NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

A SIMULATED ANNEALING ALGORITHM FOR DETECTING MOVING TARGETS

by

Jun Jie Lim

September 2018

Thesis Advisor:	Devaushi I. Singham
Second Reader:	Michael P. Atkinson

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			*Form Approved OMB No. 0704-0188*

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** September 2018	**3. REPORT TYPE AND DATES COVERED** Master's thesis

4. TITLE AND SUBTITLE A SIMULATED ANNEALING ALGORITHM FOR DETECTING MOVING TARGETS	**5. FUNDING NUMBERS**
6. AUTHOR(S) Jun Jie Lim	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A	**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release. Distribution is unlimited.	**12b. DISTRIBUTION CODE** A

13. ABSTRACT (maximum 200 words)

Target tracking and monitoring plays a crucial role in the intelligence collection domain. With the advancement of intelligence collection and data analysis methods, we can sometimes obtain a target's initial and end locations of its desired trajectory, albeit with some uncertainty. Based on such intelligence information, the target's movement can be modeled as a stochastic process using a Brownian bridge, and the target's geographical location probability distribution in time can be aggregated and mapped as a two-dimensional temporal heat map. Based on this model, we search for sensor deployment strategies that maximize the probability of target detection. This thesis adopts a random search method called simulated annealing and customizes it to the unique setting of target tracking to obtain a sensor configuration that approximately maximizes the target detection probability, accounting for uncertainty in intelligence information. To evaluate the performance of the proposed method, we perform an experimental design and compare the results from simulated annealing with a simple heuristic. Based on a drug trafficking scenario, we attempt to find the approximate best sensor configuration to maximize the probability the sensors successfully observing the target, given limited sensor coverage and uncertain intelligence.

14. SUBJECT TERMS simulated annealing, target detection, random search optimization, Brownian bridge, simulation	**15. NUMBER OF PAGES** 92
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
Unclassified	Unclassified	Unclassified	UU

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89) Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release. Distribution is unlimited.

A SIMULATED ANNEALING ALGORITHM FOR DETECTING MOVING TARGETS

Jun Jie Lim
Civilian, Singapore
BSME, Nanyang Technological University, 2012

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
September 2018

Approved by: Devaushi I. Singham
Advisor

Michael P. Atkinson
Second Reader

W. Matthew Carlyle
Chair, Department of Operations Research

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Target tracking and monitoring plays a crucial role in the intelligence collection domain. With the advancement of intelligence collection and data analysis methods, we can sometimes obtain a target's initial and end locations of its desired trajectory, albeit with some uncertainty. Based on such intelligence information, the target's movement can be modeled as a stochastic process using a Brownian bridge, and the target's geographical location probability distribution in time can be aggregated and mapped as a two-dimensional temporal heat map. Based on this model, we search for sensor deployment strategies that maximize the probability of target detection. This thesis adopts a random search method called simulated annealing and customizes it to the unique setting of target tracking to obtain a sensor configuration that approximately maximizes the target detection probability, accounting for uncertainty in intelligence information. To evaluate the performance of the proposed method, we perform an experimental design and compare the results from simulated annealing with a simple heuristic. Based on a drug trafficking scenario, we attempt to find the approximate best sensor configuration to maximize the probability the sensors successfully observing the target, given limited sensor coverage and uncertain intelligence.

THIS PAGE INTENTIONALLY LEFT BLANK

Table of Contents

List of Figures

List of Tables

THIS PAGE INTENTIONALLY LEFT BLANK

List of Acronyms and Abbreviations

AAS	Annealing Adaptive Search
AOI	area of interest
BB	Brownian bridge
BBMM	Brownian Bridge Movement Model
BM	Brownian motion
CD	Coordinate Direction
CI	Confidence Interval
HAS	Hesitant Adaptive Search
HD	Hyperspherical Direction
NOLH	Nearly Orthogonal Latin Hypercube
PAS	Pure Adaptive Search
PRS	Pure Random Search
SA	simulated annealing
TBBM	Temporal Brownian Bridge Model

THIS PAGE INTENTIONALLY LEFT BLANK

Executive Summary

Target tracking plays a crucial role in the intelligence collection domain. In this context, one of the possible objectives is to maximize the probability of target detection by carefully choosing the sensor placement configuration. Given the incomplete information about moving targets, an effective way to deploy sensors can be determined using a simulation approach. Simulation methods such as the Brownian Bridge Movement Model (BBMM) are able to generate probability heat maps that represent randomness of a target's location. The BBMM is calibrated using the target start and end positions with a Brownian bridge assumption for target motion, and the dispersion is controlled by a variance parameter related to the target's mobility (Horne et al. 2003). Since the optimal sensor placement usually cannot be determined analytically, this thesis focuses on the simulated annealing (SA) algorithm, a random search algorithm that is able to analyze the probability heat map generated and optimize sensor placement to maximize target detection probability.

There are two levels of stochasticity in the problem. First, a simulation model evaluates the objective function, which is the probability of detection. The second is that a random search optimization method samples candidates for the optimal solution. This thesis models a target's geographical location and movement in time as two-dimensional heat maps given uncertain intelligence using the Temporal Brownian Bridge Model (TBBM) as originally described by Cheng (2016). This is a modification of the BBMM to model changing heat maps over time. Next, using the TBBM as the underlying stochastic model, the thesis employs a modified SA algorithm on the TBBM to determine the optimal sensor placement strategy prior to the beginning of the target's journey, thus maximizing the overall probability of target detection.

The SA algorithm was inspired by the annealing process of metal, where metal starts out at a high initial temperature and slowly cools to form a new lattice structure. Similarly, the SA algorithm uses a temperature parameter to control its optimization process. The cooling schedule of the temperature parameter is the key aspect of SA as compared to other search algorithms, and allows the SA algorithm to accept non-improving points during its search for the optimal objective function with some probability. During the initial iterations when the temperature parameter is high, the algorithm readily accepts non-improving points to

explore the whole sample space. The temperature parameter is slowly decreased over the iterations according to the cooling schedule and the algorithm reduces the probability of accepting non-improving points. The key challenges of implementing SA methods are that the cooling schedules are almost always heuristic, and since the cooling schedules are specific to each problem in hand, this thesis uses a modified version of the SA algorithm applied to the TBBM to optimize sensor deployment configuration given a target detection scenario.

Through experimental design and analysis, the thesis evaluates the performance of the SA algorithm against a heuristic method of sensor placement and analyzes the computational run-time of the algorithm. The heuristic approach centers the sensors at the peaks of the probability heat map, assuming that the area around each peak is also surrounded by areas with relativity high probability of detection. The results of the heuristic approach serve as the theoretical lower bounds to the optimization solution, which also provides a baseline against which the SA algorithm can be compared. Five factors, namely the number of sensors, the variance parameter, sensor sizes, the number of possible paths made by the target, and the time of sensor placement are varied. The factors are varied using a 65 design point Nearly Orthogonal Latin Hypercube (NOLH) (Sanchez 2011), with four repetitions at each design point, to evaluate the performance of the SA algorithm.

The experimental results show that during cases where the variance parameter is small and the sensor size is large, the SA algorithm tends to outperform the heuristic approach. Because of the effects of large sensor sizes covering large areas of the heat map combined with a small uncertainty spread of the heat map, the SA algorithm is able to detect areas with high probability more efficiently. The performance of the SA algorithm is comparable, or sub-optimal, compared to the heuristic approach when the variance is large and sensor size is small. A smooth and flat heat map probability surface results in the combination of small detection areas and large uncertainties in target position, Therefore, the SA algorithm is not able to converge to the global optimal efficiently. In another variant of the SA algorithm, the algorithm is initialized with heuristic results as the starting point instead of using a random initialization. The experimental results show that the overall performance of the SA algorithm is the same despite the additional information input.

In terms of computational effort, the normalized computational run-time for the SA algo-

rithm is linearly proportional to the number of sensors in the scenario. As the candidate solution generator in the SA algorithm that searches and generates the next random placement for each sensor is the most computationally intensive portion of the algorithm, it dictates the main processing time of the algorithm.

Based on a narcotics smuggling scenario in Central America, the SA algorithm optimizes the sensor placement strategy for four different scenarios with the number of sensors ranging from one to four, limited by a fixed amount of total search area effort. Explicitly, the sensor size is equal to the total search area when the number of sensors is one, and is a quarter of the total search area when the total number of sensors is four. A modified loss function is used to find the robust solution, given the uncertainty of the variance parameter of the BBMM. Results show that the most robust solution is to deploy two separate sensors, splitting the overall sensor area in half to maximize probability of detection of the target.

This research demonstrates that the application of the SA algorithm is able to produce improved solutions over heuristic methods for most of the target detection scenarios. Nevertheless, to obtain the same results with less computational effort, the heuristic approach is recommended when the variance parameter is high and sensor sizes are small. The thesis also recommends incorporating additional intelligence input, such as the sensor's probability of detection and sensor live feedback, for future research. This thesis works as a basis for future real-time sensor placement in this field of interest.

References

Cheng CC (2016) *A Brownian bridge movement model to track mobile targets.* Master's thesis, Department of Operations Research, Naval Postgraduate School, Monterey, CA, https://calhoun.nps.edu/handle/10945/50522.

Horne JS, Garton EO, Krone SM, Lewis JS (2007) Analyzing animal movements using Brownian bridges. *Ecology* 88(9):2354–2363, http://www.jstor.org/stable/27651373.

Sanchez S (2011) NOLH design spreadsheet. Accessed May 20, 2018, http://harvest.nps.edu/.

