) \cdot \frac{3P\left(A_k\right)}{g\left(k\right)} \\
&= P_0\left(\omega \in F, T_m \in \left[n_{k-1}, n_k - 1\right], A_k\right) \cdot \frac{3}{g\left(k\right)}, \\
P'\left(\omega \in F, J = 1 | K = k, Z = 1\right) &= P_0\left(\omega \in F, T_m \in \left[n_{k-1}, n_k - 1\right] A_k^c | B_k\right) \cdot \frac{3P\left(B_k\right)}{g\left(k\right)} \\
&= P_0\left(\omega \in F, T_m \in \left[n_{k-1}, n_k - 1\right] A_k^c, B_k\right) \cdot \frac{3}{g\left(k\right)}, \\
P'\left(\omega \in F, J = 1 | K = k, Z = 2\right) &= P_0\left(\omega \in F, A_k^c, T_m \in \left[n_{k-1}, n_k - 1\right] | B_k^c\right) \cdot \frac{3P\left(B_k^c\right)}{g\left(k\right)} \\
&= P_0\left(\omega \in F, T_m \in \left[n_{k-1}, n_k - 1\right], B_k^c, A_k^c\right) \cdot \frac{3}{g\left(k\right)}.
\end{aligned}$$

Consequently, combining these terms

$$\begin{aligned}
&P'\left(\omega \in F, J = 1\right) = \\
&= \sum_{k=2}^{\infty} [P_0\left(\omega \in F, T_m \in \left[n_{k-1}, n_k - 1\right], A_k\right) + P_0\left(\omega \in F, T_m \in \left[n_{k-1}, n_k - 1\right] A_k^c, B_k\right) \\
&\qquad + P_0\left(\omega \in F, T_m \in \left[n_{k-1}, n_k - 1\right], B_k^c, A_k^c\right)] \\
&= \sum_{k=2}^{\infty} P_0\left(\omega \in F, T_m \in \left[n_{k-1}, n_k - 1\right]\right) = P_0\left(\omega \in F, T_m < \infty\right).
\end{aligned}$$

Since $P'\left(J = 1\right) = P_0\left(T_m < \infty\right)$, we conclude that indeed

$$P'\left(\omega \in F | J = 1\right) = P_0\left(\omega \in F | T_m < \infty\right).$$

We now argue that the expected number of function evaluations required to generate (J, ω) has finite mean. Let us assume that sampling from $P_{k,0}\left(\cdot\right), P_{k,1}\left(\cdot\right)$, and $P_{k,2}\left(\cdot\right)$ takes $O(n_k)$ function evaluations (a fact that it is not difficult to see, but nonetheless we will justify in Subsections 2.3.1.4 and 2.3.1.5). Then, we note that each proposal ω takes on the order of

$$O(\sum_{k=2}^{\infty} n_k g\left(k\right)) \leq O(\sum_{k=2}^{\infty} n_k^2 P\left(Y > n_{k-1}\mu + m\right)) < \infty$$

function evaluations; the sum is finite assuming that $\alpha > 2$, as indicated in (A.1.1). □

We close this section explaining how to sample from $P_{k,0}\left(\cdot\right), P_{k,1}\left(\cdot\right)$, and $P_{k,2}\left(\cdot\right)$. We will also verify that it takes $O(n_k)$ function evaluations to sample ω in each of these three cases as claimed in the end of Proposition 6.

2.3.1.4 Sampling from $P_{k,0}(\cdot)$ and $P_{k,2}(\cdot)$

We now explain how to use Acceptance / Rejection to obtain a sample from $P_{k,0}(\cdot)$ (i.e. sampling $(S_1, ..., S_{n_k-1})$ given A_k). Our proposal distribution, which we denote by $Q(\cdot)$, is based on a mixture $P(\cdot)$ and another distribution which we denote by $\bar{P}(\cdot)$ to be described momentarily. In particular, we shall set $Q = \frac{1}{2}P + \frac{1}{2}\bar{P}$. As we shall see, the reason for introducing P is to make sure that the acceptance ratio is bounded uniformly over μ. This will be relevant in our discussion on mixing time in heavy-traffic in Section 2.6 (i.e. when μ is close to zero). If μ is not close to zero then we can simply select $Q = \bar{P}$ and the acceptance ratio will be bounded uniformly in k, but not as $\mu \to 0$.

The distribution of $(S_1, ..., S_{n_k-1})$ under $\bar{P}(\cdot)$ is better described algorithmically. First, we sample T_k with probability mass function $r_k(\cdot)$ given by

$$r_k(j) = \frac{P(X_j > (\mu j + m)^{1-\delta})}{\sum_{j=n_{k-1}}^{n_k-1} P(X_j > (\mu j + m)^{1-\delta})},$$

for $j \in \{n_{k-1}, \ldots, n_k - 1\}$. Next, given $T_k = j$, sample X_j conditional on $X_j > (\mu j + m)^{1-\delta}$. Finally, sample X_i, for $i \neq j$ and $1 \leq i \leq n_k - 1$ from the nominal (unconditional) distribution. We then obtain that

$$\frac{d\bar{P}}{dP}(X_1, ..., X_{n_k-1}) = \frac{\sum_{j=n_{k-1}}^{n_k-1} I\left(X_j > (\mu j + m)^{1-\delta}\right)}{\sum_{j=n_{k-1}}^{n_k-1} P(X_j > (\mu j + m)^{1-\delta})}.$$

Therefore, with $P_{k,0}(\cdot) = P(\cdot|A_k)$ we obtain that

$$\begin{aligned}
&\frac{I(A_k)}{P(A_k)} \cdot \frac{dP}{dQ}(X_1, ..., X_{n_k-1}) = \\
&= 2\frac{I(A_k)}{P(A_k)} \cdot \frac{\sum_{j=n_{k-1}}^{n_k-1} P(X_j > (\mu j + m)^{1-\delta})}{\sum_{j=n_{k-1}}^{n_k-1} I\left(X_j > (\mu j + m)^{1-\delta}\right) + \sum_{j=n_{k-1}}^{n_k-1} P(X_j > (\mu j + m)^{1-\delta})} \\
&\leq c_k := \frac{2}{P(A_k)} \cdot \frac{\sum_{j=n_{k-1}}^{n_k-1} P(X_j > (\mu j + m)^{1-\delta})}{1 + \sum_{j=n_{k-1}}^{n_k-1} P(X_j > (\mu j + m)^{1-\delta})}.
\end{aligned}$$

Consequently, in order to sample from $P_{k,0}(\cdot)$ it suffices to propose from $Q(\cdot)$ and accept with probability

$$\begin{aligned}
q \quad &:= \frac{1}{c_k} \cdot \frac{I(A_k)}{P(A_k)} \cdot \frac{dP}{dQ}(X_1, ..., X_{n_k-1}) \\
&= I(A_k) \cdot \frac{1 + \sum_{j=n_{k-1}}^{n_k-1} P(X_j > (\mu j + m)^{1-\delta})}{\sum_{j=n_{k-1}}^{n_k-1} I\left(X_j > (\mu j + m)^{1-\delta}\right) + \sum_{j=n_{k-1}}^{n_k-1} P(X_j > (\mu j + m)^{1-\delta})}.
\end{aligned}$$

We note that the expected number of proposals required to accept is c_k. Moreover, as we shall quickly verify, c_k is bounded uniformly both in k and $\mu > 0$. To see this, use the fact that for $x \geq 0$, $1 - x \leq \exp(-x)$ and conclude that

$$\begin{aligned} P(A_k) &= 1 - \prod_{j=n_{k-1}}^{n_k - 1} \left(1 - P\left(X_j > (\mu j + m)^{1-\delta}\right)\right) \\ &\geq 1 - \exp\left(- \sum_{j=n_{k-1}}^{n_k - 1} P(X_j > (\mu j + m)^{1-\delta})\right). \end{aligned}$$

Let us write

$$\Lambda := \Lambda(k, \mu) = \sum_{j=n_{k-1}}^{n_k - 1} P\left(X_j > (\mu j + m)^{1-\delta}\right)$$

and therefore obtain that

$$c_k \leq \frac{2}{1 - \exp(-\Lambda)} \cdot \frac{\Lambda}{1 + \Lambda} \leq 4I\left(\Lambda \in \left[0, \frac{1}{2}\right]\right) + 6I\left(\Lambda \geq \frac{1}{2}\right) \leq 6.$$

We suggest applying a completely analogous randomization procedure to sample $P_{k,2}(\cdot)$, which corresponds to sampling given the event

$$B_k^c = \bigcup_{j=1}^{n_k - 1} \left\{X_j > (\mu n_{k-1} + m)^{1-\delta}\right\}.$$

A very similar argument as the one just discussed shows that the number of proposals required to accept is also uniformly bounded over k and μ. We therefore conclude that it takes $O(n_k)$ function evaluations to sample ω both under $P_{k,0}(\cdot)$ and $P_{k,2}(\cdot)$.

2.3.1.5 Sampling from $P_{k,1}(\cdot)$

In order to simulate from $P_{k,1}(\cdot)$ we use Acceptance / Rejection. We propose from $P(\cdot)$ (the nominal distribution). Using the fact that $\theta_k = \gamma / C_k^{1-\delta}$, note that

$$dP_{k,1} = \frac{I\left(X \leq C_k^{1-\delta}\right) \exp(\theta_k X - \psi_k(\theta_k))}{P\left(X \leq C_k^{1-\delta}\right)} dP \tag{2.3.19}$$

$$\leq \frac{I\left(X \leq C_k^{1-\delta}\right) \exp(\gamma - \psi_k(\theta_k))}{P\left(X \leq C_k^{1-\delta}\right)} dP \leq \frac{\exp(\gamma - \psi_k(\theta_k))}{P\left(X \leq C_k^{1-\delta}\right)} dP. \tag{2.3.20}$$

So, in order to sample from $P_{k,1}(\cdot)$ it suffices to propose from $P(\cdot)$ and accept with probability

$$q(\omega) := \frac{P\left(X \leq C_k^{1-\delta}\right)}{\exp(\gamma - \psi_k(\theta_k))} \frac{dP_{k,1}}{dP} = \exp(\theta_k X - \gamma) I\left(X \leq C_k^{1-\delta}\right).$$

The expected number of proposals required to obtain a successful sample X from $P_{k,1}(\cdot)$ is equal to

$$\frac{\exp(\gamma - \psi_k(\theta_k))}{P\left(X \leq C_k^{1-\delta}\right)} \leq \frac{\exp(\gamma)}{P(X \leq m)} < \infty,$$

which is clearly uniformly bounded in k. So each increment takes $O(1)$ time to be simulated and therefore we conclude it takes $O(n_k)$ function evaluations to simulate ω under $P_{k,1}(\cdot)$.

2.3.2 Building M_0 and $(S_1(\mu), ..., S_\Delta(\mu))$ from downward and upward patches

Before we move on to the algorithm let us define the following. Given a vector $\mathbf{s}$, of dimension $d \geq 1$, we let $\mathbf{L}(\mathbf{s}) = \mathbf{s}(d)$ (i.e. the d-th component of the vector $\mathbf{s}$).

Proposition 7. *The output of Algorithm 2 has the correct distribution according to (2.2.3) and (2.2.4). Moreover, if* $E|X_1|^{2+\varepsilon} < \infty$*, then the expected number of function evaluations required to sample* M_0 *and* $(S_1(\mu), ..., S_\Delta(\mu))$ *is finite.*

Proof. The fact that the output has the correct distribution follows directly from our discussion leading to (2.2.4) and from Proposition 6, which also implies that simulating a single replication of (J, ω) using Algorithm 1 requires finite expected running time. But Algorithm 2 requires a number of calls to Algorithm 1 which is geometrically distributed with mean $\frac{1}{P_0(T_m=\infty)} < \infty$. Therefore, by Wald's identity (see [Durrett, 2005], p. 178) we conclude the finite expected running time of Algorithm 2. □

```
Algorithm 2: Sampling $M_0$ and $(S_1(\mu), ...., S_\Delta(\mu))$
  Input: Same as Algorithm 1
  Output: The path $(S_1(\mu), ...., S_\Delta(\mu))$
  Initialization $\mathbf{s} \leftarrow []$, $F \leftarrow 0$, $\mathbf{L} = 0$
      // initially s is the empty vector,the variable $\mathbf{L}$ represents the
      last position of the drifted random walk
  while $F = 0$ do
      Sample $(S_1(\mu), \ldots, S_{T_{-Lm}}(\mu))$ given $S_0(\mu) = 0$
      $\mathbf{s} = [\mathbf{s}, \mathbf{L} + S_1(\mu), \ldots, \mathbf{L} + S_{T_{-Lm}}(\mu)]$
      $\mathbf{L} = \mathbf{L}(\mathbf{s})$
      Call Algorithm 1 and obtain $(J, w)$
      if $J = 1$ then
          Set $\mathbf{s} = [\mathbf{s}, \mathbf{L} + \omega]$
      else
          $F \leftarrow 1$  $(J = 0)$
  Output s.
```

2.4 From M_0 to $(S_k(\mu), M_k : k \geq 0)$: Implementation of Procedure 1

In this section we will explain in detail how to implement the steps behind the construction of the sequence $(S_n(\mu), M_n : n \geq 0)$ that were described in Procedure 1 . We will be calling Algorithm 1 and Algorithm 2 repeatedly.

2.4.1 Implementing Step 1 in Procedure 1

In Step 1 we need to sample a downward patch of the drifted random walk $(S_n(\mu) : n \geq 0)$. The goal is to detect the time where the next downward milestone is crossed, namely the next element in the sequence $(D_n : n \geq 1)$, conditional on the event that the level C_{UB} is not crossed. To this end, let us invoke a result in [Blanchet and Sigman, 2011].

Lemma 8. *Let* $0 < a < b \leq \infty$ *and consider any sequence of bounded positive measurable* *functions* $f_k : \mathbb{R}^{k+1} \longrightarrow [0,\infty)$.

$$E_0\left[f_{T_{-a}}\left(S_0(\mu),...,S_{T_{-a}}(\mu)\right)|T_b=\infty\right]=$$
$$=\frac{E_0\left[f_{T_{-a}}\left(S_0(\mu),...,S_{T_{-a}}(\mu)\right)I\left(S_i(\mu)<b,\,\forall i<T_{-a}\right)P_{S_{T_{-a}}}\left(T_b=\infty\right)\right]}{P_0\left(T_b=\infty\right)}$$

So, if $P^*(\cdot)=P_0(\cdot|T_b=\infty)$, *then*

$$\frac{dP^*}{dP_0}=\frac{I\left(S_i(\mu)<b,\forall i<T_{-a}\right)P_{S_{T_{-a}}}\left(T_b=\infty\right)}{P_0\left(T_b=\infty\right)}\leq\frac{1}{P_0\left(T_b=\infty\right)}. \tag{2.4.1}$$

The result of Lemma 8 holds due to the strong Markov property. Lemma 8 enables us to sample a downward patch by means of the Acceptance/Rejection method using the nominal (i.e. unconditional) distribution as proposal. More precisely, suppose that our current position is $S_{D_j}(\mu)$ and we know that the random walk will never reach position C_{UB} (say, if $U_j=\infty$ then $C_{UB}=S_{D_j}(\mu)+m$). Next we need to simulate the path up to time D_{j+1}. Lemma 8 says that we can propose a downward patch $s_1:=S_1(\mu),...,s_{T_{-Lm}}:=S_{T_{-Lm}}(\mu)$, under the nominal probability given $S_0(\mu)=0$ and $S_i(\mu)\leq m$ for $i\leq T_{-Lm}$. Then we accept the downward patch with probability $P_0(T_\sigma=\infty)$, where $\sigma=C_{UB}-S_{D_j}(\mu)-s_{T_{-Lm}}$. For example, if $U_j=\infty$ then $\sigma=m-s_{T_{-Lm}}\geq(L+1)m$.

Of course, to accept, we can simulate a Bernoulli, say B, with probability $P_0(T_\sigma=\infty)$ by calling Algorithm 1 with $m\longleftarrow\sigma$ and returning $B=1-J$. If the downward patch $\left(s_1,...,s_{T_{-Lm}}\right)$ is accepted we concatenate to produce the output

$$\begin{aligned}&\left(S_0(\mu),...,S_{D_j}(\mu),S_{D_j+1}(\mu),...,S_{D_{j+1}}(\mu)\right)\\&=(S_0(\mu),...,S_{D_j}(\mu),S_{D_j}(\mu)+s_1,...,S_{D_j}(\mu)+s_{T_{-Lm}}).\end{aligned}$$

Otherwise, we keep simulating downward-patch proposals until acceptance.

2.4.2 Implementing Step 2 in Procedure 1

Assume we have finished generating the path up to time D_{j+1} as explained in Subsection 2.4.1. At this point we let $\sigma=C_{UB}-S_{D_{j+1}}(\mu)\geq(L+1)m$ and define

$$\begin{aligned}\xi&=P_0\left(U_{j+1}<\infty\right|S_1,\ldots,S_{D_{j+1}},U_1,\ldots,U_j)\\&=P_0\left(T_m<\infty\right|T_\sigma=\infty)=P_0\left(M_0>m\,|\,M_0\leq\sigma\right).\end{aligned}$$

Observe that assumption in equation (2.2.1) ensures that $\xi > 0$. We will explain how to simulate $B \sim Ber\left(\xi\right)$. First, we call Algorithm 2 and obtain the output $\omega = (s_1, ..., s_\Delta)$. We compute M_0 according to (2.2.4) and keep calling Algorithm 2 until we obtain $M_0 \leq \sigma$, at which point we set $B = I\left(M_0 > m\right)$. Of course, we obtain $B \sim Ber\left(\xi\right)$ and if $B = 1$ we can write

$$\left(S_{D_{j+1}}\left(\mu\right), S_{D_{j+1}+1}\left(\mu\right), \ldots, S_{U_{j+1}}\left(\mu\right)\right) = \left(S_{D_{j+1}}\left(\mu\right), S_{D_{j+1}+1}\left(\mu\right) + s_1, ..., S_{D_j}\left(\mu\right) + s_\Delta\right). \tag{2.4.2}$$

Otherwise, $B = 0$, we could simply declare $U_{j+1} = \infty$, update $C_{UB} \leftarrow S_{D_{j+1}}\left(\mu\right) + m$ and proceed to the next iteration.

Breaking the path into “upward” and “downward” patches helps to conceptualize the logic of our method. However, it is not an efficient way of implementing the method. A more efficient implementation would be to sequentially generate versions of $\omega = (s_1, ..., s_\Delta)$ as long as $M_0 \leq\ m$. We can then output the right hand side of (2.4.2) even when $B = 0$, because the path has been simulated according to the correct distribution given $T_\sigma = \infty$. We provide a precise description of this implementation in Algorithm 3 in the next section.

2.4.3 Our algorithm to sample $\left(S_k\left(\mu\right), M_k\ : 0 \leq k \leq n\right)$ and Proof of Theorem 2

We close this section by giving the explicit implementation of our general method outlined in Subsections 2.4.1 and 2.4.2. In order to describe the procedure, let us recall some definitions. Given a vector $\mathbf{s}$ of dimension $d \geq 1$, let $\mathbf{L}\left(\mathbf{s}\right) = \mathbf{s}\left(d\right)$ (the last element of the vector) and set $\mathbf{d}\left(\mathbf{s}\right) = d$ (the length of the vector). The implementation is given in Algorithm 3.

Proof of Theorem 2. The validity of Algorithm 3 is justified following the same logic as in Proposition 7. The only difference here is that the number of trials required to simulate each upward patch is geometrically distributed with a mean which is bounded by $1/P_0\left(M_0 = 0\right) < \infty$, following the reasoning behind (2.2.9). Also note that

$$E_0(T_m I(T_m < \infty)) \leq \sum_{k=2}^{\infty} n_k g\left(k\right) < \infty.$$

Moreover, if $\sigma \geq (L+1)m$, by assumption (2.2.1)

$$E_0\left(T_m | T_m < \infty, T_\sigma = \infty\right) \leq \frac{E_0\left(T_m I(T_m < \infty)\right)}{P_0\left(T_m < \infty, T_\sigma = \infty\right)} \leq \frac{E_0\left(T_m I(T_m < \infty)\right)}{P_0\left(m < M_0 \leq \sigma\right)} < \infty.$$

So, each upward path requires finite number of function evaluations to be produced. The argument for finite expected running time then follows along the lines of Proposition 7. □

Algorithm 3: An Efficient Implementation of Procedure 1

```
Input: Same as Algorithm 1 and some $n \geq 1$
Output: $(S_k(\mu), M_k : 0 \leq k \leq n)$
Initialization $\mathbf{s} \longleftarrow [0]$, $\mathbf{N} \longleftarrow [0]$, $F \longleftarrow 0$ // Initialize the sample path with
    the 1-dimensional zero vector.
// The vector $N$, which is initially equals to zero records the times
    $D_j$ such that $U_j = \infty$
// $F$ is a Boolean variable which detects when we have enough
    information to compute $M_n$
Call Algorithm 2 and obtain $\omega = (s_1, ..., s_\Delta)$
Set $\mathbf{s} = [\mathbf{s}, \omega]$ // concatenate $\omega$ to s
Set $\mathbf{N} = [\mathbf{N}, \mathbf{d}(\mathbf{s}) - 1]$ // update N
while $F = 0$ do
    if $\mathbf{N}(\mathbf{d}(\mathbf{N}) - 1) \geq n$ then
        $F = 1$
    else
        Call Algorithm 2 and obtain $\omega = (s_1, ..., s_\Delta)$, and compute $M_0$
        if $M_0 \leq m$ then
            Set $\mathbf{s} = [\mathbf{s}, \mathbf{L}(\mathbf{s}) + \omega]$
            Set $\mathbf{N} = [\mathbf{N}, \mathbf{d}(\mathbf{s}) - 1]$
for $i = 0, ..., n$ do
    $M_i = \max(\mathbf{s}(i+1), \mathbf{s}(i+2), ...., \mathbf{s}(\mathbf{d}(\mathbf{s}))) - \mathbf{s}(i+1)$
    $S_i(\mu) = \mathbf{s}(i+1)$
Output $(S_k(\mu), M_k : 1 \leq k \leq n)$
```

2.5 Additional considerations: increments with infinite variance and computing truncated tilted distributions

2.5.1 The case where $E\left|X\right|^{\beta}<\infty$ for $\beta\in(1,2]$

We will now discuss how to relax the assumption that $E\left|X\right|^{\beta}<\infty$ for $\beta>2$ and assume only that $E\left|X\right|^{1+\varepsilon}<\infty$ for $\varepsilon\in(0,1]$.