THIS PAGE INTENTIONALLY LEFT BLANK

Acknowledgments

I am most grateful to my thesis advisor, Professor Dashi Singham, who has been exceptionally patient and helpful throughout the writing of this thesis. Her guidance and assistance in terms of coding, writing and data analyzing steered me to the correct direction whenever I needed it.

I also thank Professor Michael Atkinson as the second reader for this thesis. His valuable input and ideas increased the depth and quality of this work.

Lastly, I would like to thank the Naval Postgraduate School Department of Operations Research, which provided me with the necessary tools and knowledge to complete this thesis.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 1:
Introduction

Target tracking and monitoring plays a crucial role in the intelligence collection domain. In the context of search and detection, one of the possible objectives is to maximize the probability of target detection by carefully choosing the sensor placement configuration. Incomplete information about a moving target, however, results in a large combination of possible sensor placement strategies. Thus, it is difficult to obtain a best solution mathematically that can achieve the optimal sensor placement to obtain a maximized target detection probability. The purpose of the thesis is to show how a particular random search algorithm known as the simulated annealing (SA) algorithm can be used for such optimization problems, specifically solving the sensor placement problem for target detection. Given intelligence about a target, we can describe and model the target's trajectory as probabilistic movement dynamics. The thesis provides a discussion on the SA algorithm's application specifically by simulating target movement to produce estimates of the probability distribution of target, and maximizing target detection rate.

1.1 Search and Detection

Search and detection of targets falls in the broad category of search theory. According to Nunn (1981), the history of search theory reaches back to World War II with Bernard Koopman and his colleagues in the Anti-Submarine Warfare Operations Research Group of the U.S. Navy. Since then, search theory has been researched extensively. For reference, Washburn (2014) provides a broad overview and many mathematical models to determine the probability of detecting a target.

Other experts in this research field have established numerous methodologies which are applicable in various scenarios. Some of them deal with moving targets which are applied to scenarios such as counter-smuggling (Campos 2014; Pietz and Royset 2013), anti-submarine warfare (Ben-Yoash 2016) and search and rescue (Macwan et al. 2011). In general, most of these search and detection models makes certain assumptions on the movement of the targets. The targets sometimes are assumed to move in constant speed (Ben-Yoash 2016), or are assumed to have uncertain starting and ending points (Pietz and Royset 2013), or in

the simplest case, are assumed to be hiding in various locations (Chew 1973).

For instance, the work of Campos (2014), created a conversion algorithm that can convert the heat maps generated from the probability model into inputs to an optimization model. His model works with intelligence inputs such as waypoint locations, departure times, target velocity, and other environmental factors to estimate the target location and it's corresponding search plan. In other research, Ben-Yoash (2016), studied ways that helicopters could be deployed to search for submarines in anti-submarine warfare missions. His model assumes that the target's initial location is known, and has increasing area of uncertainty as the target escapes at a constant speed while remaining undetected. The research also uses a mathematical model to provide the optimal payload of the helicopter to strike a balance between search effectiveness and offensive capability. Pietz and Royset (2013) introduced a specific search problem, termed the Smuggler Search Problem, that can be modeled as a Mixed Integer Non-Linear Program. It creates an optimal search plan for a single sensor using the branch-and-bound method to detect and interdict multiple moving targets in an area of interest bounded by a limited amount of search effort. The model also assumes that the targets are linear in motion with constant speed, while allowing uncertainty in the target's starting and ending location and time. The research further enhanced the model to capture complicated target movement behavior and develop search plans for scenarios with multiple searchers; however, the optimization run-time increases quickly as the number of searchers increases.

The approach in this thesis is to model target movement that can depict the probability density of a target's location over a time period of travel given uncertain and incomplete information. Using the model as a basis, we calculate the probability of detection of the target given a sensor deployment configuration. Henceforth, we determine the best sensor strategy to maximize target detection probability.

1.2 Model Overview

The main problem that the thesis addresses is the optimization of sensor placement to detect moving targets with unknown trajectories. The first area of uncertainty is target trajectory prediction due to limited target information. This field of research is not new in the academic realm. For instance, Raichlen et al. (2014) used Levy Walks to explore the pattern of human

foraging; Mooshegian (2013) used epi-spline approximation to estimate the movement of sea vehicles, and Horne et al. (2007) used a Brownian Bridge Movement Model (BBMM) to predict animal migration patterns. The BBMM aggregates multiple Brownian bridges (BBs) over time to produce a normalized heat map, where the Brownian bridges model animal migration patterns which follow a general direction, but the particular trajectory is not exactly known. Due to the BBMM's unique ability to model target trajectories based on starting and ending conditions and a variance parameter, this method has been widely used in various fields of studies, for instance, those on the migration pattern of black bears (Horne et al. 2007) and mule deer (Bunnefeld et al. 2011).

The BBMM is governed by an underlying continuous-time stochastic process, and is obtained by aggregating multiple BBs between two waypoints. These BBs are Brownian motions (BMs) processes that have their start and end points tied down to some fixed points. In this context, the BB uses the target's known start and end points to model the movement of the target between the two points. The simulation of the BBMM generates a two-dimensional heat map that describes the probability distribution of the target location. Cheng (2016) and Nunez (2017) established a methodology, called the Temporal Brownian Bridge Model (TBBM), that models a temporal heat map of target movement based on starting and ending points. Different from the conventional BBMM that produces a distribution heat map that is averaged over time, the TBBM have different target distributions at different time points. In Cheng's research, he employed the TBBM to generate heat maps of the movement of aircraft carriers in the South China Sea. Nunez further enhanced the heat map generator by introducing particle filtering techniques to include intelligence updates based on sensor inputs. The TBBM also benefits from its modeling simplicity, as it only requires minimal information such as start and end points of the target to model the target trajectory, though more information can be included if available. The thesis uses the TBBM heat map, which is based on the dynamics of the BBMM to model the target's movement.

As the generation of the locational distribution in the model is stochastic, the search for the best sensor placement configuration has a large solution space. Using the heat map obtained from the TBBM that depicts the probability mass of target location, this thesis adopts a random search optimization algorithm, specifically the SA algorithm, to obtain an approximate globally optimal sensor placement strategy that maximizes target detection probability.

Other random search methods such as the cross-entropy method and genetic algorithm can obtain an optimal solution stochastically as well. The cross-entropy method is an iterative process where the algorithm generates new possible points and evaluates them using the measured cross-entropy during every iteration (Kroese et al. 2006). The measurement alters the parameters of the algorithm in the next iteration, and the iteration process repeats. As the iteration count increases, the obtained points from each iteration will converge to the optimal solution. By comparison, the genetic algorithm selects data from a population and develops new generations of results based on a selection process (Goldberg 1989). In the selection process, the algorithm selects the data in the population that have high performance ratings for merging and to create new data (offspring). The new data is then added back into the population. The genetic algorithm process obtains the optimal solution when the iteration process produces no new offspring significantly different from the population. The SA algorithm is a random search method that prefers exploration to exploitation. During every iteration, the SA algorithm provides a random choice among possible options with probability proportionate to the quality of the option (Kindl and Rowe 2012). This allows the optimization to explore a large solution space, and is less likely to be trapped in a local optimum too quickly. This thesis primarily uses the SA algorithm to search for the best sensor placement in a specific scenario of target detection because of the algorithm's ease of implementation and modification. Also, this research uses the SA algorithm because it can solve global optimization problems over a continuous space. It does not require gradient estimations like other methods and has theoretical convergence properties to reach optimality.

1.3 Scenario Overview

This thesis uses a scenario based on the maritime narcotics trafficking situation in the Eastern Pacific region near the Caribbean, shown in Figure 1.1. Drug trafficking, money laundering and violence have plagued the Caribbean region since the 1980s (United States Senate Caucus on International Narcotics Control 102nd Congress 2nd session 2012). Although less than five percent of the cocaine destined for the United States flows through the Caribbean, the United States must be prepared and anticipate the other security issues that arise due to the illicit activities in the region.

In 2012, the United States Senate Caucus on International Narcotics Control (also known as

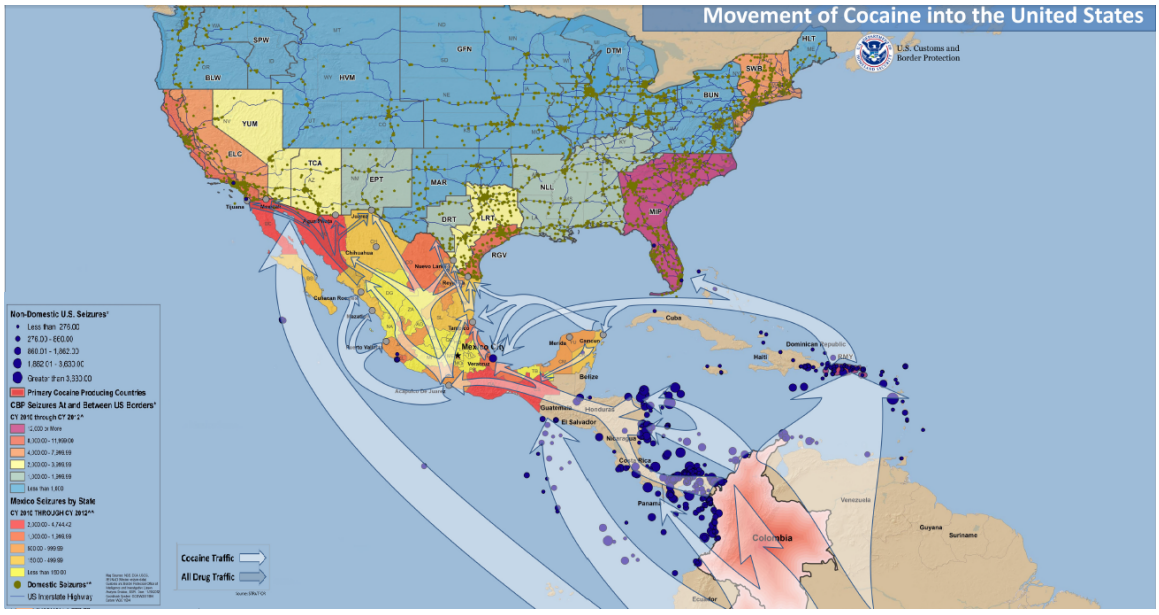


Figure 1.1. Drug trafficking routes out of South America into U.S. through the Caribbean (2012). Source: Borderland Beat (2014).

the Senate Narcotics Caucus) released a report that detailed the situation in the Caribbean (United States Senate Caucus on International Narcotics Control 102nd Congress 2nd session 2012). Collaborative efforts between the United States and neighboring nations in the region have forced some traffickers out of the Caribbean region; however, drug trafficking problems continue to distress Caribbean nations and the United States. Despite the investments made through various policies throughout the years, drug-related violence continues to rise.

Recent interdiction efforts in the region have deterred the use of conventional small aircraft as a means of transportation (United States Senate Caucus on International Narcotics Control 102nd Congress 2nd session 2012). Traffickers now use a variety of surface vehicles, such as speed boats, private crafts, fishing and cargo container vessels, to smuggle drugs across borders instead. The suspected maritime activity has increased from 80 percent in 2012 up to 93 percent in 2014, as shown in Figure 1.2. With this information in hand, there should be an efficient way to model traffickers' movements and create a corresponding search plan to improve the probability of detecting the traffickers.

Figure 1.2. Suspected maritime drug transit through the Caribbean and Central America (2014). Source: Isacson (2017).

Suppose there is available intelligence for the starting and ending locations of the traffickers, and their movement has large variability between the start and end locations, as shown in Figure 1.2, which plots data from possible past target trajectories. Since the TBBM is categorized by the start and end location as model input, and can produce a heat map that models the trajectories in Figure 1.2, we propose using TBBM to model trafficker movements such as the scenario derived from Figure 1.2. An example of such temporal heat maps is in Figure 1.3. The color represents the relative probility of finding the target at the location, and we can observe the changes in the heat map over time.

Then, we use the SA algorithm to determine when and where to place sensors to maximize probability of target detection.

1.4 Scope

The thesis focuses on sensor placement to maximize target detection probability using simulation-based optimization methods. There are two levels of stochasticity involved

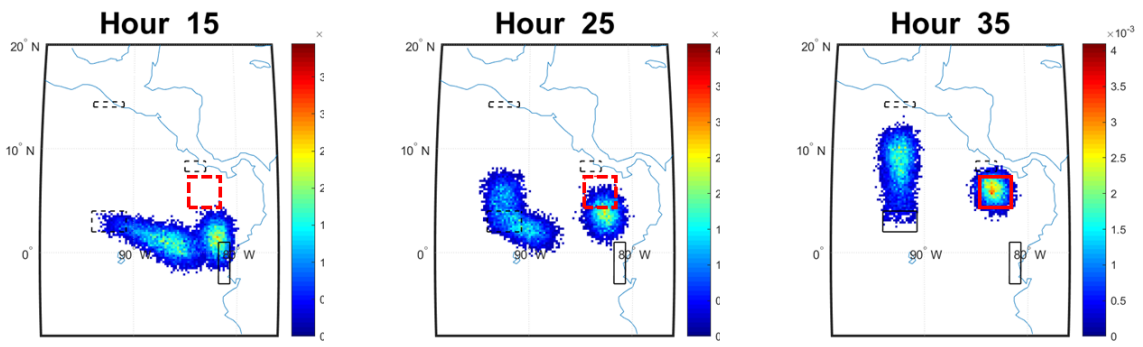


Figure 1.3. Probability heat map of detection generated from TBBM, snapshots at $t = 15, 25, 35$ hours.

The black boxes represent the waypoints of the target. The red box represents the sensor that is deployed over the area of interest.

in this problem. First, a simulation model evaluates the objective function, which is the overall probability of detection. Second, a random search optimization method samples and evaluates the optimal solution. Based on the location probability heat map over the period of travel, the thesis implements SA techniques to find an approximate global optimum solution for the sensor positions, accounting for error in intelligence information. We intend to show that using this approach to place sensors for a realistic scenario performs better than a heruistic method. The heuristic is a naive placement method that centers the sensors at the peak of the probability of detection. Since the solution from the heuristic method is feasible and is in the same feasible region, it can be used as a benchmark to evaluate the performance of the SA algorithm.

This research begins by focusing on the static optimization of sensors. The goal is to optimize all sensor placements prior to the beginning of the target's journey, without input based on ongoing information. The SA algorithm optimizes a combination of different sensors to realize the best strategy that can maximize the overall detection probability of a given scenario. The results can act as a framework for choosing where to place sensors given minimal information on a target.

1.5 Research Questions

The intent of the thesis is to develop a comprehensive framework for choosing where to place sensors when only minimal and uncertain information about a target's movement is available. The model uses the TBBM to generate heat maps to estimate the target's location distribution, and thereafter, implements a global optimization algorithm using the SA random search algorithm to optimize sensor placement and the maximize target detection rate. The thesis attempts to answer the following questions:

1. With limited sensor resources and uncertain intelligence, what methods can determine where to deploy the sensors to maximize the probability of target detection?
2. How can employing random search algorithms, such as simulated annealing, obtain improved estimates of the optimal sensor configuration, as compared with heuristic methods, in this unique setting of target tracking?

1.6 Benefits of the Thesis

The thesis benefits and extends the current research on searching for a moving target given highly uncertain target behavior and target location. Past work on the TBBM focuses more on the refinement of the movement model, and only incorporated sensors without optimizing their placement. The development of an algorithm for sensor placement would benefit many other probabilistic movement models aside from the BBMM and TBBM.

The results from this thesis show that solutions generated from the SA algorithm usually perform better than the heuristic method in the context of target detection and tracking. There are many potential applications for this work that are not limited to the target detection scenario described in Section 1.3. This includes other scenarios, such as searching for missing objects or anticipating pirate attack routes or planned attacks from terrorists. The implementation of this optimization can find the best strategy to maximize probability of detection with a minimum amount of resource input.

1.7 Methodology

The thesis begins with a review of Brownian motion, Brownian bridges and the Temporal Brownian Bridge Model in Chapter 2, and demonstrates how the TBBM can simulate and

model scenarios to produce probability heat maps. The resultant probability heat map of the target's generated location represents the distribution of the target's location over time. Then, Chapter 3 provides a detailed review on the SA algorithm, and in Chapter 4, the thesis will describe the implementation of a specific modified SA algorithm to obtain the best sensor deployment strategy based on the information obtained from the probability heat map. In Chapter 5, the algorithm finds a robust solution for patrol deployment that maximizes the target detection probability based on the maritime narcotics trafficking scenario. Based on the same scenario, sensitivity analysis is performed to evaluate the SA algorithm and its robustness towards input and parameter uncertainty due to incomplete information.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 2:
Probability Heat Map Construction

The Brownian Bridge Movement Model, or BBMM, has been used along with the Brownian bridge by experts in the ecology domain to describe and model animal movement between two points. For instance, Horne et al. (2007) used the BBMM to study the migration pattern of black bears, while Bunnefeld et al. (2011) used it to investigate the mule deer's movement patterns. The use of the BBMM, however, is not confined to the ecology domain. Recent work by Cheng (2016) and Nunez (2017) attempted to model human target movement using the BBMM. In Cheng's research, he employed the BBMM as a basis for the Temporal Brownian Bridge Model, or TBBM, to generate temporal heat maps of movement of aircraft carriers in the South China Sea. In contrast to the BBMM, which models one heat map by aggregating multiple start and end trajectories over time, the TBBM produces temporal heat maps that display the distribution of target location over the time period of travel, with the advantage of allowing additional input that can change the characteristics of the TBBM. This thesis uses the TBBM for target movement. The target movement is modeled based on simulated BBs, which are tied down at the start and end positions. Next, by aggregating these simulated BBs, a two-dimensional heat map that describes the distribution of target location can be generated. This chapter provides a brief introduction to Brownian motion, Brownian bridges, and the TBBM, and describes the underlying process of probability heat map generation in the following sections.

2.1 Brownian Motion

Brownian motion can be seen as a limit of the random walk process. Let us assume the basic random walk model, S_n is defined by

$$S_n = \sum_{i=1}^{n} X_i, \tag{2.1}$$

where X_i are independent and identically distributed (i.i.d.) random variables with mean 0 and some variance σ^2, and n is the number of terms in the sequence. BM is a continuous-

time stochastic process based on the limit of a function of S_n, as shown next. Define

$$W_n(t) = \frac{S_{\lfloor nt \rfloor}}{\sqrt{n}} \tag{2.2}$$

where $\lfloor nt \rfloor$ is the largest integer less than or equal to nt. As n approaches infinity, $W_n(t)$ approaches Brownian motion denoted as $W(t)$, $t \geq 0$. The key characteristics of BM are listed below:

1. $W(0) = 0$.
2. $W(t)$ is continuous for all $t > 0$.
3. $W(t)$ has independent and stationary increments due to i.i.d. X_i.
4. $W(t) - W(s)$ is normally distributed with mean 0 and variance $\sigma^2(t-s)$ for $0 \leq s \leq t$.