The development can be easily adapted. In order to facilitate the explanation let us discuss the adaptation in the setting of Subsection A.1, which leads somewhat weaker bounds that those assumed in (2.3.6) to (2.3.7) but strong enough to adapt the conclusion in Lemma 3 through Lemma 5.

In order to adapt equation (A.1.2), for example, we now select $\delta>0$ small enough so that $1<\alpha\leq(1+\varepsilon)(1-\delta)$. Then (A.1.2) is replaced by

$$\frac{6\cdot 2^{\alpha}}{(\alpha-1)(m+1)^{\alpha-1}\mu}\cdot E\left[\left(X_{1}^{+}\right)^{1+\varepsilon}\right]\leq 1.$$

These changes yield that inequality (2.3.6), which in turn yields the proof Lemma 3 and Lemma 5.

As for Lemma 4, let us now apply Lemma 20 with

$$A(\gamma)=\left(\frac{\gamma^{2}}{2}\cdot\frac{\exp(1)}{1-\varepsilon}+2\right)\cdot E\left(\left|X\right|^{1+\varepsilon}\right),$$

and obtain

$$\exp(\psi_{k}(\theta_{k}))\leq\exp\left(\frac{A(\gamma)}{C_{k}}\right). \tag{2.5.1}$$

Since T_m we have that $S_{T_m}\geq\mu T_m+m$, and because $T_m\in[n_{k-1},n_k-1]$ we conclude that

$$S_{T_m}\geq\mu n_{k-1}+m=C_k.$$

Therefore, on $T_m\in[n_{k-1},n_k-1]$

$$\begin{aligned}\exp(-\theta_k S_{T_m}+T_m\psi_k(\theta_k))&\leq\exp(-\theta_k C_k+n_k\psi_k(\theta_k))\\&\leq\exp(-\gamma C_k^{\delta}+A(\gamma)\frac{n_k}{C_k})\\&\leq\exp(-\gamma C_k^{\delta}+A(\gamma)\frac{2}{\mu}),\end{aligned}$$

where the last inequality was obtained from the bound $\frac{n_k}{C_k} \leq \frac{n_k}{n_{k-1}\mu}$. So, we conclude, letting $z = \mu n_{k-1}$, that

$$\frac{3\exp\left(-\gamma C_k^{\delta} + 2A\left(\gamma\right)\mu^{-1}\right)}{g\left(k\right)} \leq \frac{3\left(2z+m\right)^{\alpha}}{\left(\alpha-1\right)\left(m+1\right)^{\alpha-1}z}\exp\left(-\gamma\left(m+z\right)^{\delta} + 2A\left(\gamma\right)\mu^{-1}\right).$$

Further, if $u = \gamma^{1/\delta}(m+z)$, following the development in Subsection A.1, we arrive at

$$\begin{aligned}\frac{3\exp(-\gamma C_k^{\delta} + 2A\left(\gamma\right)\mu^{-1})}{g\left(k\right)} &\leq \frac{3\cdot 2^{\alpha}\gamma^{-\alpha/\delta}}{\left(\alpha-1\right)\left(m+1\right)^{\alpha-1}\mu}\exp\left(2A\left(\gamma\right)\mu^{-1}\right)\max_{u\geq\gamma^{1/\delta}m} u^{\alpha}\exp\left(-u^{\delta}\right)\\ &\leq \frac{3\cdot 2^{\alpha}\gamma^{-\alpha/\delta}}{\left(\alpha-1\right)\left(m+1\right)^{\alpha-1}\mu}\exp\left(2A\left(\gamma\right)\mu^{-1}\right)\left(\frac{\alpha}{\delta}\right)^{\alpha}\exp\left(-\left(\frac{\alpha}{\delta}\right)^{\delta}\right).\end{aligned}$$

For every $\gamma > 0$ we can select m large enough to make the right hand side less than one and this yields the adaptation of the proof of Lemma 4 to the case $\beta \in (1,2]$.
This discussion implies that Algorithm 3 provides unbiased samples from $(M_k, S_k\left(\mu\right) : 0 \leq k \leq n)$ in finite time with probability one. Nevertheless, if $\varepsilon \in (0,1]$, we have that $\alpha \leq \left(1-\delta\right)\left(1+\varepsilon\right) < 2$ and therefore the expected number of function evaluations required to sample J in Algorithm 1 is bounded from below by

$$\sum_k n_k^2 P\left(Y > \mu n_k + m\right) = \infty.$$

Therefore, the expected running time of Algorithm 3 is not finite.

2.5.2 The issue of evaluating $\psi_k\left(\theta_k\right)$

We are concerned with the evaluation of (2.3.13), that is, during the course of the algorithm we must decide if

$$V \leq 3\cdot\exp\left\{-\theta_k S_{T_m} + T_m\psi_k\left(\theta_k\right)\right\} I(T_m \in \left[n_{k-1}, n_k - 1\right], A_k^c, B_k) \tag{2.5.2}$$

where $V \sim U\left(0,1\right)$ independent of S_{T_m} and T_m. In order to decide if inequality (2.5.2) holds one does not need to compute $\eta_k := \exp(\psi_k\left(\theta_k\right))$ explicitly. It suffices to construct a pair of monotone sequences $\{\eta_k^+\left(n\right) : n \geq 0\}$ and $\{\eta_k^-\left(n\right) : n \geq 0\}$ such that $\eta_k^+\left(n\right) \searrow \eta_k$ as $n \to \infty$ and $\eta_k^-\left(n\right) \nearrow \eta_k$ as $n \to \infty$. It is important, however, to have the sequences converging at a suitable speed. For example, it is not difficult to show that if

$$0 \leq \eta_k^+\left(n\right) - \eta_k^-\left(n\right) \leq c_0 n^{-r}$$

for $r > 2$, and the evaluation of $\eta_k^+(n)$, $\eta_k^-(n)$ takes $O(l(k)n)$ function evaluations then the expected number of function evaluations required to terminate Algorithm 1 will be bounded if $\sum_k g(k)\,l(k) < \infty$ (this holds if $E|X|^\beta < \infty$ for some $\beta > 2$ and $l(k) = O(n_k)$, given our selection of $\alpha > 2$). Note the requirement on quadratic convergence ($r > 2$). Sequences $\eta_k^+(\cdot)$ and $\eta_k^-(\cdot)$ can be constructed assuming the existence of a smooth density for X using quadrature methods. Nevertheless, we do not want to impose the existence of a smooth density and thus we shall advocate a different approach for estimating $\psi_k(\theta_k)$, based on coupling.

The approach that we advocate proceeds as follows. First, note that if X has a lattice distribution, with span $h > 0$, then $\psi_k(\theta_k)$ can be evaluated with $O\left(C_k^{1-\delta}h^{-1}\right)$ function evaluations given k. So, the expected number of function evaluations involved in implementing Algorithm 3 and deciding (2.5.2) is bounded, since $\sum g(k)\,C_k^{1-\delta} = O(\sum g(k)\,n_k) < \infty$.

Now, suppose that the distribution of X is non-lattice. The idea is to construct a coupling between $X_j(\mu)$ and a suitably defined lattice-valued random variable $X_j'(\mu')$ so that $X_j(\mu) \leq X_j'(\mu')$, $EX_j' = 0$, and $\mu' > 0$. We will simulate the random walk associated to the $X_j'(\mu')$'s, namely, $S_j'(\mu')$ and the associated sequence $\left(M_j' : j \geq 0\right)$, jointly with $(S_j(\mu) : 0 \leq j \leq n)$. Since $\max\{S_j'(\mu') : j \geq l\} \searrow -\infty$ as $l \to \infty$ we will be able to sample $(M_k : k \leq n)$ after computing N such that

$$\max\{S_j'(\mu') : j \geq N\} \leq \min\{S_k(\mu) : k \leq n\}.$$

We now proceed to describe this strategy in detail. Given $h > 0$ define

$$X_j' = h\left\lfloor \frac{X_j}{h} \right\rfloor - E\left(h\left\lfloor \frac{X_j}{h} \right\rfloor\right).$$

We omit the dependence on h in X_j' for notational convenience. In addition, let

$$\mu' = \mu - E\left(h\left\lfloor \frac{X_j}{h} \right\rfloor\right) - h.$$

Since $E\left(h\left\lfloor \frac{X_j}{h} \right\rfloor\right) < 0$ for each $h > 0$, if we also select $h \leq \mu$ we have $\mu' > 0$. Define

$$X_j'(\mu') = X_j' - \mu' = h\left\lfloor \frac{X_j}{h} \right\rfloor - \mu + h,$$

and note that

$$X_j'(\mu') \geq X_j(\mu).$$

We then define the corresponding random walks $S_n' = X_1' + ... + X_n'$, $S_n'(\mu') = S_n' - n\mu'$ with $S_0' = 0$ and

$$M_n'(\mu') = \sup\{S_k'(\mu') : k \geq n\} - S_n'(\mu').$$

The following algorithm summarizes our strategy to simulate $(S_k(\mu), M_k : 0 \leq k \leq n)$ when $\psi_k(\theta_k)$ cannot be computed exactly.

```
Algorithm 4: Strategy for simulating $(S_k(\mu), M_k : 0 \leq k \leq n)$
    Input: Same as Algorithm 1 but for $X_j'$ and $h \in (0, \mu)$
    Output: $(S_k(\mu), M_k : 1 \leq k \leq n)$
    Call Algorithm 3 and obtain $\omega' = (S_k'(\mu'), M_k' : 0 \leq k \leq n)$
    Given $\omega' = (S_k'(\mu') : 0 \leq k \leq n)$ sample $\omega = (S_k : 0 \leq k \leq n)$ ;    // this is done by sampling $X_k$ given the simulated outcome of $\lfloor X_k/h \rfloor$
    Set $M_n^- := \min(S_k(\mu) : 0 \leq k \leq n)$
    Using Algorithm 3, continue sampling $(S_k'(\mu'), M_k' : n \leq k \leq N)$, where $N = \inf\{k \geq n : M_k' + S_k'(\mu') \leq M_n^-\}$
    Given $(S_k'(\mu') : n \leq k \leq N)$, sample $(S_k : n \leq k \leq N)$
    Set $M_k = \max\{S_j(\mu) : k \leq j \leq N\} - S_k(\mu)$ for $0 \leq k \leq n$
    Output $(S_k(\mu), M_k : 0 \leq k \leq n)$.
```

The complexity analysis (i.e. finite expected running time if $E|X_1|^{2+\varepsilon} < \infty$) carries over since $EM_0' < \infty$, $E|\min\{S_k(\mu) : k \leq n\}| < \infty$, and therefore $EN < \infty$, with N defined in Algorithm 4.

2.6 Numerical Example

We will now illustrate our algorithm by revisiting the example that was described in the Introduction. This example considers the waiting time sequence that corresponds to the single-server queue. Recall that this sequence $(W_n : n \geq 0)$ can be generated by the so-called Lindley's recursion

$$W_n = (W_{n-1} + X_n - \mu)^+ \tag{2.6.1}$$

and when in steady state, the W_n's are equal in distribution to

$$M_0 = \max\{S_m(\mu) : m \geq 0\}$$

To demonstrate the capability of our algorithm, we chose a sequence of X_n's of the form

$$X_n = h\left\lfloor \frac{c}{h} V_n \right\rfloor - E\left(h\left\lfloor \frac{c}{h} V_n \right\rfloor\right) =: Y_n - E(Y_n) \tag{2.6.2}$$

where $V_n \sim Pareto(\alpha')$, that is,

$$P(V > t) = \frac{1}{(1+t)^{\alpha'}} \qquad t > 0$$

The parameters α', c, and h can be changed in order to test the algorithm in different scenarios. $\alpha' > 2$ determines how heavy the tail of the increments is, $h > 0$ is the lattice parameter (the non-lattice case is where $h \to 0$), and $c > 0$ controls the mean of Y_n.

2.6.1 Choice of Parameters

As mentioned at the end of Subsection A.1, we used the Excel solver in the following way: given our selection of $\alpha \in (2, 4)$, we picked $\delta \in (0, 1/2]$, $\gamma \geq 0$, and $m \geq 0$ so as to minimize m subject to (2.3.6) and (2.3.7). The input parameters μ, α', h, and c are chosen to test conditions ranging from light to heavy traffic (controlled primarily by the parameter μ), and from heavy tails to relatively lighter tails (which are controlled by the parameter α').

We conclude our discussion by providing a formal comparison against the relaxation time of the Markov chain $\{W_n : n \geq 0\}$ in heavy-traffic. We chose a formal comparison because a rigorous computation of the exact relaxation time of the single-server queue is not available (to the best of our knowledge) at the level of generality at which our algorithm works, although bounds have been studied, as is the case in [Foss and Sapozhnikov, 2006]. We have argued that our algorithm is sharp, in the sense that it is applicable under close to minimal conditions required for the stability of the single-server queue. We believe that the heavy-traffic analysis provides yet another interesting perspective.

Assuming that $\beta > 2$ (i.e. the increments have finite variance), in heavy traffic, as $\mu \to 0$, it is well known that at temporal scales of order $O\left(\mu^{-2}\right)$ and spatial scales of order $O\left(\mu^{-1}\right)$ Lindley's recursion can be approximated by a one dimensional reflected Brownian motion (RBM). In fact, the approximation persists also for the corresponding stationary distribution (which converges after proper normalization to an exponential distribution, which is the stationary distribution of RBM (see [Kingman and Atiyah, 1961], for example)). The relaxation time of $\{W_n : n \geq 0\}$ is of order $O\left(\mu^{-2}\right)$ as $\mu \to 0$.

The running time analysis of our algorithm involves the "downward" patches, which take $O(m)$ random numbers to be produced. We also need to account for the simulation of the Bernoulli trials for each "upward" patch, which requires the generation of K under $g(\cdot)$, and a total of $C_0 = O\left(\sum_{k=1}^{\infty} n_k g(k)\right)$ expected random numbers to be simulated. This analysis holds because the number of proposals required to sample $P_{k,0}$, $P_{k,1}$ and $P_{k,2}$ remains bounded also as $\mu \to 0$. Therefore, the actual X_i's conditional on the E_i's can be easily simulated. A similar strategy can be implemented for $P_{k,2}$.

Consequently, the over all cost of our algorithm is driven by $C_0 = O\left(\mu^{-2} m\right)$. We also need to ensure that m is selected so that (2.3.6) and (2.3.7) are satisfied. From the analysis of (A.1.2) and (A.1.4), we see that $m = O\left(\mu^{-1}\right)$ is always a possible choice. However, this choice can be improved if one can select a large α , which in turn is feasible as long as $z^{\alpha} P\left(X > z^{1-\delta}\right) = O(1)$. In particular, we can choose $m = O\left(\mu^{-1/(\alpha-1)}\right)$, provided that δ is chosen sufficiently close to unity in order to satisfy (A.1.4). Our exact sampling algorithm in heavy traffic has a running time that is not worse that $O\left(\mu^{-3}\right)$ and it can be arbitrarily close to the relaxation time $O\left(\mu^{-2}\right)$ of the chain $\{W_n : n \geq 0\}$.

2.6.2 Simulation Results

We tested the algorithm in four different cases in which we changed the nature of the random walk increments and the traffic intensity. By picking $\alpha' = 2.9$ and $\alpha' = 7$, we considered heavy tailed increments and relatively lighter tailed increments, respectively. By changing the value of c, we changed the traffic intensity ρ, which is given by

$$\rho = \frac{E\left(h\left\lfloor \frac{c}{h} V \right\rfloor\right)}{E\left(h\left\lfloor \frac{c}{h} V \right\rfloor\right) + \mu} \approx \frac{cE(V)}{cE(V) + \mu}$$

Throughout all scenarios we used the parameters

$$L = 1.1, \quad h = 0.1, \quad \mu = 1 \quad and \quad \delta = 0.38$$

The rest of the parameters were chosen as follows:

	$\rho = 0.3$				$\rho = 0.8$			
	α	γ	c	m	α	γ	c	m
$\alpha' = 7$	4	1.7	3	16	4	0.75	25	217
$\alpha' = 2.9$	2.01	1.24	0.85	35	2.01	0.74	8	400

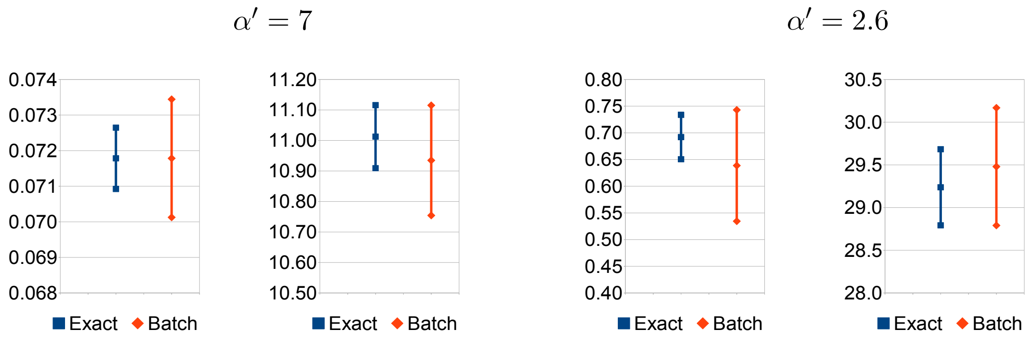
Table 2.1: Parameters Choice for Simulation

In each of the above cases we generated 100,000 exact replicas of M_0 and compared it with the chain $\{W_n : 0 \leq n \leq l\}$, where l was picked to fit the scenario. To analyze the output of the chain, we used batches with varying sizes. In the light traffic case, for both $\alpha' = 2.9$ and $\alpha' = 7$, we used $l = 10^6$ with batches of size 25. In the heavy traffic scenario, we used $l = 2 \cdot 10^6$ with batches of size 50 for $\alpha' = 7$, and $l = 4 \cdot 10^6$ with batches of size 100 for $\alpha' = 2.9$

We summarized the result in the following table (see also Figure 2.6.1):

		$\rho = 0.3$			$\rho = 0.8$		
		LCI	UCI	RT	LCI	UCI	RT
$\alpha' = 7$	Exact sampler	0.0709	0.0726	≈ 1.5	10.9092	11.1159	≈ 10
	Batch mean	0.0701	0.0734	≈ 1	10.7542	11.1152	≈ 3
$\alpha' = 2.9$	Exact sampler	0.6505	0.7336	≈ 3	28.7925	29.6832	≈ 15
	Batch mean	0.5344	0.7429	≈ 1	28.7908	30.1681	≈ 4

Table 2.2: Simulation Results. LCI/UCI=Lower/Upper 95% Confidence Interval. RT= Running Time (in minutes).



$\alpha' = 7$ $\alpha' = 2.6$

$\rho = 0.3$ $\rho = 0.8$ $\rho = 0.3$ $\rho = 0.8$

Figure 2.6.1: Exact sampler mean $E(M_0)$ VS. batches mean of $\{W_n : 0 \leq n \leq l\}$, along with the corresponding 95% confidence intervals.

In the numerical examples we see that the IID replications of M_0 appear to be a reasonable approach to steady-state estimation, especially in light traffic. The performance deteriorates somewhat in heavy traffic, which is expected given our earlier discussion on running time in heavy traffic. Nevertheless, it is important to note that while our procedure does not have any bias, batch means do not provide control on the bias with absolute certainty. Overall, we feel that a few minutes of additional running time in exchange for total bias deletion is not an onerous price to pay. Therefore, our procedure is not only of theoretical interest (as the first exact sampler for a general single-server queue), but of practical value as well.

2.7 Conclusions

The work presented in this chapter was motivated by the important role that single-server queue plays in many applications that use the DCFTP method as well as the challenge of efficiently dealing with random walks involving heavy-tailed increments. We developed an exact simulation method that can be used to simulate the stationary waiting-time sequence of a single-server queue backward in time, jointly with the input process of the queue. We provided an algorithm, which is easy to implement, that has a finite expected termination time under nearly minimal assumptions.

Part II

Diffusion Processes in Random Environment

Introduction to Part II

The work that we will present in this part is motivated by the problem of finding an optimal strategy for detecting an adversarial agent. We will describe the framework of our modeling and in what sense we define optimality, when evaluating the efficiency of a policy of detectors deployment.