2.2 Brownian Bridge

Following the definitions from Section 2.1, a standard Brownian Bridge, or BB, can be defined from the BM conditioned such that $W(0) = 0$, $W(T) = 0$ and $\sigma^2 = 1$ for some $T > 0$. The BB is defined as

$$B(t) = W(t) - t(W(T)) \quad \text{for} \quad 0 \leq t \leq T. \tag{2.3}$$

Alternatively, the preceding equation can also be expressed as

$$B(t) = (T-t)W\left(\frac{t}{T-t}\right). \tag{2.4}$$

Equation 2.3 is often preferred in simulation modelling because it is computationally easier to simulate a BM and then transform it to a BB as described. Some of the properties of a standard BB over $t \in [0, T]$ are:

1. $B(t)$ is Gaussian.
2. $E[B(t)] = 0$.
3. $B(t)$ continuous for all $t \in [0, T]$.
4. $\text{Var}[B(t)] = \sigma^2 t(T-t)$.
5. $\text{Cov}(B(s), B(t)) = \sigma^2 s(T-t)$ for $s \leq t$.

From the variance expression $\text{Var}[B(t)]$, we can observe that the BB has the largest variance when $t = T/2$, which implies that the largest deviation from 0 is likely to happen in the middle of the traverse period of $[0, T]$. We can generalize to d dimensions, where $W(t)$ defines the d-dimensional BM, and $B(t)$ defines the Brownian bridge in the d-dimensional space at time t for some $0 \leq t \leq T$; thus the generic BB can be written as

$$B(t) = B(0) + (B(T) - B(0))\frac{t}{T} + W(t) - \frac{t}{T}W(T), \tag{2.5}$$

where the BB starts at $t = 0$ at position $B(0) \in \mathbb{R}^d$ and finishes at time $t = T$ at position $B(T) \in \mathbb{R}^d$.

In a two-dimensional case, the BB is governed by two independent one-dimensional BBs, where B_x and B_y are the x and y coordinates of $B(t)$ for some $0 \leq t \leq T$. The two one-dimensional BBs are expressed as

$$\begin{aligned} B_x(t) &= B_x(0) + (B_x(T) - B_x(0))\frac{t}{T} + W_x(t) - \frac{t}{T}W_x(T), \\ B_y(t) &= B_y(0) + (B_y(T) - B_y(0))\frac{t}{T} + W_y(t) - \frac{t}{T}W_y(T). \end{aligned} \tag{2.6}$$

These equations shown in (2.6) are the fundamentals for TBBM generation. They can be generalized for any starting values of $B_x(0)$ and $B_y(0)$, and ending values of $B_x(T)$ and $B_y(T)$. An illustration of a simulated BB starting at $B_x(0) = 0, B_y(0) = 0$, and ending at $B_x(T) = 1, B_y(T) = 1$ is shown in Figure 2.1.

2.3 Brownian Bridge Models for Target Movement

The Brownian Bridge Movement Model, or BBMM, is a technique that can model a target's location by assuming BB motions between two endpoints. The BBMM aggregates multiple BBs over time to produce a probability map. Horne et al. (2007) defines it as a "continuous-time stochastic model of movement in which the probability of being in an area is conditioned on starting and ending conditions," and uses the BBMM to study black bear migration patterns in Alaska. In this thesis, we use BBs to model target movement as they allows uncertainty without needing detailed local information by specifying only the start and end points of the target, and we employ the TBBM to simulate the distribution of target location. In contrast to the works of Horne et al., the research done by Cheng

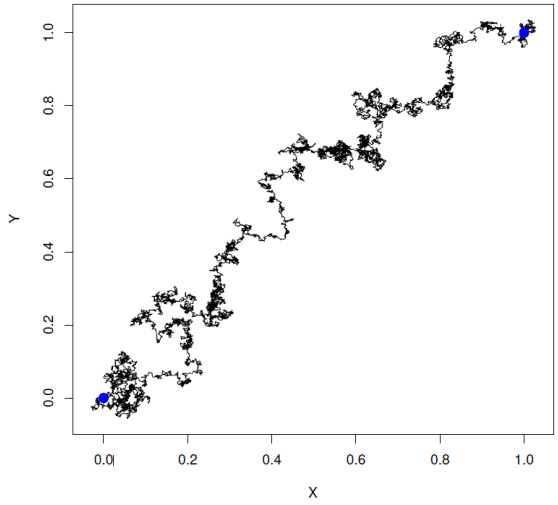


Figure 2.1. A simulated two-dimensional Brownian bridge. Source: Nunez (2017).

(2016) and Nunez (2017) produce temporal heat maps using the TBBM that provides the distribution of target location over the time period of travel, with the advantage of allowing additional input that can change the characteristics of the TBBM. As a result, the TBBM generates multiple heat maps that describe the target's positional probability over the span of time. Research from Cheng (2016) included extended intelligence options for probability heat maps, such as waypoints to allow trajectory detour, uncertainty of start and end points to simulate uncertainty in intelligence, and multipath generation to include possibility of multiple start and end points in a single scenario. Nunez (2017) incorporated particle filter updating methods and roughening techniques to further refine the heat map using information updates. Particle filtering takes each BB weighted based on affinity with intelligence and then resamples paths based on their likelihood. This allows information such as a sensor's detection and non-detection feedback to alter the possible heat map distributions in the future. Roughening techniques supplement particle filtering implementation and manage the problem of particle degeneracy by sampling new BBs during intelligence updates.

The BBMM and TBBM are parametrized by the variance σ^2 of the individual BB paths, and the start and end locations of the target. The variance parameter can also capture the aggregate level characteristics of weather and tactical behaviors, with minimal need

for detailed local behavior modeling. For instance, a straightforward path under clear weather conditions will imply a small variance value, whereas an indirect path and adverse weather conditions can be modeled with large variance. Cheng (2016) characterized the movement parameter by assuming a certain probability that a target will move out from its shortest projected path. Assume a to be the deviation from its expected central direction, as shown in Figure 2.2. According to Atkinson and Singham (2015), the probability of a

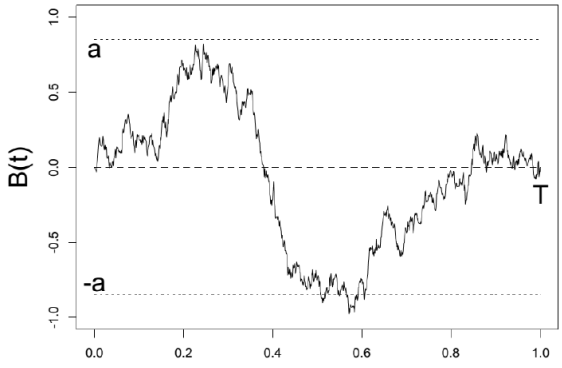


Figure 2.2. A simulated one-dimensional Brownian bridge with $(-a, a)$ bounds. Source: Cheng (2016).

one-dimensional BB moving out of a $(-a, a)$ bound is given by

$$P := Prob(|B(t)| > a) = e^{-2a^2/\sigma^2 T}$$
$$\Rightarrow \sigma^2 = -\frac{2a^2}{T \ln P}. \tag{2.7}$$

Thus, we can estimate σ^2 using Equation 2.7. The value of P is estimated by a subject matter expert. The other parameter a is also influenced by the speed of the target. When the speed of the target is high, the target tends to have a straightforward path and a lower probability of moving out of $(-a, a)$ bounds. Similarly, a slower speed implies higher variance, with the higher possibility of a target traveling backwards from its destination (in a two-dimensional case), and moving out of the $(-a, a)$ bounds (Cheng 2016). The values of σ^2 can be verified empirically by comparing them with the expected behavior of the target or heat map plots obtained from intelligence.

2.4 Generating Probability Heat Maps

The TBBM is generated through the following procedure. First, the simulation generates BBs using a discretized approximation based on user input of start and end points. Thereafter, the simulation produces a two-dimensional heat map by aggregating simulated BBs at discrete time steps. In this thesis, we follow Cheng's approach to determine the probability of the target being in an area in a specific time frame. To simulate information uncertainty, the TBBM divides the heat map area into multiple discrete cells and randomizes the starting and ending points using uniform distributions. At a certain discrete time t, the number of BBs in a cell is recorded and divided by the total number of BBs to calculate the probability of the target being in that cell (Cheng 2016).

Based on the scenario described in Section 1.3, the generated heat map that can estimate the distribution of the drug traffickers' movement over time is shown in Figure 2.3. The starting and ending locations of the target are sampled uniformly from the two-dimensional space, as depicted by the black boxes representing the uncertainty in the start and end points. There are two possible routes that the target is able to move from starting at the borders of Ecuador and Colombia; one ends in Costa Rica, and the other route ends in Mexico. The waypoint, such as the black box in the lower left area of the map, is additional information about the possible points that targets may travel to between the start and end points. The different colors of the heat map represent the probability of the target being present in the area, where areas with red have a higher probability and areas in blue have a lower probability.