We will model the dynamic of a rouge target and define the notion of optimality among the admissible detectors allocation policies. In addition, we will also provide a large deviation result that will enable us to evaluate policies and solve the problem in special cases. We will start with some notations that will help give the problem a mathematical framework.

Consider an adversary that moves (continuously) in a certain domain $G \subset \mathbb{R}^d$, $d \geq 1$, and its movement can modeled by a diffusion process $(X(t) : t \geq 0)$. We assume that the domain G represents a containment region, in the sense that if the adversary reaches the boundary of G, then he is immediately detected and neutralized. This assumption can be relaxed, as we shall explain later. In order to catch the agent, we have a number of detectors (traps) at our disposal, that we can place in the domain G. We assume that the detection coverage area of each trap can be modeled as a ball of radius a. Moreover, we assume that after the detection units have being placed their locations stay fixed. Our goal is to catch the adversary before a given deadline T. A natural deployment strategy to consider consists with selecting a set of points $\{p_i\}_{i=1}^{K} \subset G$ that represents the locations where the detection units should be placed.

Let $B(x, r)$ denote the $d-$dimensional Euclidean ball of radius $r > 0$ that centered at x. We define the Wiener sausage of radius $r > 0$ at time $t > 0$ that corresponds to the process $(X(t) : t \geq 0)$ by

$$W_t^r = \bigcup_{0 \leq s \leq t} B(X(s), r). \tag{2.7.1}$$

Notice that W_t^r is a set-valued process and that $W_s^r \subseteq W_t^r$ for $s \leq t$. Moreover, observe that the event that the agent has not been detected by time T is equivalent to

$$\left\{ \left(\bigcup_{0 \leq s \leq T} X(s) \right) \cap \left(\bigcup_{i=1}^{K} B(p_i, a) \right) = \emptyset \right\} = \left\{ W_T^a \cap \left(\bigcup_{i=1}^{K} p_i \right) = \emptyset \right\}. \tag{2.7.2}$$

Therefore, our goal is to find an allocation policy the minimizes the probability

$$P\left(W_T^a \cap \left(\bigcup_{i=1}^{K} p_i \right) = \emptyset \right).$$

A natural approach to solve this problem might be the following. Consider dividing the domain G into a grid made of squares with edges of size $\epsilon = \epsilon(d, a)$. ϵ is chosen such that the $d-$dimensional cube of volume ϵ can be inscribed within a ball of radius a. Then a deployment strategy would be a selection of a subset of size K of these squares and placement of K detectors at their centers. The detection coverage of such strategy can be roughly represented by the union of the selected K squares (see Figure 2.7.1). Under this formulation, the problem would consists of finding the optimal configurations which minimizes the probability of no detection by time T. Unfortunately, this is a hard combinatorial problem. This happens because one needs to find the best subset of size K out of a set of items whose cardinality is of order $O\left(c^d\right)$ for some $c > 1$, and then evaluate the probability of no detection for some general dynamics of an adversary. Furthermore, the problem of finding optimal trap allocation brings to mind the well-known Maximum Coverage Problem. In the Maximum Coverage Problem framework we are given a collection of subsets $\mathcal{S} = \{S_1, \ldots, S_n\}$ and tasked with finding a subset S^* which consists with at most k subsets from the collection, such that together they contain a maximal amount of elements,that is, the size $\left| \bigcup_{S_i \subset S^*} S_i \right|$ is maximal. It can be formulated as an integer programming problem and it is known to be NP hard (for more information see e.g., [Nemhauser *et al.*, 1978], [Hochbaum, 1997], and [Vazirani, 2001]).

Due to the presence of this combinatorial difficulty, it is natural to consider a continuous relaxation of the problem. We shall study one such relaxation by introducing a randomized class of policies. Instead of using a deterministic deployment policy, we will randomly place the traps according to a Poisson point process with an intensity $\beta = \left(\beta\left(x\right) : x \in \mathbb{R}^d\right)$, where $\beta\left(x\right)$ is the evaluation of β at a point x. We will refer to the resulting configuration as a Poisson cloud. The randomized problem then optimizes over some family of admissible intensities. By using this technique we might hope achieve tractability.

The problem of diffusion process moving in a random environment is a well known problem in statistical physics. The way the traps are randomly placed determines the kind of "disorder" present in the system. Quenched disordered system is a system where some of the parameters that define the system are random variables that do not evolve with time, i.e. they are quenched or frozen (spin glasses are typical examples). The counterpart of the quenched disorder is the annealed disorder. In this kind of system the random variables that define the system evolve with time and their evolution is related to the degrees of freedom that define the system.

When analyzing diffusion processed in a quenched systems, we typically consider a measure that describes the behavior of the trajectory of a particle diffusing from a known point (say the origin) in a typical realization of the quenched environment. In the annealed setup, we can use a spatial ergodic theory to handle the dynamic environment in which the diffusion process evolves.

In the mathematical community, the topic has long been a subject of research as well. In their paper [Donsker and Varadhan, 1975] considered a Brownian motion in $\mathbb{R}^d$ moving in a Poisson cloud of constant intensity $\nu > 0$. They derived a large deviation result regarding the volume of the corresponding Wiener sausage conditioned on the event that it doesn't include any point of the Poisson cloud. Later on, under the same assumptions,

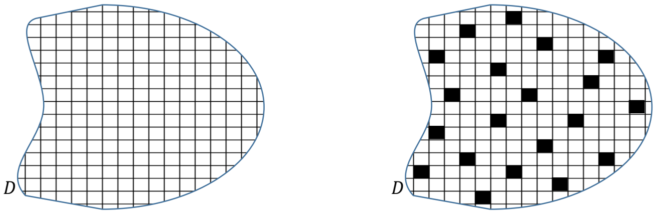


Figure 2.7.1: Deterministic deployment strategy

[Sznitman, 1991] (for $d = 2$) and [Povel, 1999] (for $d \geq 3$) showed that the probability that the Wiener sausage is confined in a ball of radius $c\, t^{1/(d+2)}$, for some (computable) $c > 0$, given that the sausage does not include any point of the Poisson cloud, tends to 1 as t tends to ∞. A comprehensive account of ideas, results and techniques, which relate to the study of Brownian motion and random obstacles, can be found, for example, in [Sznitman, 1998].

In a slightly different but related framework, [Peres *et al.*, 2011] considered a scenario both the agent and the points of the Poisson clouds are moving. Each point moves according to a Brownian motion independent of the target process and the other points of the cloud. Among other results, the authors derived a large deviation result for the probability of target detection in the case.

Problems related to diffusions avoiding obstacles borrow mathematical tools from analysis and probability. As we shall discuss, spectral properties associated to the underlying diffusion plays an important role (see [Kac, 1951]). Later, we will invoke few results from [Pinsky, 1985] and [Pinsky, 1995], which lies within the intersection of probability and partial differential equations (PDEs).

Note that he above list does not pretend to be comprehensive overview of the work that has been done in the field. There are many more in-progress and published results, papers, and books in this area that were not mentioned here.

At this point, we would like to introduce the mathematical framework and some related results that we will need later.
We will consider diffusion processes that evolve according to stochastic differential equations (SDE) of the form

$$dX\left(t\right) = \nabla Q\left(X\left(t\right)\right) dt + dB\left(t\right), \tag{2.7.3}$$

where $\left(B\left(t\right) : t \geq 0\right)$ is a standard Brownian motion in $\mathbb{R}^d$ and $Q : \mathbb{R}^d \to \mathbb{R}$ is a C^2.
The generator associated with this SDE is a differential operator L of the form

$$L = -\frac{1}{2}\triangle + \nabla Q \cdot \nabla = -\frac{1}{2}\sum_{i=1}^{d} \frac{\partial^2}{\partial x_i^2} + \sum_{i=1}^{d} Q_{x_i}\left(x\right)\frac{\partial}{\partial x_i}. \tag{2.7.4}$$

L is a self-adjoint operator over $\mathcal{L}^2$ with density $\exp(2Q)$, which makes spectral theory one of the tools that we have at our disposal. In particular we will be utilizing the Dirichlet principle eigenvalue, which is defined by

$$\lambda(G) = \text{bottom of the spectrum of } L \text{ with Dirichlet boundary condition on } G \subset \mathbb{R}^d. \tag{2.7.5}$$

Here are couple useful properties of the principle eigenvalue.

Property 1. *Let $G_1, G_2 \subset \mathbb{R}^d$ be non-empty and open sets. Then,*

$$G_1 \subset G_2 \Rightarrow \lambda(G_1) \geq \lambda(G_2). \tag{2.7.6}$$

Property 2. *Let $G = \bigcup_i G_i$ where G_i are the connected components of G. Then,*

$$\lambda(G) = \inf_i \lambda(G_i). \tag{2.7.7}$$

The principle eigenvalues plays a key role when deriving probabilistic quantities related to the diffusion process $(X(t) : t \geq 0)$. Here is one example,

Theorem 9. *Let $G \subset \mathbb{R}^d$ and let τ_G denote the first time the process $(X(t) : t \geq 0)$ exits G. Namely,*

$$\tau_G = \inf\{t \geq 0 : X(t) \notin G\}. \tag{2.7.8}$$

Then for every $x \in G$ we have

$$P_x(\tau_G > t) = e^{-\lambda(G)t + o(t)} \tag{2.7.9}$$

as $t \to \infty$, where $P_x(\cdot) = P(\cdot | X(0) = x)$ (see e.g., [Pinsky, 1985]).

Part II is organized in the following way. In Chapter 3 we will formulate the problem in detail and show that the stochastic relaxation is indeed a good approximation to the deterministic problem. We will also discuss about different frameworks for which our method can be applied to and the challenges of solving the optimization problem analytically. Chapter 4 will be defecated to presenting our main results and efficient Monte Carlo algorithms, that can numerically evaluate the performance of a given deployment policy and sample the adversary's location conditioned on him avoiding detection in an asymptotically optimal manner (as the deadline increases).

Chapter 3

Detection of Adversarial Agents: Modeling and Problem Formulation

Introduction

As mentioned earlier, finding an optimal deployment policy is combinatorially difficult. Therefore, it is natural to study continuous relaxations via randomized policies. The main goal of this chapter is to rigorously describe the framework we used to model problem of detecting an adversary, and study the nature of the types of randomized policies that can be used as a relaxation.

We will start by arguing in Section 3.1 that our method is reasonable in the sense that the randomized relaxation asymptotically approximates the optimal deployment policy as $T \to \infty$. We will show that in the particular case where the agent move according to a one-dimensional Brownian motion, which is of course limited, but tractable. We will dedicate Section 3.2 for a discussion on the advantages of taking the stochastic approach and the differences between the one-dimensional and the multi-dimensional scenario. Then in Section 3.3, we will provide a detailed construction of the model for the general framework we consider, namely for a broader family of diffusion processes and for any $d \geq 1$. In Section 3.4 we will explore the optimal solution to the optimization problem in the case of a two-dimensional Brownian motion and will discuss the possible frameworks that one can consider when implementing the stochastic approach.

3.1 One-Dimensional Brownian Case

Throughout this section we will consider an adversary that moves on the real line according to a Brownian motion $(B(t) : t \geq 0)$. He starts at a random point x_0, where x_0 is distributed according to a probability density function $f(\cdot)$ of the form

$$f(x) = \exp(-h(x)) \quad \text{and} \quad \operatorname{supp} f \subset (-m, m) \tag{3.1.1}$$

for some $m > 0$. Let $V_T(x_0)$ denote the range covered by the Brownian motion started at x_0 by time T. Namely,

$$V_T(x_0) = \bigcup_{s \leq T} \{B(s) \mid B(0) = x_0\}. \tag{3.1.2}$$

We assume that we have a containment mechanism in place, which restricts the motion of the target to the region $(-m, m)$. Such mechanism can be achieved if we place two detectors at positions $\{-m, m\}$, for example. In addition to these two detectors, we assume that we have K detectors that are available at our disposal.

We will start by deriving the optimal deterministic policy in Subsection 3.1.1. Next, in Subsection 3.1.2, we will introduce a continuous randomized relaxation to the problem and show that its optimal solution tends to the deterministic one as $T \to \infty$.

3.1.1 Deterministic Case

Under the assumption that there are K detection units that we can place in the the interval $(-m, m)$, we can describe the family of admissible strategies by

$$\begin{aligned} \Theta_K &= \left\{ \theta(x) = \sum_{i=1}^{k} I_{\{z_i\}}(x) : z_1, \ldots, z_k \in (-m, m), k \leq K \right\} \\ &= \{\theta = \{z_1, \ldots, z_k\} : z_1, \ldots, z_k \in (-m, m), k \leq K\}, \end{aligned} \tag{3.1.3}$$

where the z_i's represent the locations of the detectors placed according to a strategy θ. For every $\theta \in \Theta_K$ let σ_θ denote the time we detect and neutralize the agent if we follow policy θ. That is,

$$\sigma_\theta = \inf \left\{ t \geq 0 : V_t(x_0) \cap \left(\bigcap_{z_i \in \theta} \{z_i\} \right) \neq \emptyset \right\}. \tag{3.1.4}$$

Therefore, under any given deadline $T > 0$, the (deterministic) optimization problem can be formulated by

$$\min_{\theta\in\Theta_K} \bar{P}\left(\sigma_\theta > T\right). \tag{3.1.5}$$

Her we use the notation $\bar{P}$ to emphasize that the source of randomness in this case arises form the underlying Brownian dynamic of the target and its initial position.

Intuitively, the optimal policy, θ^*, would be the strategy that places the K detectors in equal spaces of size $\frac{2m}{K+1}$, namely,

$$\theta^*\left(x\right) = \sum_{i=1}^{K} I_{\{y_i\}}\left(x\right) \qquad \text{with } y_i = -m + \frac{i}{K+1}\cdot 2m \tag{3.1.6}$$

(see Figure 3.1.1). The following lemma proves that indeed this strategy is optimal.

Lemma 10. *Let θ^* be as defined in equation* (3.1.6)*. Then,*

$$\lim_{T\to\infty}\frac{1}{T}\log\bar{P}\left(\sigma_{\theta^*} > T\right) = -\pi^2\left(\frac{K+1}{2m}\right)^2. \tag{3.1.7}$$

Moreover,

$$\bar{P}\left(\sigma_{\theta^*} > T\right) = \min_{\theta\in\Theta_K}\bar{P}\left(\sigma_\theta > T\right) \tag{3.1.8}$$

as $T\to\infty$.

Proof. Recall that the generator of the one-dimensional Brownian motion is $L = -\frac{1}{2}\frac{d^2}{ds^2}$ and the corresponding principle Dirichlet eigenvalue on any interval (a,b) is given by

$$\lambda\left((a,b)\right) = \frac{\pi^2}{\left(b-a\right)^2}. \tag{3.1.9}$$

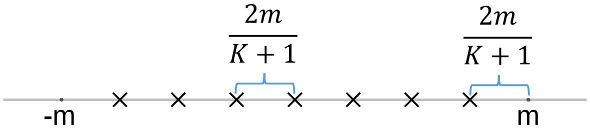


Figure 3.1.1: Optimal Policy for one-dimensional Brownian motion confined to $(-m,m)$.

Notice that

$$\{\sigma_{\theta^*} > T\} \iff \{\forall x \in (-m,m) : \exists i \in \{0,1,\dots K\}\ V_T(x) \subset (y_i, y_{i+1})\}, \tag{3.1.10}$$

where $y_0 := -m$ and $y_{K+1} := m$. Moreover, by construction we have that for every $i = 0, 1, \dots, K$

$$\lambda\left((y_i, y_{i+1})\right) = \pi^2 \left(\frac{K+1}{2m}\right)^2. \tag{3.1.11}$$

Invoking Property 2 of the principle eigenvalue and Theorem 9 we obtain that

$$P\left(\sigma_{\theta^*} > T\right) = \exp\left(-\pi^2 \left(\frac{K+1}{2m}\right)^2 T + o\left(T\right)\right), \tag{3.1.12}$$

and equation (3.1.7) follows. To show optimality, let $\theta = \{z_1, \dots, z_K\} \in \Theta_K \setminus \theta^*$ be some policy that deploys exactly K detectors in $(-m, m)$. If we adopt such strategy, then the maximal distance between two neighboring locations, say l, will be strictly greater than $\frac{2m}{K+1}$. Invoking Property 2 and Theorem 9 again, we will obtain that

$$P\left(\sigma_{\theta} > T\right) = \exp\left(-\pi^2 \left(\frac{1}{l}\right)^2 T + o\left(T\right)\right). \tag{3.1.13}$$

Therefore,

$$P\left(\sigma_{\theta} > T\right) > P\left(\sigma_{\theta^*} > T\right) \tag{3.1.14}$$

as $T \to \infty$. From similar arguments, for every $\theta = \{z_1, \dots, z_k\} \in \Theta_K$, where $k < K$, equation (3.1.14) holds, and the result follows. □

3.1.2 The Stochastic Approach

We will now provide a description of the randomized continuous relaxation to the optimization problem in the one-dimensional case. Moreover, we will show that its solution approximate the optimal deterministic policy as $T \to \infty$.

As before, our goal is to minimize the probability that an adversary avoids detection by a given deadline T over the set of admissible strategies. However, this time we want to utilize a family of strategies of stochastic nature. Specifically, we consider policies that deploy the traps at the beginning of the time horizon according to a Poisson spatial process with intensity $\beta_T(s) = T\beta(s)$, subject to the constraints that

$$\int_{-m}^{m} \beta(s)\, ds = K, \tag{3.1.15}$$

and that two detectors are initially deployed on $\{-m, m\}$. Let us denote by $\mathcal{B}$, the set of such strategies, namely,

$$\mathcal{B} = \left\{ \beta : (-m, m) : \mathbb{R} : \ \beta \geq 0, \int_{-m}^{m} \beta(s)\, ds = K \right\}. \tag{3.1.16}$$

In the following we will discuss about how to implement this continuous relaxation and why this particular design is desirable. First we will argue that as $T \to \infty$, this continuous relaxation returns the same optimal policy as the deterministic strategy discussed in the previous section. Moreover, we will see that the optimal randomized strategy consists with picking $\beta(\cdot)$ as a mixture of suitably centered delta functions (point mass measures). We can think of a delta function centered at a point p, as a density of a Gaussian random variable with mean equal to p and an infinitesimally small variance. This intuition will be made rigorous in this section.

When considering other models for the adversary's dynamic, for example a Brownian motion with some general drift coefficient, the idea is to solve for an optimal allocation by finding K centers of K Gaussian densities, each with a relatively small variance, which in turn may be a parameter to optimize.

In the following subsections we will show that

$$\lim_{T\to\infty} \frac{1}{T} \inf_{\beta\in\mathcal{B}} \log P(\sigma_R > T) = -\pi^2 \left(\frac{K+1}{2m}\right)^2.$$

Here we use the notation, P, to emphasize that in addition to the Brownian dynamics of the target and its initial position (which is also random), there is the randomness associated to the deployment policy.

3.1.2.1 Lower Bound

In this section we will establish that the deterministic solution provides a lower bound to the one of the stochastic relaxation, as stated in Lemma 11.

Lemma 11. *Consider a one-dimensional Brownian motion that is confined to an interval* $(-m, m)$. *Let* $\mathcal{B}$ *be the family of strategies defined in equation* (3.1.16). *Then,*

$$\lim_{T\to\infty} \frac{1}{T} \inf_{\beta\in\mathcal{B}} \log P\left(\sigma_R > T\right) \geq -\pi^2 \left(\frac{K+1}{2m}\right)^2 . \tag{3.1.17}$$

Lemma 11 asserts that any admissible randomized policy cannot perform better than the optimal deterministic strategy. A key component of proving the lemma consists with the following result.

Lemma 12. *In the setting on Lemma 11, we have*

$$\sup_{\beta\in\mathcal{B}} \inf_{\substack{G \text{ open} \\ G\subset(-m,m)}} \left(\int\limits_{G} \beta\left(s\right) ds + \lambda\left(G\right) \right) = \pi^2 \left(\frac{K+1}{2m}\right)^2 . \tag{3.1.18}$$

Proof. To make the proof easier to follow, we will first lay out its outline, and then describe in detail how the execute each step. We will start by defining a family of admissible policies $(\beta_\epsilon : \epsilon > 0) \subset \mathcal{B}$. For a fixed $\epsilon > 0$ we will construct a set G_ϵ for which

$$f\left(G_\epsilon\right) := \int\limits_{G_\epsilon} \beta_\epsilon\left(s\right) ds + \lambda\left(G\right) > \pi^2 \left(\frac{K+1}{2m}\right)^2 - \epsilon. \tag{3.1.19}$$

Then, we will show that for every G that is contained in G_ϵ,

$$f\left(G\right) \geq f\left(G_\epsilon\right).$$

Next, we will show that for every G that contains G_ϵ and such that,

$$\left| G \cap \bigcup_{i=1}^{K} \left(y_i - \frac{\delta}{2}, y_i + \frac{\delta}{2} \right) \right| = \eta,$$

for some $\eta > 0$,

$$f\left(G\right) \geq f\left(G_\epsilon\right).$$

Finally, we will establish that there exists $\epsilon_0 > 0$ such that for every $\epsilon \leq \epsilon_0$

$$f\left(G_\epsilon\right) = \inf_{\substack{G \text{ open} \\ G\subset(-m,m)}} \left(\int\limits_{G} \beta_\epsilon\left(s\right) ds + \lambda\left(G\right) \right) > \pi^2 \left(\frac{K+1}{2m}\right)^2 - \epsilon,$$

which will conclude the proof.

We will start by constructing the family of policies $(\beta_\epsilon : \epsilon > 0)$. For every $\epsilon > 0$ we consider $\delta(\epsilon) = \delta > 0$ which will be specified later. We define β_ϵ by

$$\beta_\epsilon(s) = \frac{1}{\delta}\sum_{i=1}^{K} I\left(s \in \left(y_i - \frac{\delta}{2}, y_i + \frac{\delta}{2}\right)\right), \tag{3.1.20}$$

where y_i, $i = 1, \ldots, K$ are as in equation (3.1.6).

For every ϵ we define a function $f_\epsilon : \{G \subset (-m, m), G \text{ open}\} \to \mathbb{R}$ by

$$f_\epsilon(G) = \int_G \beta_\epsilon(s)\, ds + \lambda(G). \tag{3.1.21}$$

Define a set

$$G_\epsilon = \left(y_i + \frac{\delta}{2}, y_{i+1} - \frac{\delta}{2}\right), \tag{3.1.22}$$

for some $i = 1, \ldots, K-1$ (see Figure 3.1.2). Notice that Property 2 of the principle eigenvalue implies that the choice of i or some union of different $i's \in \{0, 1, 2, \ldots, K\}$ does not affect the rest of the proof.

Therefore,

$$f_\epsilon(G_\epsilon) = \lambda(G_\epsilon) = \pi^2\left(\frac{2m}{K+1} - \delta\right)^2 = \pi^2\left(\frac{K+1}{2m}\right)^2 + \epsilon, \tag{3.1.23}$$

for $\delta = \delta(\epsilon)$ such that the equality

$$\epsilon = \pi^2\delta\left(\frac{K+1}{2m}\right)\left(\frac{2m}{K+1} - \delta\right)^{-1}\left(2 + \delta\left(\frac{K+1}{2m}\right)\left(\frac{2m}{K+1} - \delta\right)^{-1}\right) \tag{3.1.24}$$

is satisfied. Notice that $\epsilon \to 0 \iff \delta \to 0$.

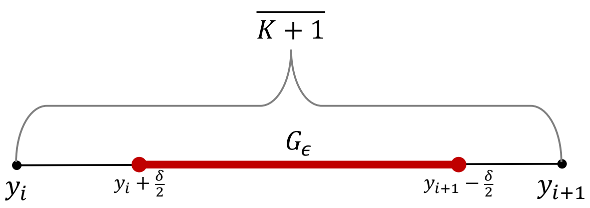


Figure 3.1.2: G_ϵ

Now consider $G \subset G_\epsilon$. Notice that $\int\limits_G \beta_\epsilon(s)\,ds = 0$ and therefore we have $f_\epsilon(G) = \lambda(G)$. Invoking Property 1 of the principle eigenvalue, we have that

$$f_\epsilon(G) \geq f_\epsilon(G_\epsilon) \qquad \forall G \subset G_\epsilon. \tag{3.1.25}$$

Next, let G be such that,

$$G_\epsilon \subset G \quad \text{and} \quad \left| G \cap \left(\bigcup_{i=1}^{k} \left(y_i - \frac{\delta}{2}, y_i + \frac{\delta}{2} \right) \right) \right| = \eta > 0. \tag{3.1.26}$$

Among all such sets, f_ϵ is minimal when $G' = G_\epsilon \cup \left(y_i + \frac{\delta}{2} - \eta, y_i + \frac{\delta}{2} \right)$ (see Figure 3.1.3). Indeed, for every G satisfying equation (3.1.26),

$$f_\epsilon(G) = \frac{\eta}{\delta} + \lambda(G). \tag{3.1.27}$$

Recalling Property 2 and Property 1 of the principle eigenvalue, we obtain that the minimum of f_ϵ over sets that satisfy equation (3.1.26) is achieved when the set is connected. An example for such set is G'.

$f_\epsilon(G')$, takes the form

$$f_\epsilon(G') = \frac{\eta}{\delta} + \pi^2 \left(\frac{2m}{K+1} - \delta + \eta \right)^{-2}. \tag{3.1.28}$$

Notice that as δ approaches 0 the first term tends to ∞ while the second term converges to $\pi^2 \left(\frac{2m}{K+1} + \eta \right)^{-2}$, which is bounded from above by $\pi^2 \left(\frac{2m}{K+1} \right)^{-2}$.

Therefore, there exists ϵ_0 such that for all $\epsilon \leq \epsilon_0$, we have

$$f_\epsilon(G) \geq f_\epsilon(G_\epsilon) \tag{3.1.29}$$

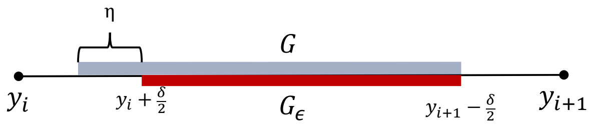


Figure 3.1.3: G_ϵ

for all sets G that satisfy equation (3.1.26).

In conclusion, we have that,

$$f_\epsilon(G_\epsilon) = \inf_{\substack{G \text{ open} \\ G\subset(-m,m)}} \left(\int_G \beta_\epsilon(s)\, ds + \lambda(G) \right) > \pi^2 \left(\frac{K+1}{2m}\right)^2 - \epsilon \tag{3.1.30}$$

for every $\epsilon \leq \epsilon_0$ and the result of the lemma follows.

□

We will close this section with the proof of Lemma 11.

Proof of Lemma 11. By conditioning on the initial position we have,

$$\begin{aligned}
\inf_{\beta\in\mathcal{B}} P(\sigma_R > T) &= \inf_{\beta\in\mathcal{B}} E\left(P_x(\sigma_R > T)\right) && (3.1.31)\\
&= \inf_{\beta\in\mathcal{B}} E\left(P_x(\sigma_R > T,\, V_T(x) \subset (-m,m))\right) && (3.1.32)\\
&= \inf_{\beta\in\mathcal{B}} E\left(\exp\left\{-\int_{V_T(x)} \beta(s)\, ds\right\} I\left(V_T(x) \subset (-m,m)\right)\right). && (3.1.33)
\end{aligned}$$

Now, applying Theorem 9 we can bound the expectation on the right hand side form below and obtain that,

$$\begin{aligned}
\inf_{\beta\in\mathcal{B}} P(\sigma_R > T) &\geq \inf_{\beta\in\mathcal{B}} \exp\left\{- \inf_{\substack{G \\ G\subset(-m,m)}} \int_G \left(\beta(s)\, ds + T\lambda(G) + o(T)\right)\right\} && (3.1.34)\\
&= \exp\left\{- \sup_{\beta\in\mathcal{B}} \inf_{\substack{G \\ G\subset(-m,m)}} \int_G \left(\beta(s)\, ds + T\lambda(G) + o(T)\right)\right\}. && (3.1.35)
\end{aligned}$$

Invoking Lemma 12 we have

$$\inf_{\beta\in\mathcal{B}} P(\sigma_R > T) \geq \exp\left\{-\pi^2 \left(\frac{K+1}{2m}\right)^2 + o(T)\right\} \tag{3.1.36}$$

and the result follows. □

3.1.2.2 Upper Bound

In this section we will construct a family of randomized placement policies that approaches optimality as $T \to \infty$. To this end, we will utilized the collection of randomized strategies $(\beta_\epsilon : \epsilon > 0)$, that where constructed in the previous section. However, this time we will specify the sequence parameters $(\epsilon = \epsilon(T) : T > 0)$ (or rather $\delta(\epsilon)$) for which $(\beta_\epsilon : \epsilon > 0)$ are asymptotically optimal.

Lemma 13. *Consider a one-dimensional Brownian motion that is confined to an interval* $(-m, m)$*. Let* $\mathcal{B}$ *be the family of strategies defined in equation* (3.1.16)*. Then,*

$$\lim_{T\to\infty} \frac{1}{T} \inf_{\beta\in\mathcal{B}} \log P(\sigma_R > T) \le -\pi^2 \left(\frac{K+1}{2m}\right)^2 . \tag{3.1.37}$$

Proof. We will start by revisiting the family of strategies defined in equation (3.1.20). For $T > 0$ we pick $\delta = \frac{1}{T}$ and define $\beta_T^*(s)$ by

$$\beta_T^*(s) = T\beta_{\delta=\frac{1}{T}}(s) = T^2 \sum_{i=1}^{K} I\left(s \in \left(y_i - \frac{1}{2T}, y_i + \frac{1}{2T}\right)\right) = T \cdot \sum_{i=1}^{K} h_T(s, y_i), \tag{3.1.38}$$

where

$$h_T(s, y_i) = TI\left(s \in \left(y_i - \frac{1}{2T}, y_i + \frac{1}{2T}\right)\right). \tag{3.1.39}$$

Notice that, for every $G \subset \mathbb{R}$ we have

$$\begin{aligned} \sum_{i=1}^{K} \int_G h(s, y_i)\, ds &= T \sum_{i=1}^{K} \left| G \cap \left(y_i - \tfrac{1}{2T}, y_i + \tfrac{1}{2T}\right) \right| \\ &\xrightarrow[T\to\infty]{} T \sum_{i=1}^{K} \tfrac{1}{T} I(y_i \in G) = \#\left\{G \cap \{y_i,\, i = 1, \dots, K\}\right\}. \end{aligned} \tag{3.1.40}$$

Overall, we have

$$\inf_{\beta\in\mathcal{B}} P(\sigma_R > T) = \tag{3.1.41}$$

$$= \inf_{\beta\in\mathcal{B}} E\left(\exp\left\{-\int_{V_T(x)} \beta(s)\, ds\right\} I(V_T(x) \subset (-m, m))\right) \tag{3.1.42}$$

$$\le E\left(\exp\left\{-\int_{V_T(x)} \beta_T^*(s)\, ds\right\} I(V_T(x) \subset (-m, m))\right) \tag{3.1.43}$$

$$= E\left(\exp\left\{-T\sum_{i=1}^{K} \int_{V_T(x)} h_T(s, y_i)\, ds\right\} I(V_T(x) \subset (-m, m))\right) \tag{3.1.44}$$

$$\approx E\left(\exp\left\{-T \cdot \#\left(V_T(x) \cap \{y_i,\, i = 1, \dots, K\}\right)\right\} I(V_T(x) \subset (-m, m))\right). \tag{3.1.45}$$

Let A denote the event

$$A = \left\{V_T(x) \subset \left(y_i + \frac{1}{2T}, y_{i+1} - \frac{1}{2T}\right),\ \text{for some } i \in \{0, \dots, K\}\right\}, \tag{3.1.46}$$

where $y_0 = -2m$ and $y_{K+1} = 2m$. Invoking Theorem 9, notice that, the expectation taken over all possible realizations of $V_T(x)$ on the event A^c is of order $o(T)$ as $T \to \infty$. Whereas on the event A, the expectation is equal to $\exp\left\{-\pi^2 \left(\frac{K+1}{2m}\right)^2 T + o(T)\right\}$ as $T \to \infty$. Overall we have,

$$\inf_{\beta \in \mathcal{B}} P(\sigma_R > T) \leq \exp\left\{-\pi^2 \left(\frac{K+1}{2m}\right)^2 T + o(T)\right\} \tag{3.1.47}$$

as $T \to \infty$, which completes the proof of Lemma 13.

□

3.2 Disscusion

Now that we have introduced a continuous stochastic relaxation to the combinatorial optimization problem, we wish to point out some more advantages that stochastic approach has over the deterministic one.

Apart from potentially more tractability that was discussed earlier, another advantage of using the randomized method arises from the modeling point of view. In reality, the detectors might suffer from technical limitations, for example the problem of detecting the presence or absence of a weak signal, when the white signal noise statistics are unknown but the fact that its distribution is within a particular class of distributions, or even the question of the reliability of the device.

Moreover, one also needs to take into account a scenario where the adversary learns about the whereabout of some detection unit. In this case, he will avoid getting near it and from practical aspect the detector is useless.

All these features can be easily incorporate into the stochastic relaxation in the following manner. Let us denote by p the probability that agent detected upon encountering a given trap. Our current model formulation assumes that $p = 1$. We can assume that $p \in (0, 1]$ and that if the target avoids detection of such given trap, then the trap becomes ineffective, in other words, the target now will always be able to evade such trap. Assuming that traps can thus be neutralized with probability p allows to incorporate a more realistic feature in the model. Such feature can be captured within our mathematical framework of randomized deployment policies by invoking the thinning theorem. In particular, introducing such a parameter p in $(0,1]$ amounts to replacing the intensity β by $(1-p)*beta$, and, therefore, also the total budget K by $(1-p)K$. Hence, without loss of generality, we will formulate our results and solve for $p=1$.

Furthermore, the stochastic approach also naturally induces some duality to the problem. Using the notion of the Wiener sausage, by conditioning on the path of the adversary by the given deadline T, we can choose to solve the problem form either the adversary or the "defender" point of view. This observation will become handy when designing a simulation scheme, as we shall see in the next chapter.

Before moving on to the general multi-dimensional framework, we would like to point out one main difference between the one-dimensional and the multi-dimensional frameworks. In the first framework, a placement of a detector results in disconnecting the region G (which is simply an interval). This is, however, not necessarily the case in the later framework. This difference does not affect the nature of the large deviation in both cases. Yet, it affects the resulting optimal policy (unless we impose some additional structural constrains), as we shall see in Section 3.4.

3.3 The General Scenario

In this section we will consider a problem of detecting an adversary that moves in a bounded region $G \subset \mathbb{R}^d$, for some $d \geq 1$. We will assume that the agent starts from a random point in the region and keep on continuously moving according to a Brownian motion with a general drift. This is a natural extension of the one dimensional case described in Section 3.1.

We will describe the problem using a two-players game. This will enable us to introduce the mathematical framework we will work in and formulate in detail the model and the assumptions we impose on the model.

Consider a game which takes place in a bounded and connected domain G in $\mathbb{R}^d$, $d \geq 1$, with a non-empty C^2 boundary. The game is played by two players and each round lasts T units of time. Here is the description of the dynamics of the game.

3.3.1 Player 1

Player 1 starts each round at a random point x_0 in G, which is distributed according to some distribution that might be unknown. From this point Player 1 starts moving (continuously) within the domain until the end of the round or until he gets caught by Player 2, whichever occurs first (we will explain what Player 2 does in the subsequent section). We assume that Player 1 wishes to survive but he has a mission to accomplish. We can characterize his movement via a diffusion process $(X(t) : t \geq 0)$ that corresponds to the following stochastic differential equation (SDE)

$$dX(t) = \nabla Q(X(t))\,dt + dB(t)\,, \tag{3.3.1}$$

where $B(t)$ is a standard Brownian motion in $\mathbb{R}^d$. $Q(\,\cdot\,)$ can be interpret as an instantaneous reward for the player, and the he seeks to move locally in order to maximize its reward. The reward might be associated, as mentioned earlier, to accomplishing a mission.

A simple example of the dynamics of Player 1 arises when the reward is constant throughout the domain. In this case, since Player 1 has no preference to which path he should take, he simply moves according to a Brownian motion (since $\nabla Q \equiv 0$). A more interesting example occurs when there are places with high reward. In this case, we can model Q as a linear combination of Gaussian functions, where the means of the densities represent locations that are desirable (or undesirable), if the sign of the coefficient in the linear combination is positive (or negative).

In general, we will assume that $Q \in C^2\left(\bar{G}\right)$. Notice that this assumption implies that there exists some $\delta > 0$ such that

$$\delta \leq \exp\left(Q(x)\right) \leq \frac{1}{\delta} \quad \forall x \in \bar{G} \tag{3.3.2}$$

and that ∇Q is bounded on $\bar{G}$.

A key quantity in our computations is the principal eigenvalue of the generator, L, of the SDE (3.3.1). L is a differential operator defined by

$$L = -\frac{1}{2}\triangle + \nabla Q \cdot \nabla, \tag{3.3.3}$$

where $\triangle$ is the Laplace operator. The eigenvalue problem of L on $G \subset \mathbb{R}^d$ with Dirichlet boundary condition is given by

$$\begin{cases} Lu = \lambda u & x \in G \\ u = 0 & x \in \partial G \end{cases} \tag{3.3.4}$$

Under our assumptions, it is known that L has a sequence of strictly positive eigenvalues

$$0 < \lambda_1(G) \leq \lambda_2(G) \leq \dots \tag{3.3.5}$$

(see e.g., [Pinsky, 1995]). The principle eigenvalue of L in G is defined as the bottom of its spectrum (i.e., $\lambda_1(G)$) and for the rest of this chapter we will use the notation

$$\lambda(G) := \lambda_1(G). \tag{3.3.6}$$

As we shall see, at the end of each round Player 1 would have maximized his probability of survival by staying within a region which is free of detectors.

3.3.2 Player 2

The goal of **Player 2** is to catch Player 1 by the end of the round. To achieve this goal, at the beginning of each round Player 2 places several detection units (traps) within G. Once the units are deployed they stay stationary until the end of the round, at which point they can be redeployed. We assume that Player 2 has no (or little) knowledge of the distribution of the starting point of Player 1 (namely, x_0) and his reward function $Q(\cdot)$. In this formulation, we assume that Player 2 knows $Q(\cdot)$ exactly, but an extension in which $Q(\cdot)$ is itself random can be easily incorporated.

The placement policy that Player 2 uses is represented by a function $\beta(\cdot)$ and the number of units that are available for him is limited by a budget constraint which can be expressed as

$$\int_G \beta(x)\, dx \leq K, \tag{3.3.7}$$

where K might depend on the volume of the set G.

Assume that each unit has a detection radius $a = a(T)$ and let us denote by $\mathcal{P}_\beta = \{P_1, P_2, \dots\}$ the locations in G which were selected according to the policy β. Therefore, we can express the coverage configuration (denoted by $\mathcal{R}_\beta$), that is induced by placing the detectors at locations $\mathcal{P}_\beta$, as the union of the balls

$$\mathcal{R}_\beta = \bigcup_{P_i \in \mathcal{P}_\beta} B\left(P_i, a\right) \tag{3.3.8}$$

where $B\left(x, r\right)$ denotes the Euclidean open ball centered at x with radius r. Let $\sigma_{\mathcal{R}_\beta}$ denote the time Player 1 is detected by Player 2, and let $\alpha_\beta\left(x, T\right)$ denote the probability that Player 1 has not been detected by the end of the round given that he started the round at point x. Formally, $\sigma_{\mathcal{R}_\beta}$ is the first time the process $\left(X\left(t\right) :\, t \geq 0\right)$ enters the set $\mathcal{R}_\beta$,

$$\sigma_{\mathcal{R}_\beta} = \inf\left\{t \geq 0 :\; X\left(t\right) \in \mathcal{R}_\beta\right\} \tag{3.3.9}$$

and $\alpha_\beta\left(x, T\right)$ is the probability

$$\alpha_\beta\left(x, T\right) = P_x\left(\sigma_{\mathcal{R}_\beta} > T\right). \tag{3.3.10}$$

Below we describe an algorithm that can be used to generate a Poisson cloud with intensity $\beta\left(x\right)$.

Algorithm 5: Generating a Poisson Cloud

```
**Input**: Policy $\beta(x)$, domain $G$, grid size $\epsilon > 0$
**Output**: $\mathcal{P}$ a Poisson cloud in $G$ with intensity $\beta$
Divide $G$ into cubes $\{\square_i\}_{i \in \mathcal{I}}$ with side length $\epsilon$
For each square $\square$
begin
    Set $\lambda_\square = \int_\square \beta(x)\, dx$
    Sample $N_\square \sim Poisson(\lambda_\square)$
    Given $N_\square = n$: Sample $p_1, \ldots, p_n \sim f \propto \beta |_\square$
```

However, we will approach the problem from a different direction. As we mentioned before, one of advantages of relaxing the problem using a Poisson point process is that it gives rise to a dual way of viewing the problem. Recall that, the Wiener sausage of radius a generated by the process $\left(X\left(t\right) : 0 \leq t \leq T\right)$ is defined by

$$W_T^a = \bigcup_{0 \leq s \leq T} B\left(X\left(s\right), a\right).$$

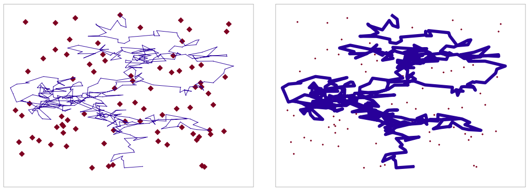


Figure 3.3.1: Poisson Cloud Vs. Wiener Sausage

By conditioning on the sample path generated by time T, we can rewrite equation (3.3.10) and obtain that

$$\begin{aligned}&P_x\left(\sigma_{\mathcal{R}} > T\right) = E_x\left[\mathbf{1}\left\{X\left(s\right) \notin \mathcal{R},\ \forall s \in [0,T]\right\}\right] = \\ &E_x\left[P\left(\left\{X\left(s\right) \notin \mathcal{R},\ \forall s \in [0,T]\right\} \mid \left\{X\left(s\right),\ s \in [0,T]\right\}\right)\right] = \\ &E_x\left[\exp\left\{-T \int\limits_{W_T^a} \beta\left(s\right) ds\right\}\right]\end{aligned} \tag{3.3.11}$$

(see Figure 3.3.2). This duality allows us to simulate the problem form the adversary (Player 1) point of view. In other words, instead of realizing the allocation policy an checking whether or not the target crossed paths with the set $\mathcal{R}$ by time T, we can simulating the movement of the target up to time T and calculate the volume that its corresponding Wiener sausage occupied in $\mathbb{R}^d$.