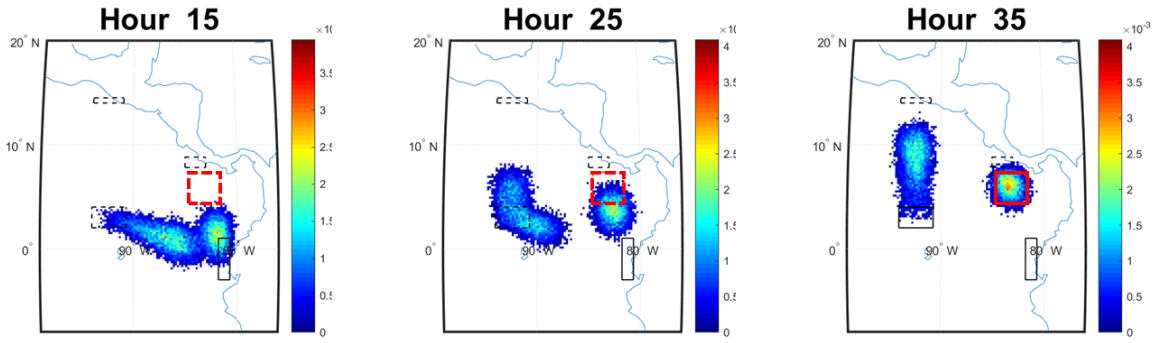


Figure 2.3. Probability heat map of detection generated from TBBM, snapshots at $t = 15, 25, 35$ hours.

The black boxes represent the waypoints of the target. The red box represents the sensor that is deployed over the area of interest.

Sensors are assumed to be rectangular in shape, as depicted by the red boxes in Figure 2.3, and have a constant probability of detection of 0.6 throughout the experiments (Cheng 2016). The sensor coverage area is plotted over the heat map to estimate the probability of detection by the sensors. To calculate the probability of detection, we count the number of detected BBs that pass through the cells of the sensor area at a snapshot of time when the sensors are assumed active. To obtain the overall probability of detection, we count only BBs detected for the first time by any one of the sensors with positive sighting (i.e., they have not been detected before), so there is no double counting. The simulation process finally calculates the probability of detection of a target in the scenario by dividing the total number of BBs detected over the total number of BBs simulated.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 3:
Simulated Annealing

Simulated annealing (SA) is a type of random search optimization algorithm. It is usually applied to global optimization problems that are without either closed-form solutions or knowledge of the internal structures of the objective (Zabinsky 2003). Given that our scenarios are based on the Temporal Brownian Bridge Model and the objective function for the probability of detection cannot be written analytically, this approach may be able to provide an approximate optimal solution.

This chapter provides a background on stochastic search for global optimization, the basics of the SA algorithm, SA's temperature cooling scheme and step size algorithm, and general guidelines for its implementation based on the book written by Zabinsky (2003). Subsequently, a customized SA procedure designed to search for the best sensor placement in a TBBM-based scenario is discussed in the next chapter.

3.1 Random Search Optimization

Stochastic optimization is global optimization in the presence of randomness in the objective function (Zabinsky 2003). Random search methods which have been used to solve deterministic optimization problems are now being used to solve stochastic optimization problems. Some of the more common methods of random search are SA, genetic algorithm and multistart methods. The advantage of using a random search method is its ease of implementation and ability to search non-convex spaces.

The basic concept of random search optimization is the "random" generation of new candidate points from the solution space after each iteration (Zabinsky 2003). Different optimization methods vary in terms of the acceptance criteria used for each randomly selected new point. In the wide range of optimization methods, there are two types of random search algorithm at the extremes; namely, Pure Random Search (PRS) and Pure Adaptive Search (PAS). The PRS is the simplest search case, which generates the sequence of search points randomly and uniformly in the feasible region, regardless of previous results. On the other hand, PAS takes into account previous information and proceeds to new points that

are improving from the current point in terms of objective function value. Both PRS and PAS are theoretical and not typically practiced due to their disadvantages. The best solution obtained for PRS is based on its stopping conditions, and there is no guarantee that the global optimal is reached. For PAS, actual implementation is extremely difficult as the selection of an improving point becomes increasingly complex for an arbitrary global optimization problem. In the middle lies SA and other search algorithms such as Hesitant Adaptive Search (HAS) and Annealing Adaptive Search (AAS), etc. (Zabinsky 2003). They differ from each other based on the generation and acceptance criteria of new points to improve the convergence rate and accuracy of the solution as compared to PRS and PAS. In the case of HAS, it does not accept non-improving points, and there is a chance that the algorithm rejects the better solution point with a small probability. Therefore, PAS is a special case of HAS where the probability of rejecting improving points is zero. While PAS and HAS sample new points from an improving set of the original sample space, AAS always samples from the original sample space while modifying its underlying sampling distribution. The parameter in AAS controls the sampling distribution, which allows the algorithm to focus its search from a uniform distribution to a distribution heavily concentrated around the global optimal as the value of the parameter decreases. If the parameter is set to infinity, then AAS is always sampling from the uniform distribution, and is approximated to a PRS.

This thesis focuses mainly on the SA method and attempts to find a global optimal solution by allowing non-improving objective function values to be explored with some probability to escape its local optimum. The following section describes some applications of the SA algorithm to solve optimization problems in diverse fields.

3.2 Simulated Annealing Algorithm Literature Review

Simulated annealing (SA) is a popular technique over the past three decades, due to its ease of implementation, convergence properties and means to escape local optima by allowing non-improving points to search for the possible global optimum. The SA algorithm has had success in solving problems in diverse fields, such as sonar track detection (Chen 1990), sensor placement for coverage (Branch 2012), path planning (Kindl and Rowe 2012; Yu and Lin 2015), logistic and inventory planning problems (Jawahar et al. 2012; Wei et al. 2018), and the influence maximization problem in the social network domain (Liu et al. 2017).

Research done by Chen (1990) discussed the implementation of the SA algorithm in the context of the sonar track detection problem. By fine tuning the algorithm, he successfully obtained the exact position of a low frequency acoustic track in the presence of a large amount of noise. Other research done by Jawahar et al. (2012) addresses a logistic distribution problem associated with backorder, known as the multi-period fixed charge distribution problem. The objective is to obtain a distribution plan that can minimize the total logistic cost incurred during the inventory holding period. An implementation of the SA algorithm solved the resultant pure integer non-linear programming problem. In another logistics problem, Wei et al. (2018) attempts to solve a problem that consists of the "capacitated vehicle routing problem and two-dimensional bin packing problem." The research employed the SA algorithm to solve the combination of two complex problems and compared the results obtained against best-known solutions from the literature. Wei et al. concluded that the SA algorithm outperforms existing algorithms in this setting with comparable computational time.

The approach in this thesis is to optimize sensor placement using the SA algorithm to maximize the overall target detection probability in a given scenario. With these past successes in solving problems in diverse fields, we will develop a SA algorithm in the unique setting of target tracking and detection. The following sections describe the SA algorithm in detail.

3.3 Basics of Simulated Annealing

The unique characteristic of the SA algorithm is its temperature parameter T_k that governs the sampling distribution, where k is the iteration count. The temperature parameter originated from the annealing process of metals. In metal annealing, solid metals are heated to break their lattice structure and slowly cooled back to form a new lattice structure. At a high temperature, the metal molecules move around to reform a lattice structure. Once the temperature is low, the metal molecules lose their excess energy to displace them from their local position and only vibrate slightly around their vicinity. Similarly in the SA algorithm, when the temperature is high initially, the algorithm searches around freely in the solution space for new points. As the temperature T_k decreases gradually over iterations, the search effort is localized.

To see the implementation of the SA algorithm, we first define a generic global optimization problem (P) (Zabinsky 2003) that minimizes the objective function value $f(X)$, such that

$$(P) : Z = \min_{x \in S} f(X), \tag{3.1}$$

where X is a vector of n decision variables, S is an n-dimensional feasible non-empty solution space, and f is a real-valued function defined over S. Let the global optimal solution to (P) be denoted by (X_{min}, Z_{min}) where

$$X_{min} = arg \min_{X \in S} f(X), \tag{3.2}$$

and

$$Z_{min} = f(X_{min}) = \min_{X \in S} f(X). \tag{3.3}$$

The general framework of the SA algorithm to solve the minimization problem (P) is shown next.

Initialization: Initialize algorithm parameters, such as the initial temperature T_0, and the initial candidate solution $X_0 \in S$, where S is the solution space for the problem, $\hat{Z} = f(X_0)$ where $\hat{Z}$ is the current best objective function value so far, and $k = 0$, where k is the iteration index.

Step 1: Generate a new candidate point $V_{k+1} \in S$ based on a Step Size Algorithm (to be defined).

Step 2: Update the next point X_{k+1} using

$$X_{k+1} = \begin{cases} V_{k+1}, & \text{with probability } P_{T_k}(X_k, V_{k+1}). \\ X_k, & \text{otherwise.} \end{cases} \tag{3.4}$$

where $P_{T_k}(X_k, V_{k+1})$ is the probability of accepting new point V_{k+1} given X_k and temperature T_k.

Step 3: Update algorithm parameters obtained in iteration k, including $T_{k+1} = \tau(Z_k, k, T_k)$, where τ is the temperature cooling schedule function and Z_k is the objective function value in iteration k. Update the best objective value obtained so far, $\hat{Z}$ as $\hat{Z} = \min\{\hat{Z}, f(X_{k+1})\}$

Step 4: Stop the optimization if a stopping criterion is met. Otherwise, increase k by one and return to Step 1.

In Step 1 of the SA framework, V_{k+1} is generated from a specific generator. These generators, also known as the Step Size Algorithms, are typically specific to each optimization algorithm. The generator makes a move from X_k to a nearby point V_{k+1} on iteration k in the same solution space S. Zabinsky (2003) mentioned several methods to generate candidate points, and these methods are discussed in depth in Section 3.5.