3.4 Optimal Policy: Two-Dimensional Brownian Motion

In this section we will consider an adversary that moves according to a Brownian motion within a domain $G \subset \mathbb{R}^2$. As in the one-dimensional, we assume that there is a containment mechanism in place. In other words, we assume that once the target reaches the boundary, he is caught and neutralized. Following the large deviation result will be established in Theorem 14 (see Chapter 4 Section 4.1) , our goal is to solve the optimization problem

$$\sup_{\beta} \inf_{D \subset \Omega} \left(\int\limits_{D} \beta\left(s\right) ds + \lambda\left(D\right) \right). \tag{3.4.1}$$

The key to understand the structure of the solution to this optimization problem lays with behavior of the principle Dirichlet eigenvalue in a domain with holes (see e.g., [Rauch and

Taylor, 1975], [Flucher, 1995] and [Harrell *et al.*, 2001]). First, notice that unlike the one-dimensional case, in the multi-dimension scenario, the placement of an obstacle or more might not result in disconnecting the domain G. Moreover, it is helpful to keep in mind that the eigenvalues of the Laplacian, which is the generator of a Brownian motion, are invariant under translations and rotations.

3.4.1 Placing K Traps in G

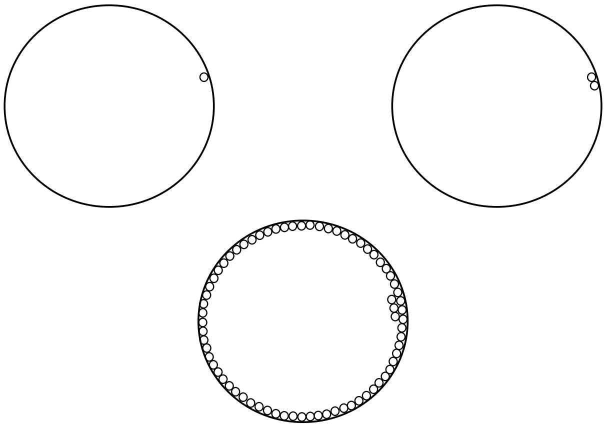
In this section we will assume that the domain G is an Euclidean ball of radius m, for some $m > 0$ Since eigenvalue of the Laplacian is invariant under translation, without loss of generality, we can assume that G is centered at the origin.
The two-dimensional version of the Faber-Krahn inequality (see [Fab, 1923] and [Krahn, 1925]), states that

$$\lambda\left(B\left(x, \sqrt{\frac{A}{\pi}}\right)\right) = \min\left\{\lambda\left(D\right) : Area\left(D\right) = A\right\}. \tag{3.4.2}$$

That is, among all domain in $\mathbb{R}^2$ of area A, the domain that minimizes the principle eigenvalue of the Laplacian is a ball. Following the Faber-Krahn inequality, the intuition is that the configuration $\mathcal{R}$ of traps that will minimize the eigenvalue when the radius of the ball inscribed in $G \setminus \mathcal{R}$ is maximal.

Therefore, if we will not impose any other constraints (apart from those who were mentioned in Section 3.3), then an optimal placement strategy will do the following: The policy will start by picking a random point on the boundary and place the first trap so that its boundary will be tangent form within to ∂G at the chosen point (see Figure 3.4.1 (a)). Next, the second trap will be placed in such a way that its boundary will be tangent both to ∂G and to the boundary of the first trap(see Figure 3.4.1 (b)). Once the boundary is completely covered, the strategy will place the following traps to create a second layer (see Figure 3.4.1 (c)) and so on.

This, however, is not a sensible solution in our case. Unless, perhaps, if we relax the assumption that the adversary is stopped once he reaches the boundary.



(*a*) (*b*) (*c*)

Figure 3.4.1: Minimal traps allocation in a ball

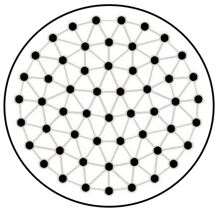


Figure 3.4.2: Tessellation of trap allocation in a ball

Thus, we will impose an additional constraint on the problem. In addition to the previous constraints, we will only consider deployment strategies that ensures that there will be distance of at least $d_a(K)$ between the individual detectors and from each detector to the boundary.

In this case, if the distance $d_a(K)$ satisfies

$$\frac{2\pi}{|\log(a)|}\left(d_a(K)\right)^{-2}\underset{a\to 0}{\longrightarrow}0. \tag{3.4.3}$$

Then the optimal strategy will result with a tessellation like configuration (see Figure 3.4.2). For more information see [Rauch and Taylor, 1975] and [Flucher, 1995].

This result is more sensible and induces a partition on the set G. Moreover, this solution suggests an alternative formulation of the problem.

3.4.2 Partitioning G to K Subsets

Instead of placing K detectors, we can consider partition G into K importance subset. In the context of catching an adversary, the alternative formulation would be the following. We have the task of partitioning the region G into K subregions, each of size proportional to likelihood of the presence of the target. Namely, areas in which the adversary is more likely to be present will be smaller, and the other way around. To impose the partition on G we could think of placing fences, that detect the target as soon as it get close enough, or as the routes along which patrol. The later scenario can be modeled as a fence with a probability p of detection.

The "detectors" in case has a very small detection radius (which is consistent with the assumption that $a(T) \to 0$ as $T \to \infty$ as we shall see later), and can be considered as part of a fence. In other words, we can think of the fence as a long string of "tiny detectors" aligned in a certain shape.

Here is one way to characterize the intensity $\beta(\cdot)$ in this case. Let $\phi(x, \mu)$ be the Gaussian density centered at μ. That is,

$$\phi(x, \mu) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2}\right). \tag{3.4.4}$$

Then we can set $\beta(\cdot)$ such that,

$$-\log \beta(x) \propto \sum_{i=1}^{K} \phi(x, \mu_i). \tag{3.4.5}$$

After solving for optimal centers μ_i, the fence will be placed along the lines that consist only with the saddle points of the optimal strategy. Notice that after a partition is obtained, we could still place K detection units (one per each subregion). However, this allocation will be different than the one the that will be obtain by considering the previous framework.

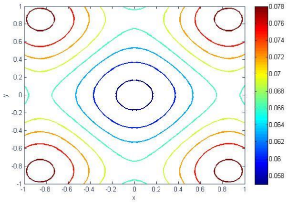


Figure 3.4.3: Level contours to determine fence location

Chapter 4

Detection of Adversarial Agents: Main Results and Algorithms

Introduction

In this chapter will provide a method to evaluate a strategy that deploys K detectors in a the domain G asymptotically as the deadline T tends to infinity. We will work in the same framework that was described earlier in Section 3.3.

First, recall that the configuration of traps in the space can be represented by the set $\mathcal{R}_\beta = \bigcup_{P_i \in \mathcal{P}} B\left(P_i, a\right)$ and the first hitting time of the diffusion process $\left(X\left(t\right) : t \geq 0\right)$ to this set were defined by $\sigma_{\mathcal{R}_\beta} = \inf\left\{t \geq 0 : X\left(t\right) \in \mathcal{R}_\beta\right\}$.

In this chapter we aim to achieve two goals. The first is to find the optimal policy β^* that solves the optimization problem

$$\min_{\beta} \max_{x \in G} P_x\left(\sigma_{\mathcal{R}_\beta} > T\right). \tag{4.0.1}$$

The second goal it to construct an efficient simulation scheme to sample from the law of $\left\{X\left(T\right) \mid \sigma_{\mathcal{R}_\beta} > T\right\}$, the location distribution of an adversary, given that he avoided detection by time T. Notice that as T tends to infinity the event $\left\{\sigma_{\mathcal{R}_\beta} > T\right\}$ becomes a rare event.

In section 4.1 we will state our main theorem (Theorem 14) , which is a large deviation result for $\alpha_\beta\left(x, T\right) = P_x\left(\sigma_{\mathcal{R}_\beta} > T\right)$. The proof of Theorem 14 will be divided into two parts, which will be given in Sections 4.2 and 4.3. We will conclude with a simulation schemes that sample $P_x\left(\sigma_{\mathcal{R}_\beta} > T\right)$ and from the conditional distribution $\left\{X\left(T\right) \mid \sigma_{\mathcal{R}_\beta} > T\right\}$, which will be described in Section 4.4. We would like to point out that these algorithms generalize the results that were presented in [Blanchet and Dupuis, 2014].

4.1 Main Result

We will start by stating our main result.

Theorem 14. *Let* $\left(X\left(t\right) : t \geq 0\right)$ *as defined in equation* (3.3.1) *with* $Q \in C^2\left(\bar{G}\right)$*. Assume that there exists a constant* $\tilde{\delta} > 0$ *such that*

$$\tilde{\delta} \leq \beta\left(x\right) \qquad \forall x \in \bar{G}.$$

Moreover, assume that $a = a\left(T\right)$ *satisfies*

$$Ta^d\left(T\right) \underset{T\to\infty}{\longrightarrow} 0. \tag{4.1.1}$$

Then,

$$\alpha_\beta\left(x,T\right) = P_x\left(\sigma_{\mathcal{R}_\beta} > T\right) = \exp\left\{-TI\left(x\right) + o\left(T\right)\right\} \qquad \forall x \in G, \tag{4.1.2}$$

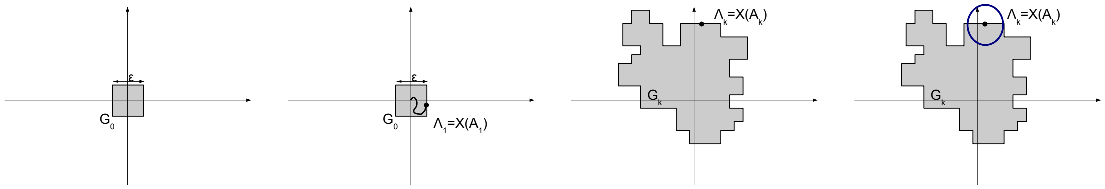
where $I\left(x\right) = \inf\limits_{\substack{x\in D\\ D \text{ open}}} \left\{\int\limits_D \beta\left(s\right) ds + \lambda\left(D\right)\right\}$ *and* $\lambda\left(D\right)$ *is as defined in equation* (3.3.6).

We will divide the proof of Theorem 14 into two parts. We will start by showing that

$$P_x\left(\sigma_{\mathcal{R}_\beta} > T\right) \leq \exp\left\{-TI\left(x\right) + o\left(T\right)\right\}. \tag{4.1.3}$$

The strategy of proving that consists of partitioning $\mathbb{R}^d$ by a grid of size $\epsilon = \epsilon\left(d\right)$, where $\epsilon > 0$ is small enough. Then we use the grid to approximate the Wiener sausage W_T^a form within as time progresses and the process evolve. In order to control the growth of W_T^a, we introduce a change of measure that relates to a stopping time τ, which is defined as follows. For every open set $G \subset \mathbb{R}^d$ we define the stopping time τ_G by

$$\tau_G := \inf\left\{t \geq 0 : X\left(t\right) \notin G\right\}. \tag{4.1.4}$$



(a) (b) (c) (d)

Figure 4.2.1: Construction of the set G_k

That is, τ_G represent the first time the process $(X(t) : t \geq 0)$ exits the set G. We also need the following lemma.

Lemma 15. *Let us make the same assumptions as in Theorem 14. Then, there exists a constant $K = K(d, Q)$ that depends only on the dimension d and the function Q, such that for every $\rho \in (0, 1)$ and $x \in G$ we have*

$$E_x \left(\exp\left\{(1-\rho)\lambda(G)\tau_G\right\}\right) \leq \frac{K(d,Q)}{\rho^{1+d/2}}. \tag{4.1.5}$$

Proof. See Appendix B.1. □

In Section 4.3 we will provide the proof of the lower bound, namely,

$$P_x\left(\sigma_{\mathcal{R}_\beta} > T\right) \geq \exp\left\{-TI(x) + o(T)\right\}. \tag{4.1.6}$$

The proof uses the fact that the intensity $\beta(\cdot)$ is bounded.

4.2 Upper bound

Let $\mathbb{Z}^d$ be the $d-$dimensional integer lattice and for $\epsilon > 0$ define $\mathcal{Z}_\epsilon = \epsilon^{\frac{1}{d}}\mathbb{Z}^d$. For each point c in $\mathcal{Z}_\epsilon$ we assign an open cube of volume ϵ centered at c, $R_\epsilon(c)$. Notice that for $c_1, c_2 \in \mathcal{Z}_\epsilon$, $c_1 \neq c_2$ we have $R_\epsilon(c_1) \cap R_\epsilon(c_2) = \emptyset$. Moreover, $\bigcup_{c \in \mathcal{Z}_\epsilon} \overline{R_\epsilon(c)} = \mathbb{R}^d$. For every $d \geq 1$ we pick $\epsilon = \epsilon(d)$ such that a cube of volume ϵ centered at the origin is inscribed in a sphere of radius a, which is also centered at the origin. For example, for $d = 2$ we pick $\epsilon \leq 2a^2$, for $d = 3$ we let $\epsilon \leq \left(\frac{4}{3}\right)^{3/2} a^3$, and so on. Our strategy is therefore, to track the Wiener sausage W_T^a using these $\epsilon-$squares and sequentially apply a change of measure adapted by the underlying process in the following manner.

4.2.1 Initialization

Initially we start with the set $G_0 = R_\epsilon(0)$, a $d-$dimensional cube of volume $\epsilon = \epsilon(d)$ centered at zero. Let $A_0 = 0$, $\Lambda_0 = 0 \in \mathbb{R}^d$ and define

$$\varphi_0(x) = E_x\left[\exp\left\{\theta_0 \tau_{G_0}\right\}\right], \tag{4.2.1}$$

where τ_{G_0} is as defined in (4.1.4), $\theta_0 = \lambda(G_0)(1-\rho_0)$ and $\rho_0 \in (0,1)$. We define the process $\{X_0(t) : t \geq 0\}$ by

$$\{X_0(t) : t \geq 0\} \overset{d}{=} \{X(t) : t \geq 0\}, \, X_0(0) = \Lambda_0 = 0, \tag{4.2.2}$$

and apply a change of measure that corresponds to the martingale

$$M_0(t) = \frac{\varphi_0(X_0(t \wedge \tau_{G_0}))}{\varphi(X_0(0))} \cdot \exp\left\{\theta_0(t \wedge \tau_{G_0})\right\}. \tag{4.2.3}$$

The intuition behind this choice of measure is the fact that the hitting time to the boundary ∂G_0 behaves like an Exponential random variable with rate $\lambda(G_0)$. Hence, this change-of-measure will drive the process to stay within G_0 as long as possible (see Figure 4.2.1(a)). We let the process evolve under the new measure and detect the first time it hits the boundary, A_1, and the corresponding hitting location $\Lambda_1 = X_0(A_1)$ (see Figure 4.2.1(b)).

4.2.2 Step k, $k \geq 1$

At the beginning of the kth step $k = 1, 2, \ldots$ we have at hand the path up to time A_k and the current location $\Lambda_k = X_{k-1}(A_k)$. First, we construct the next set of interest G_k,

$$G_k = G_{k-1} \cup \left\{R_\epsilon \text{ that adjacent to } \Lambda_k\right\}. \tag{4.2.4}$$

That is, we slightly enlarge the set G_{k-1} by adding another $\epsilon-$cube to the edge adjacent to Λ_k (see Figure 4.2.1(c) and 4.2.1(d)). Then, similarly to what we did before, we define the function

$$\varphi_k(x) = E_x\left[\exp\left\{\theta_k \tau_{G_k}\right\}\right], \tag{4.2.5}$$

with $\theta_k = \lambda(G_k)(1-\rho_k)$, $\rho_k \in (0,1)$ and the process

$$\{X_k(t) : t \geq 0\} \overset{d}{=} \{X_{k-1}(t) : t \geq 0\}, \, X_k(0) = \Lambda_k.$$

We then apply a change of measure that corresponds to the martingale

$$M_k(t) = \frac{\varphi_k\left(X_k\left(t \wedge \tau_{G_k}\right)\right)}{\varphi\left(X_k(0)\right)} \cdot \exp\left\{\theta_k\left(t \wedge \tau_{G_k}\right)\right\}$$

and let the process run until the time hitting time to the boundary A_{k+1} and the hitting point Λ_{k+1} are detected.

Stopping criterion We stop when we reach the predetermined time T

4.2.3 Proof of upper bound

Lemma 16. *Let* $(X(t) : t \geq 0)$ *as defined in equation* (3.3.1) *with* $Q \in C^2\left(\bar{G}\right)$*. Moreover, assume that* $a = a(T)$ *satisfies*

$$Ta^d(T) \underset{T\to\infty}{\longrightarrow} 0. \tag{4.2.6}$$

Then,

$$P_x\left(\sigma_{\mathcal{R}_\beta} > T\right) \leq \exp\left\{-TI(x) + o(T)\right\} \qquad \forall x \in G, \tag{4.2.7}$$

where $I(x) = \inf\limits_{\substack{x\in D \\ D \text{ open}}} \left\{\int_D \beta(s)\,ds + \lambda(D)\right\}$ *and* $\lambda(D)$ *is as defined in equation* (3.3.6).

Proof of Upper Bound. We denote by P^M and E^M the probability and expectation under the new measure, respectively. Following the procedure that was described in the previous subsection, we denote by $N(t)$ the number of cubes that were needed to track the Wiener sausage W_a^T by time T. Therefore,

$$N(t) = \text{Last step of algorithm} - 1 \tag{4.2.8}$$

Notice that the combined likelihood ratio at time t can be written as

$$Y(t) = \exp\left\{\int_0^t \theta_{N(s)+1}\,ds\right\} \prod_{j=0}^{N(t)} \frac{\varphi_j\left(\Lambda_{j+1}\right)}{\varphi_j\left(\Lambda_j\right)}. \tag{4.2.9}$$

Therefore, we have

$$\begin{aligned}
P_x\left(\sigma_{\mathcal{R}} > T\right) &= E_x\left[\exp\left\{-T\int_{W_T^a} \beta(s)\,ds\right\}\right] \\
&= E_x^M\left[\exp\left\{-T\int_{W_T^a} \beta(s)\,ds - \int_0^T \theta_{N(s)+1}\,ds\right\} \prod_{j=0}^{N(T)} \frac{\varphi_j(\Lambda_j)}{\varphi_j(\Lambda_{j+1})}\right] \\
&= E_x^M\left[\exp\left\{-T\int_{W_T^a} \beta(s)\,ds - \int_0^T \theta_{N(s)+1}\,ds\right\} \prod_{j=0}^{N(T)} \varphi_j\left(\Lambda_j\right)\right],
\end{aligned} \tag{4.2.10}$$

since $\varphi_j(\Lambda_{j+1}) = 1$ for $j = 0, 1, \ldots, N(t)$. Lemma 15 states that there exists a constant $K(d, Q)$ such that for $\rho \in (0,1)$ we have that

$$\varphi_j(\Lambda_j) = \mathbb{E}_j\left[\exp\left\{\theta_j \tau_{G_j}\right\}\right] \leq \frac{K(d,Q)}{\rho_j^{\frac{d}{2}+1}}. \tag{4.2.11}$$

Therefore, we can pick $\rho_T = \rho(T, d, Q)$ and set $\rho_j = \rho_T$ for all j so that

$$\varphi_j(\Lambda_j) \leq \exp\{T\epsilon\}. \tag{4.2.12}$$

Consequently, we have

$$P_x(\sigma_{\mathcal{R}} > T) \leq E_x^M\left[\exp\left\{-T\int\limits_{W_T^a} \beta(s)\,ds - T\theta_{N(T)+1} + T\epsilon N(T)\right\}\right]. \tag{4.2.13}$$

Moreover, since $G_j \subset G_{j+1}$ and $G_{N(T)} \subset W_T^a$, by Property 1 of the principle eigenvalue we have that

$$\lambda(G_{j+1}) \leq \lambda(G_j) \qquad \text{and} \qquad \lambda(W_T^a) \leq \lambda\left(G_{N(T)}\right). \tag{4.2.14}$$

Keeping in mind that $\theta_j = \lambda(G_j)(1-\rho_T)$, we can write

$$\begin{aligned} P_x(\sigma_{\mathcal{R}} > T) &\leq E_x^M\left[\exp\left\{-T\left(\int\limits_{W_T^a} \beta(s)\,ds - \epsilon N(T)\right) - T\lambda\left(G_{N(T)+1}\right)(1-\rho_T)\right\}\right] \\ &\leq E_x^M\left[\exp\left\{-T\left(\int\limits_{W_T^a} \beta(s)\,ds - \epsilon N(T)\right) - T\lambda(W_T^a)(1-\rho_T)\right\}\right] \\ &= E_x^M\left[\exp\left\{-T\left(\int\limits_{W_T^a} \beta(s)\,ds + \lambda(W_T^a)\right) + T\epsilon N(T) + \rho_T T\lambda(W_T^a)\right\}\right]. \end{aligned} \tag{4.2.15}$$

Recall that ϵ was defined to be such that a $d-$dimensional cube of volume ϵ is inscribed in a $d-$dimensional sphere of radius a. Therefore, $\epsilon N(T) = O\left(T^{1/2}a^d\right)$ as $T \to \infty$. As a result, we have

$$\begin{aligned} P_x(\sigma_{\mathcal{R}} > T) &\leq E_x^M\left[\exp\left\{-T\left(\int\limits_{W_T^a} \beta(s)\,ds + \lambda(W_T^a)\right) + o(T)\right\}\right] \\ &\leq \exp\left\{-T \inf_{\substack{x\in G \\ G \text{ open}}} \left(\int\limits_G \beta(s)\,ds + \lambda(G)\right) + o(T)\right\} \end{aligned} \tag{4.2.16}$$

as $T \to \infty$, which concludes our proof. □

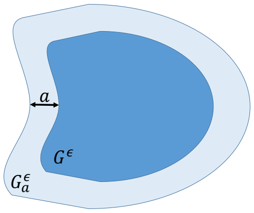


Figure 4.3.1: G^{ϵ} and G_a^{ϵ}

4.3 Lower bound

Lemma 17. *Let* $(X(t) : t \geq 0)$ *as defined in equation* (3.3.1) *with* $Q \in C^2(\bar{G})$*. Assume that there exists a constant* $\tilde{\delta} > 0$ *such that*

$$\tilde{\delta} \leq \beta(x) \qquad \forall x \in \bar{G}.$$

Moreover, assume that $a = a(T)$ *satisfies*

$$a^d(T) \underset{T\to\infty}{\longrightarrow} 0. \tag{4.3.1}$$

Then,

$$P_x(\sigma_{\mathcal{R}} > T) \geq \exp\{-TI(x) + o(T)\} \qquad \forall x \in G, \tag{4.3.2}$$

where $I(x) = \inf\limits_{\substack{x\in D\\ D\ open}} \left\{ \int\limits_D \beta(s)\,ds + \lambda(D) \right\}$ *and* $\lambda(D)$ *is as defined in equation* (3.3.6).

Proof of Lower Bound. Let $x \in \mathbb{R}^d$ and recall that $I(x)$ was defined by

$$I(x) = \inf_{\substack{x\in G\\ G \text{ open}}} \left\{ \int\limits_G \beta(s)\,ds + \lambda(G) \right\}.$$

Therefore, for every $\epsilon > 0$ we can find G^{ϵ} such that $x \in G^{\epsilon}$, G^{ϵ} is open and connected subset of $\mathbb{R}^d$ and such that

$$\int\limits_{G^{\epsilon}} \beta(s)\,ds + \lambda(G^{\epsilon}) \leq I(x) + \epsilon. \tag{4.3.3}$$

Let G_a^{ϵ} be the enlargement of the set G^{ϵ} so that the distance between the boundaries of the sets is a, as illustrated in Figure 4.3.

Notice that,

$$\tau_{G^\epsilon} > T \iff W_T^a \subset G_a^\epsilon.$$

Therefore,

$$\begin{aligned} P_x\left(\sigma_{\mathcal{R}} > T\right) &= E_x\left[\exp\left\{-T\int_{W_T^a}\beta\left(s\right)ds\right\}\right] \\ &\geq E_x\left[\exp\left\{-T\int_{W_T^a}\beta\left(s\right)ds\right\}I\left(\tau_{G^\epsilon} > T\right)\right] \\ &\geq E_x\left[\exp\left\{-T\int_{G_a^\epsilon}\beta\left(s\right)ds\right\}I\left(\tau_{G^\epsilon} > T\right)\right] \\ &= \exp\left\{-T\int_{G^\epsilon}\beta\left(s\right)ds\right\}P_x\left(\tau_{G^\epsilon} > T\right)\exp\left\{-T\int_{G_a^\epsilon\setminus G^\epsilon}\beta\left(s\right)ds\right\} \\ &\geq \exp\left\{-T\int_{G^\epsilon}\beta\left(s\right)ds - T\lambda\left(G^\epsilon\right)\right\}\exp\left\{-T\tilde{\delta}\,Vol\left(G_a^\epsilon\setminus G^\epsilon\right)\right\} \\ &\geq \exp\left\{-TI\left(x\right)-T\epsilon\right\}\exp\left\{-T\tilde{\delta}\,Vol\left(G_a^\epsilon\setminus G^\epsilon\right)\right\}. \end{aligned} \tag{4.3.4}$$

And so,

$$\frac{1}{T}\log P_x\left(\sigma_{\mathcal{R}} > T\right) \geq -I\left(x\right)-\epsilon+o\left(T\right) \tag{4.3.5}$$

as $T \to \infty$. □

4.4 Asymptotically Optimal Monte Carlo for Detection Policies

This section is divided into two parts. The first part involves evaluating a given deployment policy, so we are interested in estimating efficiently via simulation the probability $P\left(\sigma_R > T\right)$ for large T. After providing our algorithm, based on importance sampling, we shall prove that the algorithm is asymptotically optimal in the sense of controlling the rate of decay of the second moment of the associated estimator.

The second subsection involves estimating the conditional distribution of the target $X\left(\cdot\right)$ and, in particular, the position of the target at time T, given that the target has avoided detection. In both tasks we will leverage off our large deviations results. As we mentioned earlier, the algorithms that will be presented in this section generalize the results that were introduced in [Blanchet and Dupuis, 2014]. There the authors used importance sampling based algorithm to evaluate the probability of a Brownian motion that avoid hitting, traps which where distributed according to a Poisson process with constant intensity. Our algorithms can be applied to more general scenarios.

4.4.1 Estimating $P(\sigma_R > T)$

In Theorem 14 we have proved a large deviations results for $P(\sigma_R > T)$. However, as is typical in large deviations theory, the result is more interesting from a qualitative standpoint because we are only evaluating the exponential rate of decay, so lower order terms such as sub-exponential factors, which might play a significant role, are not accounted for. Therefore, it is of interest to provide efficient Monte Carlo simulation for estimating such large deviations probabilities.

When estimating large deviations probabilities one often takes advantage of a technique called importance sampling. We shall briefly review the elements of importance sampling momentarily. Large deviations suggests how to perform importance sampling by using the change of measure which is suggested by the proof of a the lower bound. There are counter-examples, however, to the use of such change of measure as an importance sampling distribution (see [Glasserman and Wang, 1997], for example). Fortunately, the proof that we presented immediately suggests an asymptotically optimal importance sampling estimator.

4.4.1.1 Importance Sampling

Importance Sampling is a general technique for estimating properties of a particular distribution, while only having samples generated from a different distribution than the distribution of interest (see e.g., [Asmussen and Glynn, 2007]). Assume that we wish to compute the expectation

$$z = E(Z),$$

where $E(\cdot)$ denotes the expectation with respect to the probability measure $P(\cdot)$. Now let $\tilde{P}$ denote a different probability measure. If the measure P is absolutely continuous with respect to $\tilde{P}$, that is, if

$$P(dw) = L(w)\,\tilde{P}(dw) \tag{4.4.1}$$

for all $w \in \Omega$ (the whole probability space). Then the Importance Sampling technique asserts that we can sample according to the measure $\tilde{P}$ and then our estimator would be

$$z = E(Z) = \tilde{E}(LZ),$$

where $\tilde{E}$ is the expectation with respect to the measure $\tilde{P}$.

4.4.1.2 Algorithm for sampling $P(\sigma_R > T)$

```
Algorithm 6: Sampling $P(\sigma_R > T)$
Input: Deadline $T$, grid size $\epsilon = \epsilon(d)$.
Output: $p$
Initialization Set:
$k \leftarrow 0$, $G_0 \leftarrow R_\epsilon(0)$, $A_0 \leftarrow 0$, $\Lambda_0 \leftarrow 0$, $\varphi_0(x) \leftarrow E_x[\exp\{\theta_0 \tau_{G_0}\}]$
where $\tau_{G_0}$ is as defined in (4.1.4), $\theta_0 = \lambda(G_0)(1-\rho_0)$ and $\rho_0 \in (0,1)$.
Define $\{X_0(t) : t \geq 0\}$: $\{X_0(t) : t \geq 0\} \overset{d}{=} \{X(t) : t \geq 0\}, X_0(0) = \Lambda_0 = 0$
Apply a change of measure that corresponds to the martingale
$M_0(t) = \frac{\varphi_0(X_0(t \wedge \tau_{G_0}))}{\varphi(X_0(0))} \cdot \exp\{\theta_0(t \wedge \tau_{G_0})\}$.
Simulate until $X_0(t) \in \partial G0$.
Set $A_1 \leftarrow$hitting location, $\Lambda_1 \leftarrow X_0(A_1)$ (see Figure 4.2.1($b$)).
repeat
    Set $k \leftarrow k+1$, $G_k$ as in equation (4.2.4), $\varphi_k(x) \leftarrow E_x[\exp\{\theta_k \tau_{G_k}\}]$ with
    $\theta_k = \lambda(G_k)(1-\rho_k)$, $\rho_k \in (0,1)$.
    Define:$\{X_k(t) : t \geq 0\} \overset{d}{=} \{X_{k-1}(t) : t \geq 0\}, X_k(0) = \Lambda_k$.
    Apply a change of measure that corresponds to the martingale
    $M_k(t) = \frac{\varphi_k(X_k(t \wedge \tau_{G_k}))}{\varphi(X_k(0))} \cdot \exp\{\theta_k(t \wedge \tau_{G_k})\}$
    Simulate until $X_k(t) \in \partial G_k$.
    Set $A_{k+1} \leftarrow$hitting location, $\Lambda_{k+1} \leftarrow X_0(A_1)$
until Deadline $T$;
Set $p \leftarrow \exp\left\{-T \int_{W_T^a} \beta(s)\,ds - \int_0^T \theta_{N(s)+1} ds\right\} \prod_{j=0}^{N(T)} \varphi_j(\Lambda_j)$
where $N(s) = k$ at time $s$.
return $p$
```

4.4.1.3 Asymptotic optimality

We will now argue that our algorithm is asymptotically optimal. First, let us recall the definition of asymptotically optimal estimator (see [Asmussen and Glynn, 2007], for example).

Definition 18. *$Z(T)$ is an asymptotically optimal estimator, if for any $\varepsilon > 0$ we have that*

$$\sup_T \frac{EZ^2(T)}{(E(Z(T)))^{2-\varepsilon}} < \infty.$$

The definition intuitively indicates, that

$$\log EZ^2(T) \approx \log\left(E(Z(T))^2\right)$$

as $T \to \infty$ and therefore the variance is controlled as $T \to \infty$. In other words, the relative variance grows at most sub-exponentially as $T \to \infty$.

Theorem 19. *The estimator $Z(T)$ given in Algorithm 6 is asymptotically optimal.*

Proof. Let $\varepsilon > 0$. Then,

$$\begin{aligned} \frac{EZ^2(T)}{(E(Z(T)))^{2-\varepsilon}} &= \frac{E\left(\exp\left\{-2T\int\limits_{W_T^a}\beta(s)ds - 2\int\limits_0^T\theta_{N(s)+1}ds\right\}\prod\limits_{j=0}^{N(T)}(\varphi_j(\Lambda_j))^2\right)}{(P(\sigma_{\mathcal{R}}>T))^{2-\varepsilon}} \\ &\leq \frac{\exp(2(IT+o(T)))}{\exp((2-\varepsilon)(IT+o(T)))}, \end{aligned} \tag{4.4.2}$$

where the last inequality is due to Theorem 14.

□

4.4.2 Sampling form P_* the law of $\{X(T)\,|\,\sigma_{\mathcal{R}} > T\}$

Another question that is of interest is the following. Assume that we have not managed to detect the adversary by time T. Giving this information what can we say about his current position? How can we sample form the law of $\{X(T)\,|\,\sigma_{\mathcal{R}} > T\}$?

For the case of a Brownian motion moving in a Poisson cloud with constant intensity, [Sznitman, 1991] (for $d = 2$) and [Povel, 1999] (for $d \geq 3$) showed that the probability that the W_T^a is confined in a ball of radius $c\,t^{1/(d+2)}$, for some (computable) $c > 0$, given that the sausage does not include any point of the Poisson cloud, goes to 1 as t goes to ∞.

In the general case, we can use the procedure described earlier to find the typical shape of W_T^a. Let P^M denote the measure induced by the procedure described in Algorithm 6. Recall that

$$dP^M = Y(T)\,dP$$

with

$$Y(t) = \exp\left\{\int_0^t \theta_{N(s)+1} ds\right\} \prod_{j=0}^{N(t)} \frac{\varphi_j(\Lambda_{j+1})}{\varphi_j(\Lambda_j)}. \tag{4.4.3}$$

We propose the following procedure. First sample from the law P^M and obtain $\mathbf{X}_1, \ldots, \mathbf{X}_n$ IID sample paths. For each such sample path we can compute a weight

$$\xi_i = \exp\left(-\int_{(W_T^a)_i} \beta(s)\,ds\right)(Y_i(T))^{-1}, \tag{4.4.4}$$

where $(W_T^a)_i$ and $Y_i(T)$ are the Wiener sausage and the likelihood ratio, respectively, that correspond to the sample path $\mathbf{X}_i$. We define $\hat{\mu}^n$ by

$$\hat{\mu}^n(dx) = \frac{1}{n}\sum_{i=1}^n \delta_{\mathbf{X}_i}(dx)\cdot\xi_i \tag{4.4.5}$$

and set

$$\hat{\mu}_*^n(dx) = \frac{\sum_{i=1}^n \delta_{\mathbf{X}_i}(dx)\cdot\xi_i}{\sum_{j=1}^n \xi_j}.$$

Observe that that $\hat{\mu}_*^n$ is a biased, but consistent, estimator of P_* in the sense that for every continuous and bounded $f : C[0,1] \to C[0,\infty)$, we have that

$$\int f(x)\,\hat{\mu}_*^n(dx) \to E[f(\mathbf{X}) \mid \sigma_R > T].$$

Moreover, due to the asymptotic optimality of our estimator for $P(\sigma_R > T)$ given in our algorithm in the previous subsection, we have that

$$E\left[\left|E[f(\mathbf{X})I(\sigma_R > T)] - \int f(x)\,\hat{\mu}^n(dx)\right|^2\right] = Var_M\left(\int f(x)\,\hat{\mu}_*^n(dx)\right) \tag{4.4.6}$$

$$= \frac{1}{n} Var_M(f(\mathbf{X}_i)\,\xi_i) \tag{4.4.7}$$

$$\leq \|f\|_2^2 \cdot \frac{1}{n} Var_M(\xi_i) \leq \|f\|_2^2 \cdot \frac{c}{n} \cdot P(\sigma_R > T)^{2-\varepsilon}, \tag{4.4.8}$$

where $Var_M(\cdot)$ is the variance under the probability P^M.

It is important to note that the mean-squared error of our estimator for $E\left[f(\mathbf{X}) \mid \sigma_R > T\right]$ can be controlled due to (4.4.8) even if T is large. In particular, it suffices to let let $n = O\left(1/P\left(\sigma_R > T\right)^{\varepsilon}\right)$ for any fixed $\varepsilon > 0$ in order to provide accurate estimates.

Now, note that $\int f(x)\,\hat{\mu}_*^n(dx)$ has a small bias which is asymptotically negligible, but such bias, as we can see, can actually be deleted due to the fact that our importance sampling estimator induced a bounded likelihood ratio. We will explain this feature next.

4.4.3 Sampling $\{X(T) \mid \sigma_{\mathcal{R}} > T\}$ when the rate function I is computable

If the rate function I is computable, then we can device an Acceptance-Rejection based algorithm to produce an exact sampler to the conditional probability. First, let us recall the general Acceptance-Rejection procedure.

4.4.3.1 Acceptance-Rejection

The Acceptance-Rejection algorithm is a well known sampling technique, which was proposed by John Von Neumann on 1951 ([von Neumann, 1951]). It might be considered one of the most important building block in field of Simulation Theory. The goal of the Acceptance-Rejection algorithm is to generate samples from a distribution $P(\,\cdot\,)$, which is usually unaccessible or unknown in closed form, using and another distribution $\tilde{P}(\,\cdot\,)$, which is easier to simulate. The measure P needs to be absolutely continuous with respect to $\tilde{P}$ and

$$\frac{dP(x)}{d\tilde{P}(x)} \leq C < \infty$$

for all x for some constant C.

The algorithm is then executed as follows:

Algorithm 7: Acceptance-Rejection Algorithm

```
Output: $x \sim P$
repeat
    Sample $x \sim \tilde{P}$
    Sample $u \sim U(0,1)$ independent of $x$
until $u \leq \frac{1}{C} \cdot \frac{dP(x)}{d\tilde{P}(x)}$;
return $x$
```

4.4.3.2 Acceptance-Rejection based algorithm for sampling $\{X(T) \,|\, \sigma_{\mathcal{R}} > T\}$

Let $\mathbf{X} := (X(t) : 0 \le t \le T)$ denote the path of the adversarial agent until the given deadline T, and let $\mathcal{N}$ denote the law of the Poisson cloud with intensity β. Consider the following joint density

$$\mathbb{P}_*(d\omega) = P(d\omega|\, \sigma(\omega) > T) = \frac{P(d\omega)\, I(\sigma(\omega) > T)}{P(\sigma(\omega) > T)}\,, \tag{4.4.9}$$

where $\omega = (\mathbf{X}, \mathcal{N})$ and $\sigma(\omega) = \sigma_{\mathcal{R}}$. We are interested in sampling form its marginal distribution, $\mathbb{P}_*(d\mathbf{X})$. To this end, we will integrate the joint density given in equation 4.4.9 and obtain

$$\begin{aligned}
\mathbb{P}_*(d\mathbf{X}) &= \int\limits_{\mathcal{N}\in\Omega} \mathbb{P}_*(d\mathbf{X}, d\mathcal{N}) && (4.4.10)\\
&= \int\limits_{\mathcal{N}\in\Omega} \frac{E\left[I(d\mathbf{X} \in dx,\, d\mathcal{N} \in d\eta)\, I(\sigma(\omega) > T)\right]}{P(\sigma(\omega) > T)} && (4.4.11)\\
&= \int\limits_{\mathcal{N}\in\Omega} \frac{P(I(\mathcal{N} \in d\eta,\, I(\sigma(\omega) > T)|\, \mathbf{X}))}{P(\sigma(\omega) > T)}\, P(d\mathbf{X}) && (4.4.12)\\
&= \frac{\exp\left(-\int\limits_{W_T^a} \beta(s)\, ds\right)}{E\left[\exp\left(-\int\limits_{W_T^a} \beta(s)\, ds\right)\right]}\, P(d\mathbf{X})\,. && (4.4.13)
\end{aligned}$$

In order to sample from $\mathbb{P}_*$, we suggest to use the Acceptance-Rejection algorithm and propose a sample from the measure P^M that was introduced in Subsection 4.2.

Notice that

$$P^M(\omega) = P^M(d\mathbf{X})\, P(d\mathcal{N}|\, \mathcal{N} \cap W_T^a,\, \mathbf{X})\,. \tag{4.4.14}$$

Moreover,

$$P(\mathbf{X} \in A|\, \sigma > T) = \frac{E\left[\exp\left(-\int\limits_{W_T^a} \beta(s)\, ds\right) I(\mathbf{X} \in A)\right]}{P(\sigma > T)}\,. \tag{4.4.15}$$

Overall, we have that the likelihood ratio is given by

$$\frac{d\mathbb{P}_*}{dP^M}(\mathbf{X}) = \frac{\exp\left(-\int\limits_{W_T^a} \beta(s)\, ds\right)(Y(T))^{-1}}{P(\sigma > T)} \le \frac{\exp(-T(1-\rho_T)\, I)}{P(\sigma > T)} := C\,. \tag{4.4.16}$$

Notice that $C = \exp\left(o\left(T\right)\right)$ as $T \to \infty$ and is finite for $T < \infty$. Therefore, the probability of accepting a proposed sample is

$$\frac{1}{C} \cdot \frac{d\mathbb{P}_*}{dP^M}\left(\mathbf{X}\right) = \frac{\exp\left(-\int\limits_{W_T^a} \beta\left(s\right) ds\right)\left(Y\left(T\right)\right)^{-1}}{\exp\left(-T\left(1-\rho_T\right)I\right)}. \tag{4.4.17}$$

Notice that the all expressions in the nominator are known (as a by product of the simulation) and we assume that the rate function I is computable, for example, if the principle eigenvalue is computable, then the Acceptance-Rejection algorithm can be applied in a straightforward way.

4.5 Numerical Results

The following subsections summarize the simulation results we obtain, comparing the performance of the importance sampling based algorithm (Algorithm 6) with a naïve Monte Carlo sampler. First, we tested the algorithm in the simple case of a one-dimensional Brownian motion (Subsection 4.5.1). Later, in Subsection 4.5.2, we considered more complex scenarios: a two-dimensional diffusion process with different choices of drift coefficients. For these scenarios we also tested the variance reduction achieved by the importance sampling based estimator. In both subsections, the implementation extends the sampling methods proposed in [Blanchet and Dupuis, 2014]. At this point we would like to thank Elioth Mirsha Sanabria Buenaventura for his help with the implementation.

4.5.1 One-dimensional Brownian Motion

In this subsection we considered a target that moves according to a one-dimensional Brownian motion. We tested the algorithm by evaluating the target's survival probability for different deployment policies. We considered three cases:

Case 1: Traps are placed according to a Poisson process with constant intensity $\beta = 0.25$.