The key differentiating feature of the SA algorithm as compared to other stochastic optimization algorithms is its property of accepting points that have poorer objective values according to the probability function $P_{T_k}(X_k, V_{k+1})$. In the case where $f(V_{k+1}) \geq f(X_k)$, the algorithm will accept the new point V_{k+1} with a probability greater than zero, according to the Equation 3.5:

$$P_{T_k}(X_k, V_{k+1}) = \begin{cases} 1, & \text{if improving, i.e., } f(V_{k+1}) < f(X_k). \\ \exp\left[\dfrac{f(X_k) - f(V_{k+1})}{T_k}\right] & \text{otherwise, i.e., } f(V_{k+1}) \geq f(X_k). \end{cases} \tag{3.5}$$

We note that it always accepts improving points with probability 1, and only accepts non-improving candidate points at probability lower than 1. The preceding equation can also be written as

$$P_{T_k}(X_k, V_{k+1}) = \min\left(1, \exp\left[\frac{f(X_k) - f(V_{k+1})}{T_k}\right]\right). \tag{3.6}$$

The acceptance criterion described by Equation 3.6 is known as the Metropolis Criterion. We can see from the equation that the algorithm will accept non-improving points with high probability when the temperature parameter T_k is high, and the probability of accepting non-improving points also drops to zero as T_k decreases to zero. The decrease of T_k is usually scheduled to decrease with the performance of the next variable or the number of iterations, $T_{k+1} = \tau(Z_k, k, T_k)$, where τ is the temperature cooling schedule function that depicts the decreasing rate of T_k. Various schedules are discussed in detail in Section 3.4.

The convergence of the SA algorithm to reach the global optimal is ensured when the number of iterations k is allowed to approach infinity and T_k to approach zero. Therefore, a stopping criterion is usually included to terminate the algorithm. The stopping condition typically is

a function of $\hat{Z}$, and it is typically tailored to specific problems or user requirements. The stopping condition used in this thesis is discussed in Section 3.6.

3.4 Temperature Cooling Schedules

The cooling schedule for the temperature parameter T_k in the SA algorithm is like what is used in a metal annealing process. In metal annealing, solid metals are heated to break their lattice structure and slowly cooled back to form a new lattice structure. If the temperature reduction rate is sufficiently slow, the metal will have improved structural integrity. If the temperature is reduced too quickly, the resultant metal will be inferior and full of crystal defects. Similarly in the SA algorithm, if the cooling rate is too high, the algorithm may be trapped in a local optimum point too quickly. On the other hand, if the rate is too low, the algorithm will take a long time to converge to the solution.

Zabinsky (2003) described how the cooling schedule allows the algorithm to sample from the feasible region S in an approximate Boltzmann distribution around the global optimal point. For a given temperature T, the Boltzmann distribution is parameterized by $T > 0$ with density

$$g_T(x) = \frac{\exp[-f(x)/T]}{\int_S \exp[-f(z)/T]\, dz.} \tag{3.7}$$

where S is the feasible region. Assuming the case where T_k approaches infinity, then SA becomes a PRS, and as T_k approaches zero, the sampling region will shrink to a single point at the global optimal point Z_{min}. In Chapter 6 of her book, Zabinsky (2003) discussed a few cooling schedules that could be implemented in the SA algorithm:

Schedule 1: Adaptive cooling using a known global optimum Z_{min}:

$$T_k = \frac{2(f(X_k) - Z_{min})}{\chi^2_{1-\alpha}(n)}$$

.

Schedule 2: Adaptive cooling using an estimated global optimum $\tilde{f}_{min}$:

$$T_k = \frac{2(f(X_k) - \tilde{f}_{min})}{\chi^2_{1-\alpha}(n)},$$

Schedule 3: Geometric:

$$T_k = C^k T_0, \text{ where } C \in (0, 1),$$

Schedule 4: Logarithmic:

$$T_{k+1} = \frac{1}{\log(k+1)} T_0,$$

where $\chi^2_{1-\alpha}(n)$ is the $100(1-\alpha)$%-quantile of the chi-squared distribution with n degrees of freedom, where α is chosen to be small and $0 < \alpha < 1$.

Zabinsky's book presented the adaptive cooling schedule using an estimated global optimum after k iterations for Schedule 2, $\tilde{f}_{min}$, given by :

$$\tilde{f}_{min}(X_0, X_1, \ldots, X_k) = Z_{(0)k} - \frac{Z_{(1)k} - Z_{(0)k}}{(1-\alpha)^{-n/2} - 1}, \tag{3.8}$$

where $Z_{(0)k}$ and $Z_{(1)k}$ are the best and second-best objective function values obtained so far. The estimated objective function minimum, $\hat{f}_{min}$ is also the lower endpoint of a $100(1-\alpha)$% Confidence Interval (CI) for Z_{min}. An adaptive temperature cooling schedule like Schedules 1 and 2 decreases T in a way that allows the algorithm to generate new improved points with relatively high probability. It sets out to determine a new temperature T_{k+1} given the current objective function value Z_k, and using a Boltzmann distribution with temperature parameter T_k, the probability of finding a better Z in the next iteration is more than $1-\alpha$, such that

$$P(f(X_{k+1}) < Z_k) \geq 1 - \alpha, \tag{3.9}$$

where $0 < \alpha < 1$.

In this thesis, we adopt an adaptive schedule, Schedule 2, and discuss a modification of a static schedule, Schedule 4, as the cooling schedule for our customized algorithm. The customization and modifications are discussed in Chapter 4 in greater detail.

3.5 Step-Size Algorithm

The most common method of creating a candidate point is to take a step in a specific vector direction, known as the direction-step paradigm (Zabinsky 2003). In sequential search, the basics of generating a new point V_{k+1} is to take a step of length λ_k in the

specified direction D_k from the current point X_k. The iterative sequence of this method is known as the Hit-and-Run. The following sections describe the basics of Hit-and-Run, direction vector generation, point generation, and its discrete variations. This thesis adopts, with modification, the Hit-and-Run framework, and the new point is discretized with the step-function approach.

3.5.1 Basics of Hit-and-Run

In general, the Hit-and-Run sequence generates a sequence of points $\{V_k, k = 0, 1....\}$ on a bounded set $S \subseteq \mathbb{R}^n$, based on the following sequence (Zabinsky 2003):

Initialization: Initialize $X_0 \in S$ and set $k = 0$.

Step 1: Generate a random direction D_k according to the direction generator.

Step 2: Generate λ_k to set a new point

$$V_{k+1} = X_k + \lambda_k D_k$$

that is distributed over the following line set:

$$L_k = \{v : v \in S \text{ and } v = X_k + \lambda_k D_k \text{ , } \lambda_k \text{ a real scalar}\}.$$

If $V_{k+1} \notin S$, go back to Step 1.

Step 3: Stop when the overall optimization algorithm meets its stopping condition. Otherwise increase k by one and return to Step 1.

We also define the "neighborhood" of X_k to be an area around X_k or the set of V_{k+1} that is reachable from X_k in one iteration of the Hit-and-Run sequence (Henderson et al. 2003). According to Zabinsky (2003), under certain conditions, the sequence of points converges to a uniform distribution in the S space. The next section describes the method of generating a direction vector, D_k.

3.5.2 Direction Generator

There are various ways to sample the direction vector D_k mentioned in Phase 1 of the Hit-and-Run sequence, as follows:

Method 1: Sample D_k using a Hyperspherical Direction (HD) method to sample the vector from a uniform distribution on a unit hypersphere in $\mathbb{R}^n$ space.

Method 2: Sample D_k using a Coordinate Direction (CD) method to choose one direction uniformly from the n coordinate vectors spanning $\mathbb{R}^n$.

Method 3: Sample D_k using a n-uniform direction to choose uniformly in $(-(U_i - L_i)/2\ ,\ (U_i - L_i)/2)$ in each n dimension for $i = 1, ..., n$, where L_i and U_i are the lower and upper bounds of the n-dimensional S space.

In Method 1, the HD samples a unit vector that points from the initial point X_k to a point on a unit-hyperspherical surface in the n-dimensional S space. Method 2 selects one dimension uniformly from the n-dimensional space as its next direction vector. In the third method, the n-uniform direction method first uniformly samples in all n dimensions within its respective lower and upper boundaries in S space, with L_i and U_i at the midpoint of the sample space S. Subsequently, the combination of the n sampled dimension vectors results in the sampled direction vector D_k. This method effectively samples D_k in an n-dimension hyper-rectangular space.

While CD is typically able to reach convergence to the optimal faster than HD, the alignment of CD against the problem's coordinate direction significantly affects its rate and probability of convergence (Zabinsky 2003). Both the HD and the n-uniform direction method guarantee convergence; however, empirical tests have shown that the latter has the fastest convergence rate in the Hit-and-Run application (Zabinsky 2003). In this thesis, the optimization algorithm uses Method 3 for its direction generator due to its convergence properties.

3.5.3 Point Generation

Once the direction vector of iteration k, D_k is determined, the generation of a random candidate point depends on the sampled value of λ_k. The variable λ_k scales the distance and direction of V_{k+1} from X_k in S space. When $\lambda_k < 0$, the direction vector will point in the opposite direction of D_k from X_k. Also, when λ_k is close to zero, V_{k+1} will be very close to X_k. For V_{k+1} to exist in S, λ_k can be sampled uniformly over the interval bounded by the lower and upper bounds, $\lambda_{\min}$ and $\lambda_{\max}$ such that $X_k + \lambda_k D_k \in S$ for $\lambda_{\min} \leq \lambda_k \leq \lambda_{\max}$. There are two methods mentioned by Zabinsky (2003) for determining the bounds of λ_k.