Case 2: Traps are placed within distance 3 of one another.

Case 3: Traps are placed within distance 2 of one another.

The three cases can be fitted the scenario where the target is confined to the interval $(-6, 6)$ and we can place up to K detectors. Cases 1 and 2 correspond to $K = 3$, while Case 3 corresponds to $K = 5$. We set a deadline of $T = 10$ and use a lattice of size 0.01 for all scenarios. In each case, we sampled the Brownian motion using the importance sampling technique (Algorithm 6) in a naïve way. The estimations of the survival probability are based on sample of size 1,000 per case per technique. The estimations are given by intervals with 95% confidence.

We summarized the results in table 4.1 below. As expected, point-mass type strategy (Case 2) performs better than the uniform intensity (Case 1). Moreover, placing more traps (Case 3) reduced the probability of survival.

	Case 1		Case 2		Case 3	
	LCI	UCI	LCI	UCI	LCI	UCI
IS	$7.52 \cdot 10^{-4}$	$8.79 \cdot 10^{-4}$	$4.58 \cdot 10^{-5}$	$4.59 \cdot 10^{-5}$	$4.32 \cdot 10^{-5}$	$4.35 \cdot 10^{-5}$
Naïve	$7.38 \cdot 10^{-4}$	$9.68 \cdot 10^{-4}$	$4.58 \cdot 10^{-5}$	$4.60 \cdot 10^{-5}$	$4.32 \cdot 10^{-5}$	$4.36 \cdot 10^{-5}$

Table 4.1: Survival probability: Importance Sampling (IS) vs. naïve Simulation: One-dimensional Brownian motion.

4.5.2 Two-dimensional diffusion process

In this subsection, we considered a target that moves in the plane according to a two-dimensional Brownian motion and a drift coefficient $\nabla Q(x, y)$.

We tested the algorithm for different choices of the function Q. In particular, we considered the following four cases:

(a) $Q(x, y) = Constant$.

(b) $Q(x, y) = C\left(x^2 + y^2\right)$.

(c) $Q(x, y) = e^{-\left(x^2+y^2\right)}$.

(d) $Q(x, y) = e^{-(x-0.5)^2-(y-0.5)^2}$.

For $Q = Constant$, time horizon $T = 5$, and 10,000 samples of the survival probability $\hat{\alpha}_T$, we have

Method	$\hat{\alpha}_T$	LCI	UCI
Naïve	$6.20 \cdot 10^{-3}$	$5.42 \cdot 10^{-3}$	$6.99 \cdot 10^{-3}$
IS	$6.30 \cdot 10^{-3}$	$5.51 \cdot 10^{-3}$	$7.09 \cdot 10^{-3}$

Table 4.2: Survival probability: Importance Sampling (IS) vs. naïve Simulation: Two-dimensional Brownian motion.

where LCI and UCI represent the lower and upper 95% confidence intervals, respectively.

In the previous section we proved that Algorithm 6 is asymptotically optimal. Therefore, a natural thing was to explore when one can see the variance reduction. To this end, we tested the algorithm in each scenario for different sizes of boundary L, which is defined in the following manner: Let us denote by A_t the set of points that the process has visited by time t and assume that the process starts at x_0 at time $t = 0$. Then,

$$A_0 = \{x : d\left(x, x_0\right) \leq L\} \tag{4.5.1}$$

For example, assume that $x_0 = (0, 0)$ and $L = 2$. Then A_0 is the blue region in Figure 4.5.1 on the left. Now suppose that the process starts and the first point that it visits outside the blue set (A_0), which is the point $y = (1, 2)$ (the red point in Figure 4.5.1). The set A_1 can now be written as

$$A_1 = A_0 \cup \{x :\ d\left(x, y\right) \leq L\} \tag{4.5.2}$$

In Figure 4.5.1 (right), A_1 is represented by the union of blue and red points.

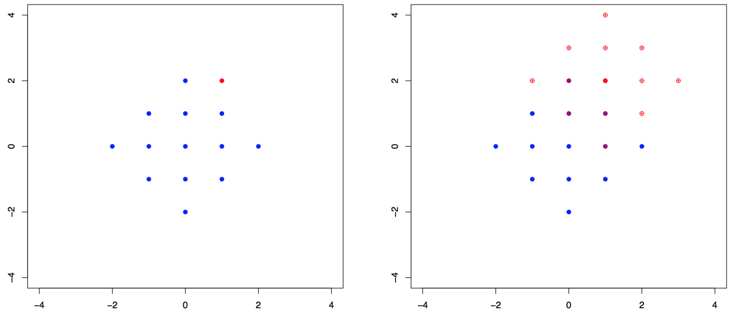
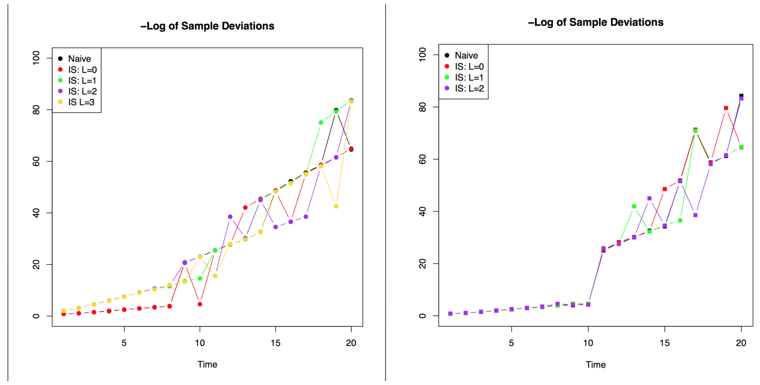
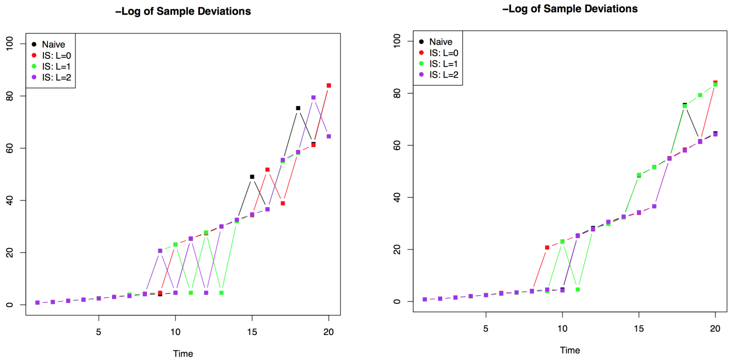


Figure 4.5.1: A_0 (on the left) and A_1 (on the right) for $L = 2$.

We ran the simulation for different choices of the function $Q(x, y)$ and boundary sizes L. For each selection of $Q(x, y)$, we simulated 10,000 samples for each choice of L, and compared the estimators' variance for different values of L. The results of the simulation are illustrated in Figure 4.5.2. The plots represent the standard deviation of the estimator (on a negative log scale). Each sub-figure corresponds to a different choice of the function $Q(x, y)$. In each sub-figure, the black line corresponds to a naïve Monte Carlo simulation, and the red, green and purple lines corresponds to the results of the importance sampling for the choices of $L = 0$, $L = 1$, and $L = 2$, respectively.



(a) $Q = Constant.$ (b) $Q = \left(x^2 + y^2\right).$



(c) $Q = e^{-\left(x^2+y^2\right)}.$ (d) $Q = e^{-(x-0.5)^2-(y-0.5)^2}.$

Figure 4.5.2: -Log of the estimator's standard deviation.

4.6 Conclusions

The problem of detecting an adversarial agent, that was discussed in this part, is combinatorially NP hard. The continuous relaxation we propose, no matter what kind of framework we consider (traps or fences), requires that we solve

$$\sup_{\beta} \inf_{D \subset \Omega} \left(\int_{D} \beta(s)\, ds + \lambda(D) \right). \tag{4.6.1}$$

with β being the intensity for either detector or fence placement. However, solving equation 4.6.1 analytically, involves knowing the value of the rate function (or alternatively, computing the value of the principle eigenvalue), which is also difficult.

On the bright side, we can use our algorithms to numerically evaluate policies and gain some knowledge regarding the location of an adversary given that he has not been detected by a given deadline.

The importance of the work presented in this part is in the fact that it lies down a mathematical foundation and it provides a new approach to the solution of finding the optimal policy that solves detection problem. There are many future directions that one could make to take our results one step forward. For example, it will be interesting to check if combining our technique with current methods for solving the Maximal Coverage problem, that was mention in the introduction. Another possible directions, are to consider more than one agent that is moving in the domain, or when the traps are not stationary.

Part III

Bibliography

Bibliography

[Asmussen and Glynn, 2007] S. Asmussen and P. W. Glynn. *Stochastic Simulation: Algorithms and Analysis.* Springer-Verlag, 2007.

[Asmussen, 2003] S. Asmussen. *Applied Probability and Queues, 2nd ed.* Springer, New York, 2003.

[Blanchet and Chen, 2012] J. Blanchet and X. Chen. Steady-state simulation for reflected Brownian motion and related networks. *http://arxiv.org/pdf/1202.2062.pdf*, 2012.

[Blanchet and Dong, 2012] J. Blanchet and J. Dong. Sampling point processes on stable unbounded regions and exact simulation of queues. In *Proceedings of the 2012 Winter Simulation Conference C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A. M. Uhrmacher, eds*, 2012.

[Blanchet and Dupuis, 2014] Jose Blanchet and Paul Dupuis. Fast simulation of brownian motion avoiding random obstacles, 2014.

[Blanchet and Sigman, 2011] J. Blanchet and K. Sigman. On exact sampling of stochastic perpetuities. *Journal of Applied Probability*, 48A:165–182, 2011.

[Donsker and Varadhan, 1975] Monroe D. Donsker and S.R.Srinivasa Varadhan. Asymptotics for th wiener sausage. *Communications on Pure and Applied Mathematics*, 1025:525–565, 1975.

[Durrett, 2005] R. Durrett. *Probability: Theory and Examples.* Duxbury Advanced Series, 2005.

[Ensor and Glynn, 2000] K. B. Ensor and P. W. Glynn. Simulating the maximum of a random walk. *Journal of Statistical Planning and Inference*, 85:127–135, 2000.

[Fab, 1923] *dass unter allen homogenen Membranen von gleicher Fläche und gleicher Spannung die kreisförmige den tiefsten Grundton gibt*, 1923.

[Flucher, 1995] Martin Flucher. Approximation of dirichlet eigenvalues on domains with small holes. *Journal of Mathematical Analysis and Applications*, 193(1):169 – 199, 1995.

[Foss and Sapozhnikov, 2006] S. Foss and A. Sapozhnikov. Convergence rates in monotone separable stochastic networks. *Queueing Systems*, 52(2):125–137, 2006.

[Glasserman and Wang, 1997] Paul Glasserman and Yashan Wang. Counterexamples in importance sampling for large deviations probabilities. *Ann. Appl. Probab.*, 7(3):731–746, 08 1997.

[Harrell *et al.*, 2001] Evans M. Harrell, Pawel Kroger, and Kazuhiro Kurata. On the placement of an obstacle or a well so as to optimize the fundamental eigenvalue. *SIAM journal on mathematical analysis*, 33(1):240–259, 2001.

[Hochbaum, 1997] Dorit S. Hochbaum. *Approximation Algorithms for NP-hard Problems*, chapter Approximating Covering and Packing Problems: Set Cover, Vertex Cover, Independent Set, and Related Problems, pages 94–143. PWS Publishing Co., Boston, MA, USA, 1997.

[Kac, 1951] Mark Kac. On some connections between probability theory and differential and integral equations, 1951.

[Kendall, 1998] W. S. Kendall. Perfect simulation for the area-interaction point process. In *Probability Towards 2000 (ed. L. Accardi and C.C. Heyde). Lecture Notes in Statistics.* volume 128, New York; Springer-Verlag, 218-234, 1998.

[Kingman and Atiyah, 1961] J. F. C. Kingman and M. F. Atiyah. The single server queue in heavy traffic. *Proceedings of the Cambridge Philosophical Society*, 57:902, 1961.

[Krahn, 1925] E. Krahn. Über eine von rayleigh formulierte minimaleigenschaft des kreises. *Mathematische Annalen*, 94(1):97–100, 1925.

[Murthy *et al.*, 2013] K. Murthy, S. Juneja, and J. Blanchet. State-independent importance sampling for random walks with regularly varying increments. *http://arxiv.org/pdf/1206.3390.pdf*, 2013.

[Nemhauser *et al.*, 1978] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions i. *Mathematical Programming*, 14(1):265–294, December 1978.

[Peres *et al.*, 2011] Yuval Peres, Alistair Sinclair, Perla Sousi, and Alexandre Stauffer. Mobile geometric graphs: Detection, coverage and percolation. In *SODA*, pages 412–428, 2011.

[Pinsky, 1985] Ross Pinsky. On the convergence of diffusion processes conditioned to remain in a bounded region for large time to limiting positive recurrent diffusion processe. *The Annals of Probability*, 13:363–378, 1985.

[Pinsky, 1995] Ross Pinsky. *Positive Harmonic Functions and Diffusion: An Integrated Analytic and Probabilistic Approach.* Cambridge, 1995.

[Povel, 1999] Tobias Povel. Confinement of brownian motion among poissonian obstacles in r^d , $d \geq 3$. *Probability Theory and Related Fields*, 114:177–205, 1999.

[Propp and Wilson, 1996] J. G. Propp and D. B. Wilson. Exact sampling with coupled Markov chains and applications to statistical mechanics. *Random Structures & Algorithms (Atlanta, Georgia: Proceedings of the Seventh International Conference on Random Structures and Algorithms)*, 9:223–252, 1996.

[Rauch and Taylor, 1975] Jeffrey Rauch and Michael Taylor. Potential and scattering theory on wildly perturbed domains. *Journal of Functional Analysis*, 18(1):27 – 59, 1975.

[Sigman, 2011] K. Sigman. Exact simulation of the stationary distribution of the FIFO M/G/c queue. *Journal of Applied Probability*, 48A:209–216, 2011.

[Sigman, 2012] K. Sigman. Exact simulation of the stationary distribution of the FIFO M/G/c queue: The general case for $\rho < c$. *Queueing Systems*, 70:37–43, 2012.

[Sznitman, 1991] Alain-Sol Sznitman. On the confinement property of two-dimensional brownian motion among poissonian obstacles. *Communications in Pure and Applied Mathematics*, 44:1137–1170, 1991.

[Sznitman, 1998] Alain-Sol Sznitman. *Brownian Motion, Obstacles and Random Media*. Springer Monographs in Mathematics, 1998.

[Vazirani, 2001] Vijay V. Vazirani. *Approximation Algorithms*. Springer-Verlag New York, Inc., New York, NY, USA, 2001.

[von Neumann, 1951] John von Neumann. Various techniques used in connection with random digits. monte carlo methods. *U.S. Dept. of Commerce, National Bureau of Standards; Applied mathematics series*, 12:36–38, 1951.

Part IV

Appendices

Appendix A

Appendix for Chapter 2

Chapter 2 describes a method of perfect sampling of stationary and time reversed queues. To ensure that our algorithm works properly, in Subsection 2.3.1.1 we formulated our assumptions regarding the parameters δ, α, m, and γ. The first section in this appendix will discuss the generality of these assumptions. The subsequent sections of this appendix we will present the proofs for the technical lemmas that were stated in Subsection 2.3.1.2.

A.1 Discussion on the generality of the assumptions imposed and selection of parameters

In this section we will argue that the inequalities (2.3.5)-(2.3.7) can always be satisfied under our underlying assumption that $E\left|X_k\right|^{\beta} < \infty$ for $\beta = 2 + \varepsilon > 2$ (the case $\beta > 1$ is discussed in Section 2.5). First, the selection of L in (2.2.1) is always feasible, as indicated earlier $L = 1$ is most of the time feasible; for example $L = 1$ will be feasible if X_1 is non-lattice.

Clearly the selection of m satisfying (2.3.5) is always feasible. Now, note that we can always select $\delta > 0$ so that

$$2 < \alpha \leq (2 + \varepsilon)(1 - \delta). \tag{A.1.1}$$

Then observe that if $m \geq 1$, applying Chebyshev's inequality,

$$\begin{aligned} \frac{6\left(1+2\mu z+m\right)^{\alpha}}{\left(\alpha-1\right)\left(m+1\right)^{\alpha-1}\mu}P(X>(\mu z+m)^{1-\delta}) &\leq \frac{6\cdot 2^{\alpha}\left(\mu z+m\right)^{\alpha}}{\left(\alpha-1\right)\left(m+1\right)^{\alpha-1}\mu}\cdot\frac{E\left[\left(X_1^{+}\right)^{2+\varepsilon}\right]}{\left(\mu z+m\right)^{(2+\varepsilon)(1-\delta)}} \\ &\leq \frac{6\cdot 2^{\alpha}}{\left(\alpha-1\right)\left(m+1\right)^{\alpha-1}\mu}\cdot E\left[\left(X_1^{+}\right)^{2+\varepsilon}\right] \end{aligned}$$

So, condition (2.3.6) is automatically satisfied if m is chosen sufficiently large so that

$$\frac{6\cdot 2^{\alpha}}{\left(\alpha-1\right)\left(m+1\right)^{\alpha-1}\mu}\cdot E\left[\left(X_1^{+}\right)^{2+\varepsilon}\right]\leq 1. \tag{A.1.2}$$

Next, for (2.3.7), we optimize over z and obtain

$$\frac{z}{\left(m+z\right)^{2(1-\delta)}}\leq\frac{1}{m^{1-2\delta}}\cdot\frac{\left(1-2\delta\right)^{1-2\delta}}{\left(2\left(1-\delta\right)\right)^{2(1-\delta)}}, \tag{A.1.3}$$

for all $\delta \in (0,1/2]$. Use Chebyshev's inequality, together with (A.1.3), and the change of variable $u=\gamma^{1/\delta}(m+z)$ to obtain

$$\begin{aligned} &\frac{3\left(1+2z+m\right)^{\alpha}}{\left(\alpha-1\right)\left(m+1\right)^{\alpha-1}z}\exp\left(-\gamma\left(m+z\right)^{\delta}+\frac{\gamma^{2}e^{\gamma}E\left(X^{2}\right)z}{\left(m+z\right)^{2(1-\delta)}\mu}+4\frac{z}{\mu}P\left(X>\left(z+m\right)^{1-\delta}\right)\right)\leq \\ &\leq\frac{3\left(1+2z+m\right)^{\alpha}}{\left(\alpha-1\right)\left(m+1\right)^{\alpha-1}z}\exp\left(-\gamma\left(m+z\right)^{\delta}+\frac{(\gamma^{2}e^{\gamma}+4)E\left(X^{2}\right)\left(1-2\delta\right)^{1-2\delta}}{\left(2\left(1-\delta\right)\right)^{2(1-\delta)}\mu m^{1-2\delta}}\right) \\ &\leq\frac{3\cdot 2^{\alpha}\gamma^{-\alpha/\delta}}{\left(\alpha-1\right)\left(m+1\right)^{\alpha-1}\mu}\exp\left(\frac{(\gamma^{2}e^{\gamma}+4)E\left(X^{2}\right)\left(1-2\delta\right)^{1-2\delta}}{\left(2\left(1-\delta\right)\right)^{2(1-\delta)}\mu m^{1-2\delta}}\right)\max_{u\geq\gamma^{1/\delta}m}u^{\alpha}\exp\left(-u^{\delta}\right). \end{aligned}$$

Thus, we can first select $\gamma=1$, for example, and then pick the smallest m so that

$$\frac{3\cdot 2^{\alpha}}{\left(\alpha-1\right)\left(m+1\right)^{\alpha-1}\mu}\exp\left(\frac{7E\left(X^{2}\right)\left(1-2\delta\right)^{1-2\delta}}{\left(2\left(1-\delta\right)\right)^{2(1-\delta)}\mu m^{1-2\delta}}\right)\max_{u\geq\gamma^{1/\delta}m}u^{\alpha}\exp\left(-u^{\delta}\right)\leq 1. \tag{A.1.4}$$

This can be done numerically or, explicitly by simply by noting (using elementary calculus) that

$$\max_{u\geq\gamma^{1/\delta}m}u^{\alpha}\exp\left(-u^{\delta}\right)\leq\left(\frac{\alpha}{\delta}\right)^{\alpha}\exp\left(-\left(\frac{\alpha}{\delta}\right)^{\delta}\right).$$

In the numerical examples that we will discuss in Section 2.6 we noted that the performance of the algorithm is not too sensitive to the selection of α, and thus we advocate picking α somewhat larger than 2, for instance $\alpha \in (2,4]$, but it is important to constrain α and δ so that $z^{\alpha}P\left(X>z^{1-\delta}\right)=O(1)$, due to (2.3.6).

It is constraint (2.3.7) the one that has the highest impact in the algorithm's performance and we noted that the selection of m, in particular, was the most relevant parameter.

So, we simply used the Excel solver; given our selection of α we picked $\delta \in \left(0, \frac{1}{2}\right]$, $\gamma \geq 0$ and $m \geq 0$ so as to minimize m subject to (2.3.6) and (2.3.7). The optimization is done only once and it took a second.

In Section 2.6 we also argued that the running time of our algorithm is close to the relaxation time of the Markov chain from a heavy-traffic perspective.