The first method is to bound λ_k in the hyper-rectangular feasible region and use the one-dimensional acceptance criterion along the resulting line segment in n, until a feasible point is found for a specific value of λ_k, i.e., for all $X = (x_1, \ldots, x_n) \in S$,

$$L_i \leq x_i + \lambda_k d_i \leq U_i \text{ for } i = 1, \ldots, n,$$

where L_i and U_i are the lower and upper bounds of the n^{th} dimension of X_k for $i = 1, \ldots, n$, and $U_i > L_i$ for $i = 1, \ldots, n$. The drawback of this method is the tendency for the point generated from a point near the corner of the hyper-rectangular feasible region to be near the corner as well. This implies that the generated point tends to be trapped in the corner of the feasible region.

The second method, called the reflection generator (Zabinsky 2003) is able to reflect the point at the corner off the boundaries and then map them to the center of the region. For instance, suppose a feasible region is a n-dimensional box bounded by lower and upper bounds L_i and U_i for $i = 1, \ldots, n$. The point is now generated over an expanded box, where the lower and upper boundaries of the larger box are

$$\tilde{L}_i = L_i - (U_i - L_i),$$

$$\tilde{U}_i = U_i + (U_i - L_i),$$

for $i = i, \ldots, n$. The new point sampled, $\tilde{V}_i$, will be within the expanded region, and then mapped back to the original box to determine V_i, such that

$$V_i = \begin{cases} L_i + (L_i - \tilde{V}_i), & \text{if } \tilde{V}_i < L_i, \\ \tilde{V}_i, & \text{if } L_i \leq \tilde{V}_i \leq U_i, \\ U_i - (\tilde{V}_i - U_i), & \text{if } \tilde{V}_i > U_i, \end{cases}$$

for $i = i, \ldots, n$. This method is simple to implement, and convergence is proven via numerical methods (Zabinsky 2003). Furthermore, the reflection generator overcomes the issue faced in the first method, and sampled points will not be trapped in the corners of the hyper-rectangular feasible region. Therefore, this thesis adopted the reflection method, which is examined in Chapter 4 in greater detail.

3.5.4 Discrete Variations

The analogue version of the Hit-and-Run algorithm has variations to accommodate a feasible space that is in the discrete or mixed continuous-discrete domain. In this thesis, the feasible space contains a discretized variable for time, and thus there is a need to evaluate the problem in a mixed continuous-discrete space. In this section, we discuss two approaches that are described by Zabinsky (2003).

The first discrete variation is called the step-function approach. Here, the generator always generates points in the continuous domain, and the objective function is mapped back into the continuous-discrete domain. The problem (P) with feasible space S can be reformulated, with x_1 as the variables in n-dimension continuous space $\mathbb{R}^n$ and x_2 as the variables in the m-dimension integer space $\mathbb{Z}^m$ so that

$$(P) : \min_{X \in S} f(X)$$

$$S \subseteq \{(x_1, x_2) : x_1 \in \mathbb{R}^n, x_2 \in \mathbb{Z}^m, l \leq x_1 \leq u, L \leq x_2 \leq U\}$$

is transformed to (P') and $\underline{S}$, where

$$(P') : \min_{X \in \underline{S}} f(X)$$

$$\underline{S} = \text{closure}(\{(x_1, x_2) : x_1 \in \mathbb{R}^n, x_2 \in \mathbb{R}^m, (x_1, \lfloor x_2 \rfloor) \in S\}).$$

This approach creates piece-wise flat surfaces in the x_2 domain by mapping the continuous domain onto a step-wise domain, and the objective function is evaluated at the integer point nearest $x_2 \in S$. This creates a multidimensional step-wise objective function over S. Even though (P') is not continuous, the Hide-and-Seek algorithm is still able to converge in probability to the global optimum using the step-function approach (Zabinsky 2003).

In the second variation, the rounding approach has a similar concept as the step-function approach. The algorithm only keeps the discretized point that is generated from the continuous domain to be used for the next iteration. The next iteration always generates the next point starting from a discrete point. Although this approach is similar to the step-function approach, there is no proof of convergence using the rounding approach in general (Zabin-

sky 2003). Therefore, this thesis uses the step-function approach to generate new points in the continuous domain and then rounds them off to the original integer domain. The specifics are discussed in the next chapter.

3.6 Stopping Criterion

Since the algorithm performs better as k approaches inifinity, the SA algorithm has to be stopped by some predetermined conditions. As stopping conditions are not well defined in the literature, one way is to use the iteration count as a stopping condition. Alternatively, if the objective function value is met within a predetermined confidence interval before the iteration ends, the algorithm can be stopped to save processing time. As mentioned in Phase 3 of the SA framework (Section 3.5.1), the stopping criterion is usually tailored to the specific problem. Typically, it can use a known lower bound value for a maximization problem, or an upper bound for a minimization problem. For instance, if the objective function is to reduce cost, then the stopping condition can be a maximum error to the cost estimate that the user is willing to accept. If the objective function is to maximize profit, then the stopping criterion is the minimum profit target acceptable.

3.7 Implementation Guidelines

The parameters in the temperature cooling schedules, such as the geometric coefficient C and initial temperature T_0 mentioned in Section 3.4, have to be fine-tuned by observing the objective function values. Data analysis can also predict the parameters that fit the problem or user requirements so as to obtain consistent and reliable results. This thesis refines the parameters to obtain a robust result through the SA algorithm.

According to Henderson et al. (2003), cooling schedules are almost always heuristic, and can be categorized as static or adaptive. Static schedules are specified prior to the optimization, while adaptive schedules take input during the optimization to adjust the cooling rate. There are various studies that have been done to optimize the cooling schedules and equilibrium results. In general, a low cool down rate allows the search to explore more of the solution space. There is a longer period in which the algorithm will accept non-improving neighboring points. However, this is obtained at the expense of longer processing times. Conversely, a high temperature cool down rate converges to an optimal quickly, but the

lack of exploration will turn the algorithm into a "greedy search" and risk being trapped in a local optimal point too quickly. Both static and adaptive schedules are presented in the subsequent chapter.

The neighborhood, which is the area in the original solution space that is reachable from an iteration, affects the efficiency of the SA algorithm. A smooth neighborhood with no steep local optimum is preferred to a neighborhood with several deep local optima for the algorithm to converge reliably. The neighborhood size, which is related to the step-size convergence algorithm, affects both the speed to reach the global optimal and the precision of the result. If the step size is too large, the points may be moving around the true optimal for an excessive amount of time before convergence. If the step size is too small, the time to explore the solution space increases exponentially and exploring capabilities will be degraded. In this thesis, the neighborhood typically approaches smooth and continuous over the solution space as the number of simulated BBs approach infinity and the discretization size of the heat map cells approach zero. Therefore, the choice of step size will not pose a major issue for our implementation.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 4:
Simulated Annealing Algorithm for Sensor Placement

In this chapter, we describe a particular implementation of the SA framework for the optimization problem in the context of target detection. The thesis implements the SA algorithm based on the temporal probability heat map generated from the TBBM to attempt to obtain the best sensor configuration, and determines the deployment of sensors that can maximize the overall target detection probability based on probability heat maps and known target starting and ending locations.

The first section of this chapter walks through the framework of the SA algorithm with reference to the implementation on the sensor placement objective to obtain the maximum target detection probability. Next we describe the temperature schedule and step-size algorithm that are custom to the problem. The last section evaluates the stopping condition that is tailored to this specific setting.

4.1 Model Outline

The indices used in the subsequent chapters are defined as follows:

$i = 1, \ldots, N$	number of sensors,
$j = \{x, y, t\}$	dimension in solution space,
$k = 1, \ldots, L$	iteration count.

We define the fixed parameters in a specific scenario within its area of interest (AOI) in the following manner:

T_U	Upper limit of time of mission (hours),
T_L	Lower limit of time of mission (hours),
X_U	Upper longitudinal boundary in AOI (degrees),
X_L	Lower longitudinal boundary in AOI (degrees),
Y_U	Upper latitudinal boundary in AOI (degrees),
Y_L	Lower latitudinal boundary in AOI (degrees).

We define the set of decision variables $Q^{(k)}$ and the individual decision variables in this set $q_{ij}^{(k)}$, as

$Q^{(k)} = \{q_{ij}^{(k)}\}$	Set of decision variables at iteration k, for $i \in \{1, \ldots, N\}$, $j \in \{x, y, t\}$,
$q_{ij}^{(k)}$	Value of j^{th} dimension of the i^{th} sensor on the k^{th} iteration.

At iteration k, $q_{ix}^{(k)}$ is the longitudinal position and $q_{iy}^{(k)}$ is the latitudinal position of sensor i in the AOI , and $q_{it}^{(k)}$ is the time at which sensor i is present to try to detect the target. The sensor decision variables are bounded such that $q_{ix}^{(k)} \in [X_L, X_U]$, $q_{iy}^{(k)} \in [Y_L, Y_U]$ and $q_{it}^{(k)} \in [T_L, T_U]$.

We define our optimization problem (P) that maximizes the objective function value $f(Q)$, the probability of target detection in the optimization problem, to be

$$(P) : Z = \max_{Q \in S} f(Q), \tag{4.1}$$

where S is a three-dimensional feasible non-empty solution space, defined as

$$S = \{(x, y, t) : \ x \in [X_L, X_U], \ y \in [Y_L, Y_U], \ t \in [T_L, T_U]\}. \tag{4.2}$$

Since the true probability of detection is unknown, we estimate its value using the average observed in the TBBM simulation. Define Q as the areas contained by the active sensors and $BB_b \in Q$ denotes the event that a BB that travels through active sensor set Q. The objective

function f is the detection probability function, which is obtained from the total number of simulated BBs successfully detected by all sensors Q, divided by the total number of simulated BBs in a scenario m, defined as

$$f(Q) = \frac{1}{m} \sum_{b=1}^{m} 1_{(BB_b \in Q)}, \tag{4.3}$$

where

$$1_{(BB_m \in Q)} = \begin{cases} 1, & \text{if } (BB_m \in Q) \text{ is true .} \\ 0, & \text{otherwise.} \end{cases} \tag{4.4}$$

The SA algorithm to determine the maximum detection probability is described here.