A.2 Proof of Lemma 3

Proof. Notice that

$$\begin{aligned} P\left(A_k\right) &\leq \sum_{j=n_{k-1}}^{n_k-1} P\left(X_j > (j\mu + m)^{1-\delta}\right) \\ &\leq n_k P\left(X_1 > (n_{k-1}\mu + m)^{1-\delta}\right). \end{aligned}$$

It is straightforward to verify (using Chebyshev's inequality, the fact that $E\left|X_1\right|^{\beta} < \infty$ for $\beta > 1$ and the definition of n_k) that for any $\delta > 0$,

$$\sum_k n_k P\left(X_1 > (n_{k-1}\mu + m)^{1-\delta}\right) < \infty.$$

Now we have for $k \geq 2$

$$\begin{aligned} \frac{3P\left(A_k\right)}{g\left(k\right)} &\leq 3\bar{G}\left(m\right) \frac{n_k P(X_1 > (n_{k-1}\mu + m)^{1-\delta})}{\int_{m+\mu n_{k-1}}^{m+\mu n_k} P\left(Y > s\right) ds} && \text{(A.2.1)} \\ &\leq 3\bar{G}\left(m\right) \frac{n_k P(X_1 > (\mu n_{k-1} + m)^{1-\delta})}{\mu n_{k-1} P\left(Y > m + n_k\right)} && \text{(A.2.2)} \\ &= 6\bar{G}\left(m\right) \frac{P(X_1^+ > (\mu n_{k-1} + m)^{1-\delta})}{\mu P\left(Y > m + \mu n_k\right)} && \text{(A.2.3)} \\ &\leq \frac{6\left(1 + 2\mu n_{k-1} + m\right)^{\alpha}}{\left(\alpha - 1\right)\left(m + 1\right)^{\alpha-1}\mu} P\left(X > (\mu n_{k-1} + m)^{1-\delta}\right) && \text{(A.2.4)} \\ &\leq 1 && \text{(A.2.5)} \end{aligned}$$

Making $z = \mu n_{k-1} = \mu 2^{k-2}$ and using (2.3.6) we obtain the conclusion of the lemma. $\square$

A.3 Proof of Lemma 4

Before we prove Lemma 4, we will first introduce an auxiliary lemma, which will be proved at the end of this section.

Lemma 20. *Set* $\theta = \gamma / u^{1-\delta}$ for $\delta \in (0,1)$, $u, \gamma > 0$ *and* suppose that $E(X) = 0$. *If* $E\left(|X|^{1+\varepsilon}\right) < \infty$ *for some* $\varepsilon \in (0,1)$ *and*

$$\frac{E\left(|X|^{1+\varepsilon}\right)}{u^{(1-\delta)(1+\varepsilon)}} \leq \frac{1}{2}, \tag{A.3.1}$$

then

$$E\left[\exp(\theta X) \,\middle|\, X \leq u^{1-\delta}\right] \leq \exp\left\{\frac{A}{u^{(1-\delta)(1+\varepsilon)}}\right\} \tag{A.3.2}$$

with

$$A = \left(\frac{\gamma^2}{2} \cdot \frac{\exp(\gamma)}{1-\varepsilon} + 2\right) \cdot E\left(|X|^{1+\varepsilon}\right). \tag{A.3.3}$$

Moreover, if $E\left(X^2\right) < \infty$ *and*

$$\frac{E\left(X^2\right)}{u^{2(1-\delta)}} \leq \frac{1}{2} \tag{A.3.4}$$

then

$$E\left[\exp(\theta X) \,\middle|\, X \leq u^{1-\delta}\right] \leq \exp\left(\frac{\gamma^2 \exp(\gamma) E(X^2)}{2u^{2(1-\delta)}} + 2P\left(X > u^{1-\delta}\right)\right) \tag{A.3.5}$$

$$\leq \exp\left\{\frac{A}{u^{2(1-\delta)}}\right\}, \tag{A.3.6}$$

with

$$A = \left(\frac{\gamma^2 \exp(\gamma)}{2} + 2\right) \cdot E\left(X^2\right). \tag{A.3.7}$$

If in addition $u \geq 1$ *and* $0 < \delta \leq \varepsilon/2$ *then from (A.3.2) we obtain*

$$E\left[\exp(\theta X) \,\middle|\, X \leq u^{1-\delta}\right] \leq \exp\left(\frac{A}{u}\right), \tag{A.3.8}$$

and if $EX^2 < \infty$ *inequality (A.3.8) follows from (A.3.6) choosing* $0 \leq \delta \leq 1/2$.

Having Lemma 20 at hand we are now ready to prove Lemma 4

Proof of Lemma 4. Since $m \geq 1$ satisfies inequality (2.3.5), then we can invoke Lemma 20 with $u = n_{k-1}\mu + m = C_k$ and obtain

$$\exp(\psi_k(\theta_k)) \leq \exp\left(\frac{\gamma^2 \exp(\gamma) E(X^2)}{2C_k^{2(1-\delta)}} + 2P\left(X > C_k^{1-\delta}\right)\right). \tag{A.3.9}$$

By definition of T_m we have that $S_{T_m} \geq \mu T_m + m$, and because $T_m \in [n_{k-1}, n_k - 1]$ we conclude that

$$S_{T_m} \geq \mu n_{k-1} + m = C_k.$$

Therefore, on $T_m \in [n_{k-1}, n_k - 1]$

$$\exp(-\theta_k S_{T_m} + T_m \psi_k(\theta_k)) \leq \exp(-\theta_k C_k + n_k \psi_k(\theta_k)). \tag{A.3.10}$$

Combining (A.3.9) and (A.3.10), and letting $z = \mu n_{k-1}$, we obtain that

$$\begin{aligned} &\exp(-\theta_k S_{T_m} + T_m \psi_k(\theta_k)) \leq \\ &\leq \exp\left(-\gamma(\mu n_{k-1} + m)^{\delta} + \frac{\gamma^2 \exp(\gamma) E(X^2) n_{k-1}}{(\mu n_{k-1} + m)^{2(1-\delta)}} + 2 n_k P\left(X > (\mu n_{k-1} + m)^{(1-\delta)}\right)\right) \\ &= \exp\left(-\gamma(z+m)^{\delta} + \frac{\gamma^2 \exp(\gamma) E(X^2) z}{(z+m)^{2(1-\delta)} \mu} + 4\frac{z}{\mu} P\left(X > (z+m)^{(1-\delta)}\right)\right). \end{aligned}$$

Therefore, using (2.3.7) we conclude that

$$\begin{aligned} &\frac{3\exp(-\theta_k S_{T_m} + T_m \psi_k(\theta_k))}{g(k)} \leq \\ &\leq \frac{3(1+2z+m)^{\alpha}}{(\alpha-1)(m+1)^{\alpha-1} z} \exp\left(-\gamma(z+m)^{\delta} + \frac{\gamma^2 \exp(\gamma) E(X^2) z}{(z+m)^{2(1-\delta)} \mu} + 4\frac{z}{\mu} P\left(X > (z+m)^{(1-\delta)}\right)\right) \\ &\leq 1, \end{aligned}$$

thereby obtaining the result. $\square$

We conclude this appendix with the proof of the auxiliary lemma.

Proof of Lemma 20. Since $EX = 0$, $E[XI\left(X \leq u^{1-\delta}\right)] < 0$, and therefore a Taylor expansion of second order yields

$$E\left[\exp\left\{X\frac{\gamma}{u^{1-\delta}}\right\}, X \leq u^{1-\delta}\right] \leq 1 + \frac{\gamma^2}{2} E\left[\left(\frac{X}{u^{1-\delta}}\right)^2 \exp\left\{\frac{\gamma X}{u^{1-\delta}}\right\} I\left(X \leq u^{1-\delta}\right)\right]$$

If $EX^2 < \infty$, we conclude that

$$E\left[\exp\left\{X\frac{\gamma}{u^{1-\delta}}\right\}, X \leq u^{1-\delta}\right] \leq 1 + \frac{\gamma^2 \exp(\gamma)}{2} \cdot E(X^2) \cdot \frac{1}{u^{2(1-\delta)}}.$$

Since $1+x \le \exp(x)$ for $x \ge 0$ we conclude that

$$E\left[\exp\left\{X\frac{\gamma}{u^{1-\delta}}\right\}, X \le u^{1-\delta}\right] \le \exp\left(\frac{\gamma^2\exp(\gamma)}{2}\cdot E(X^2)\cdot\frac{1}{u^{2(1-\delta)}}\right).$$

On the other hand

$$P\left(X \le u^{1-\delta}\right) = 1 - P\left(X > u^{1-\delta}\right) \ge 1 - \frac{E(X^2)}{u^{2(1-\delta)}}$$

and since $1-x \ge \exp(-2x)$ for $x \in \left(0, \frac{1}{2}\right)$ we conclude that if (A.3.4) holds then

$$E\left[\exp\left\{X\frac{\gamma}{u^{1-\delta}}\right\} \,\middle|\, X \le u^{1-\delta}\right] \le \exp\left(\frac{\gamma^2\exp(\gamma)\,E(X^2)}{2u^{2(1-\delta)}} + 2P\left(X > u^{1-\delta}\right)\right),$$

which yields (A.3.6).

Now, let's assume that $\varepsilon \in (0,1)$ and $E|X|^{1+\varepsilon} < \infty$. Since $z^2\exp(-z) \le 4\exp(-2) < 1$ for $z \ge 0$ we have that

$$\begin{aligned} &E\left[\left(\frac{X\gamma}{u^{1-\delta}}\right)^2 \exp\left\{\frac{X\gamma}{u^{1-\delta}}\right\} I\left(X \le u^{1-\delta}\right)\right] \le \\ &\le \gamma^2\exp(\gamma)\,E\left[\left(\frac{X}{u^{1-\delta}}\right)^2 I\left(|X| \le u^{1-\delta}\right)\right] + P\left(X < -u^{1-\delta}\right). \end{aligned}$$

In addition,

$$\begin{aligned} E\left[|X|^2 I\left(|X| \le u^{1-\delta}\right)\right] &= 2E\left[\int_0^{u^{1-\delta}} sI\left(|X| > s\right) ds\right] && \text{(A.3.11)} \\ &= 2\int_0^{u^{1-\delta}} sP\left(|X| > s\right) ds && \text{(A.3.12)} \\ &\le \frac{E|X|^{1+\varepsilon}}{1-\varepsilon} u^{(1-\varepsilon)(1-\delta)} && \text{(A.3.13)} \end{aligned}$$

Therefore,

$$E\left[\left(\frac{X}{u^{1-\delta}}\right)^2 I\left(|X| \le u^{1-\delta}\right)\right] \le \frac{E|X|^{1+\varepsilon}}{1-\varepsilon}\cdot\frac{1}{u^{(1+\varepsilon)(1-\delta)}}.$$

Since

$$P\left(X < -u^{1-\delta}\right) \le \frac{E|X|^{1+\varepsilon}}{u^{(1+\varepsilon)(1-\delta)}},$$

we conclude combining (A.3.13) and (A.3) that

$$E\left[\exp\left\{X\frac{\gamma}{u^{1-\delta}}\right\}, X \leq u^{1-\delta}\right] \leq 1+\frac{\gamma^2}{2}\cdot E\left|X\right|^{1+\varepsilon}\cdot\left(\frac{\exp\left(\gamma\right)}{\left(1-\varepsilon\right)}+1\right)\cdot\frac{1}{u^{(1+\varepsilon)(1-\delta)}} \tag{A.3.14}$$

$$\leq 1+\gamma^2\cdot E\left|X\right|^{1+\varepsilon}\cdot\frac{\exp\left(\gamma\right)}{\left(1-\varepsilon\right)}\cdot\frac{1}{u^{(1+\varepsilon)(1-\delta)}}. \tag{A.3.15}$$

Similarly to the finite variance case we conclude that if (A.3.4) holds, then

$$E\left[\exp\left\{X\frac{\gamma}{u^{1-\delta}}\right\}\,\middle|\, X \leq u^{1-\delta}\right] \leq$$
$$\leq \exp\left(\gamma^2\cdot E\left|X\right|^{1+\varepsilon}\cdot\frac{\exp\left(\gamma\right)}{\left(1-\varepsilon\right)}\cdot\frac{1}{u^{(1+\varepsilon)(1-\delta)}}+2E\left|X\right|^{1+\varepsilon}\cdot\frac{1}{u^{(1+\varepsilon)(1-\delta)}}\right),$$

which in turn yields (A.3.2). The last part of the result, namely (A.3.8) follows from elementary algebra and the fact that we are requiring $u \geq 1$. □

A.4 Proof of Lemma 5

Proof. Notice that

$$\begin{aligned} P\left(B_k^c\right) &\leq \sum_{j=n_{k-1}}^{n_k-1} P\left(X_j > \left(j\mu+m\right)^{1-\delta}\right) \\ &\leq n_k P\left(X_1 > \left(n_{k-1}\mu+m\right)^{1-\delta}\right). \end{aligned}$$

Now we can continue and apply the same arguments as in Lemma 3 to conclude the proof. □

Appendix B

Appendix for Chapter 4

B.1 Proof of Lemma 15

The goal of this section is to establish the result in Lemma 15, which is stated again below

Lemma (Lemma 15). *Under the assumptions as in Theorem 14. There exists a constant* $K = K(d, Q)$ *that depends only on the dimension* d *and the function* Q, *such that for every* $\rho \in (0, 1)$ *and* $x \in G$ *we have*

$$E_x \left(\exp \left\{ (1-\rho) \lambda(G) \tau_G \right\} \right) \leq \frac{K(d, Q)}{\rho^{1+d/2}} \tag{B.1.1}$$

In order to prove Lemma15 we will need the following two results

Lemma 21. *Under the assumptions as in Theorem 14.*

$$P_x(\tau_G > t) \leq \frac{1}{\delta^2} E_x \left(\exp \left\{ - \int\limits_0^t V(B(s))\, ds,\ \tau_G > t \right\} \right) \tag{B.1.2}$$

where

$$V(x) := \frac{1}{2} e^{-Q(x)} \cdot \triangle e^{Q(x)} \tag{B.1.3}$$

and $(B(t) : , t \geq 0)$ *is the* $d-$*dimensional Brownian motion.*

Lemma 22. *Let* $\tilde{\lambda}_V(G)$ *be the Dirichlet principle eigenvalue of the generator* $\tilde{L} = -\frac{1}{2}\triangle + V$ *that was introduced earlier. Then, under the assumption of Theorem 14, there exists a* *constant* $c(d,Q) \in (1,\infty)$ *that depends only on the dimension and the function* Q *(or alternatively, the function* V*) such that,*

$$\sup_{x\in G} \tilde{E}_x\left(\exp\left\{-\int\limits_0^t V(B(s))\,ds,\ \tau_G > t\right\}\right) \le c(d,V)\left(\left(\tilde{\lambda}_V(G)\,t\right)^{d/2}+1\right)\exp\left\{-\tilde{\lambda}_V(G)\,t\right\} \tag{B.1.4}$$

We will provide the proof of Lemma 21 at the end of this section. The proof of Lemma 22 can be found in [Sznitman, 1998] (Theorem 1.2 page 93).

Remark 23. *Notice that Lemma 15 implies that*

$$\lambda(G) = \tilde{\lambda}_V(G) \tag{B.1.5}$$

B.1.1 Proof of Lemma 15

Proof of Lemma 15. First notice that by using integration by parts, we have for every $\eta > 0$

$$\begin{aligned}\int\limits_0^\infty \eta e^{\eta t} P_x(\tau_D > t)\,dt &= E_x\left[\int\limits_0^\infty \mathbf{1}(\tau_D > t)\,de^{\eta t}\right]\\ &= E_x\left[\int\limits_0^{\tau_D} de^{\eta t}\right]\\ &= E_x\left[e^{\eta\tau_D}-1\right] = E_x\left[e^{\eta\tau_D}\right]-1\end{aligned} \tag{B.1.6}$$

Using integration by parts and Lemma 22 we have,

$$\begin{aligned}&E_x\left(\exp\{(1-\rho)\lambda(G)\tau_G\}\right) =\\ &= 1+\int\limits_0^\infty (1-\rho)\lambda(G)\,e^{(1-\rho)\lambda(G)t}P_x(\tau_G > t)\,dt\\ &\le 1+\tfrac{c(d,V)}{\delta^2}\int\limits_0^\infty (1-\rho)\lambda(G)\left(\left(\tilde{\lambda}_V(G)\,t\right)^{d/2}+1\right)\exp\left\{(1-\rho)\lambda(G)\,t-\tilde{\lambda}_V(G)\,t\right\}\end{aligned} \tag{B.1.7}$$

Now using the fact that $\lambda(G) = \tilde{\lambda}_V(G)$ and making a change in variables we have that the left hand side of equation B.1.7

$$\begin{aligned}&= 1+\tfrac{c(d,V)(1-\rho)}{\delta^2}\int\limits_0^\infty \left(1+u^{d/2}\right)\exp(-\rho u)\,du\\ &= \begin{cases} 1+\tfrac{c(d,V)(1-\rho)}{\delta^2}\left(\tfrac{1}{\rho}+\tfrac{\Gamma\left(1+\frac{d}{2}\right)}{\rho^{1+d/2}}\right) & \text{if } \lambda(G)\neq 0\\ 0 & \text{otherwise}\end{cases}\end{aligned} \tag{B.1.8}$$

□

B.1.2 Proof of Lemma 21

Proof of Lemma 21. Consider the following stochastic differential equation

$$d\tilde{X}(t) = dB(t) - V \tag{B.1.9}$$

where V is as in equation (B.1.3). Notice that the process defined by (B.1.9) is a $d-$dimensional Brownian motion under the influence of a potential $-V$. The corresponding generators to the by

$$\tilde{L} = -\frac{1}{2}\triangle + V \tag{B.1.10}$$

In order to connect the two processes we will invoke Grisanov's Theorem which in this case states that $\forall x \in \bar{G}$

$$\begin{aligned} \tfrac{dP_x}{d\tilde{P}_x}|_{\mathcal{F}_t} &= \exp\left\{\int\limits_0^t \nabla Q(X_s)\, dX_s - \tfrac{1}{2}\int\limits_0^t \|\nabla Q(X_s)\|_2^2\, ds\right\} \\ &=: M_x(t) \end{aligned} \tag{B.1.11}$$

Observe that applying Itô's lemma yields that

$$Q(X_t) - Q(X_0) = \frac{1}{2}\int\limits_0^t \triangle Q(X_s)\, ds + \int\limits_0^t \nabla Q(X_s)\, dX_s \tag{B.1.12}$$

Moreover, notice that,

$$\begin{aligned} e^{-Q(x)} \cdot \triangle e^{Q(x)} &= e^{-Q(x)} \sum_{i=1}^{d} \tfrac{\partial^2}{\partial x_i^2} e^{Q(x)} \\ &= e^{-Q(x)} \sum_{i=1}^{d} \tfrac{\partial}{\partial x_i} Q_{x_i}(x)\, e^{Q(x)} \\ &= e^{-Q(x)} \sum_{i=1}^{d} \left[Q_{x_i x_i}(x)\, e^{Q(x)} + (Q_{x_i}(x))^2 e^{Q(x)}\right] \\ &= \triangle Q(x) + \|\nabla Q(x)\|_2^2 \end{aligned} \tag{B.1.13}$$

Therefore, $M_x(t)$ in (B.1.11) can be written as

$$\begin{aligned} M_x(t) &= \exp\left\{Q(X_t) - Q(X_0) - \tfrac{1}{2}\int\limits_0^t \triangle Q(X_s)\, ds - \tfrac{1}{2}\int\limits_0^t \|\nabla Q(X_s)\|_2^2\, ds\right\} \\ &= \exp\left\{Q(X_t) - Q(X_0) - \int\limits_0^t \tfrac{1}{2} e^{-Q(X_s)} \cdot \triangle e^{Q(X_s)} ds\right\} \end{aligned} \tag{B.1.14}$$

Using the notations,

$$V(X_s) = \frac{1}{2} e^{-Q(X_s)} \cdot \triangle e^{Q(X_s)} \qquad \text{and} \qquad u(X_s) = e^{Q(X_s)} \tag{B.1.15}$$

we obtain that

$$\begin{aligned} \mathbb{P}_x\left(\tau_G > t\right) &= \tilde{E}_x\left[I\left(\tau_G > t\right) M_x\left(t\right)\right] \\ &= \tilde{E}_x\left[I\left(\tau_G > t\right) \tfrac{u(X_s)}{u(X_0)} \exp\left\{-\int\limits_0^t V\left(X_s\right) ds\right\}\right] \end{aligned} \tag{B.1.16}$$

Notice that there exists $\delta\left(G\right) = \delta > 0$ so that

$$\delta \leq e^{Q(x)} \leq \frac{1}{\delta}$$

for all $x \in \bar{G}$. And so we can bound

$$\begin{aligned} \mathbb{P}_x\left(\tau_G > t\right) &\geq \delta^2 \tilde{E}_x\left[I\left(\tau_G > t\right) \exp\left\{-\int\limits_0^t V\left(X_s\right) ds\right\}\right] \\ \mathbb{P}_x\left(\tau_G > t\right) &\leq \tfrac{1}{\delta^2} \tilde{E}_x\left[I\left(\tau_G > t\right) \exp\left\{-\int\limits_0^t V\left(X_s\right) ds\right\}\right] \end{aligned} \tag{B.1.17}$$

□