Initialization: Initialize algorithm parameters, including initial temperature $T^{(0)}$, initial points $Q^{(0)} \in S$, and $k = 0$, the best Z so far, $\hat{Z} = f(Q^{(0)})$ and the corresponding best Q so far, $\hat{Q} = Q^{(0)}$.

Step 1: Generate a candidate point $V^{(k+1)} \in S$ according to the Hit-and-Run algorithm with the n-uniform direction generator and reflection generator.

Step 2: Update the current point $Q^{(k+1)}$ using

$$Q^{(k+1)} = \begin{cases} V^{(k+1)}, & \text{with probability } P_{T^{(k)}}(Q^{(k)}, V^{(k+1)}). \\ Q^{(k)}, & \text{otherwise.} \end{cases} \tag{4.5}$$

where $P_{T^{(k)}}(Q^{(k)}, V^{(k+1)}) = \min \left\{ 1, \exp \left[\frac{(f(V^{(k+1)}) - f(Q^{(k)}))}{T^{(k)}} \right] \right\}$.

Step 3: Update algorithm parameters, including $T^{(k+1)} = \tau(k, T^{(k)}, \hat{Z})$. Update $\hat{Z} = \max\{\hat{Z}, f(Q^{(k+1)})\}$, and $\hat{Q}$ to be the current best solution.

Step 4: Terminate the algorithm if a stopping criterion is met. Otherwise increment k and return to Step 1.

Step 5: Obtain an estimate of the optimal objective function value as $\hat{Z}$ and optimal solution of $\hat{Q}$.

4.2 Static Temperature Schedule

There are two cooling schemes that this thesis evaluates in the algorithm to define $T^{(k+1)}$ in Step 2 of the SA algorithm. One way is to use a static logarithmic cooling scheme for τ, defined as

$$T^{(k+1)} = \frac{T^{(0)}}{\log\left(\frac{k}{C_T N} + 1\right)}, \tag{4.6}$$

where $C_T > 0$ is a scalar constant that scales the rate of cooling and N is the number of sensors to optimize. The inclusion of N in the equation ensures that the increase in dimensionality due to the increased number of sensor parameters is accounted for. Since the logarithmic term increases at a slower rate, the temperature is cooled at a lower rate as N increases. The allocated proportion of iterations for initial exploration should be therefore proportionately increased by a factor of N. For instance, given the same $T^{(0)}$ and C_T, we can see that the algorithm has to reach a higher iteration count k as N gets larger to obtain a particular temperature T.

With the static temperature schedule, we can also write the probability of acceptance P_{T_k} in Step 2 of the algorithm in the following manner:

$$P_{T^{(k)}} = \min\left\{1, \exp\left[\frac{(f(V^{(k+1)}) - f(Q^{(k)}))}{T^{(0)}} \log\left(\frac{k}{C_T N} + 1\right)\right]\right\}. \tag{4.7}$$

We can regard the term $f(V^{(k+1)}) - f(Q^{(k)})$ as the error from current point $Q^{(k)}$. The other constants C_T and $T^{(0)}$ can be defined by the user to adjust the cooling rate and initial probability of acceptance accordingly.

As the selection of constants proposed in the static schedule is mostly arbitrary and heuristic, this method will require some trial and error. To reduce errors due to input uncertainty, this thesis focuses on an adaptive temperature schedule described in the subsequent section.

4.3 Adaptive Temperature Schedule

An adaptive temperature schedule was discussed in Section 3.4, where the schedule adapts to the changes of the objective function during the iteration. In this thesis, a modified

adaptive temperature schedule (Zabinsky 2003) is proposed as

$$T^{(k+1)} = \frac{2(\tilde{Z}^{(k)} - Z^{(k)})}{\chi^2_{1-\alpha}(n)} r_A^k \tag{4.8}$$

where $Z^{(k)}$ is the objective function value in the k^{th} iteration, $\tilde{Z}^{(k)}$ is the estimated best objective function value, r_A^k is a geometric reduction factor, $0 < r_A < 1$, $\chi^2_{1-\alpha}(n)$ is the $100(1-\alpha)\%$-quantile of the chi-squared distribution with n degrees of freedom, where n is the cardinality of $Q^{(k)}$. Given that $\hat{Z}^{(k)_0}$ is the best Z obtained so far, and $\hat{Z}^{(k)_1}$ is the second best Z obtained so far after k iterations, we can obtain the estimate of the best objective function $\tilde{Z}^{(k)}$ (Zabinsky 2003), defined as follows,

$$\tilde{Z}^{(k)} = \hat{f}(Q^{(0)}, \ldots, Q^{(k)}) = \hat{Z}^{(k)_0} + \frac{\hat{Z}^{(k)_0} - \hat{Z}^{(k)_1}}{(1-\alpha)^{-n/2} - 1}, \tag{4.9}$$

where $Z^{(k)_0} = \max_m(f(Q^{(m)})),\ m = 0, ..., k$, with m_0 being the value of m that maximizes the expression, and $Z^{(k)_1} = \max_m(f(Q^{(m)})), m = 0, ..., k, m \neq m_0$.

We can write the probability of acceptance $P_{T^{(k)}}$ in the k^{th} iteration, in the following manner:

$$P_{T^{(k)}} = \min\left\{1, \exp\left[\frac{(f(V^{(k+1)}) - f(Q^{(k)}))\chi^2_{1-\alpha}(n)}{2\left(\left(\hat{Z}^{(k)_0} + \frac{\hat{Z}^{(k)_0} - \hat{Z}^{(k)_1}}{(1-\alpha)^{-n/2} - 1}\right) - Z^{(k)}\right) r_A^k}\right]\right\}. \tag{4.10}$$

The inclusion of the geometric reduction factor r_A^k ensures that the temperature will decrease gradually even if no new improving candidate point is found. Then, r_A can be picked arbitrarily as long as r_A is very close to 1. If r_A is too small, then $T^{(k+1)}$ will decrease too quickly, giving poor results. In this thesis, we use the adaptive temperature schedule, and r_A is set as 0.999. Also, α will be set to 0.05, which means that the estimate of $\tilde{Z}$ is the upper endpoint of a $100(1-\alpha)\% = 95\%$ confidence interval for the true optimal value of Z.

4.4 Modified Step-Size Algorithm

The algorithm uses a Hit-and-Run method with a n-uniform direction generator. The generator is enhanced with the implementation of the reflection generator mentioned in Section 3.5.3. We index the new boundaries in the following manner, such that $q_{ix}^{(k)} \in [L_x, U_x]$, $q_{iy}^{(k)} \in [L_y, U_y]$ and $q_{it}^{(k)} \in [L_t, U_t]$, i.e., $q_{ij}^{(k)} \in [L_j, U_j]$ where

$$\begin{aligned} L_x &= X_L - (X_U - X_L)/2, \\ U_x &= X_U + (X_U - X_L)/2, \\ L_y &= Y_L - (Y_U - Y_L)/2, \\ U_y &= Y_U + (Y_U - Y_L)/2, \\ L_t &= T_L - (T_U - T_L)/2, \\ U_t &= T_U + (T_U - T_L)/2. \end{aligned}$$

The boundaries of S are expanded into S', defined as

$$S' = \{(x, y, t) : \ x \in [L_x, U_x], \ y \in [L_y, U_y], \ t \in [L_t, U_t]\}. \tag{4.11}$$

After this transformation, the feasible space effectively grew two times per dimension, or eight times in volume. The expansion of space is done to ensure that the sampled points will not be trapped in the corners of the hyper-rectangular feasible region.

The direction vector of $q_{ij}^{(k)}$ is generated uniformly using their respective expanded boundaries as shown here,

$$d_{ij} \in \text{Unif}\,(L_j - U_j, U_j - L_j) \quad \forall\, i, j. \tag{4.12}$$

The new point generated in iteration k, $v_{ij}^{(k+1)}$, which is scaled by a scalar $\lambda_i^{(k)} \in \mathbb{R}$, is defined as

$$v_{ij}^{(k+1)} = q_{ij}^{(k)} + \lambda_i^{(k)} d_{ij}^{(k)} \quad \forall\, i, j, k, \tag{4.13}$$

where the $\lambda_i^{(k)}$ is the same for all three dimensions of j for any sensor i.

Using the one-dimensional acceptance-rejection criterion mentioned in Section 3.5.3, we determine the maximum and minimum range of λ_i, $\lambda_{i_{max}}$ and $\lambda_{i_{min}}$ that bounds v_{ij} in the

expanded region S'. We define the boundaries of λ_i in each iteration k in the following manner:

$$\lambda_i \leq \lambda_{i_{max}} = \min \begin{cases} \dfrac{U_j - q_{ij}}{d_{ij}} & \text{for } j \text{ such that } d_{ij} > 0 \\ \dfrac{L_j - q_{ij}}{d_{ij}} & \text{for } j \text{ such that } d_{ij} < 0 \end{cases}, \forall\, i, \tag{4.14}$$

$$\lambda_i \geq \lambda_{i_{min}} = \min \begin{cases} \dfrac{L_j - q_{ij}}{d_{ij}} & \text{for } j \text{ such that } d_{ij} > 0 \\ \dfrac{U_j - q_{ij}}{d_{ij}} & \text{for } j \text{ such that } d_{ij} < 0 \end{cases}, \forall\, i. \tag{4.15}$$

Subsequently, λ_i is sampled uniformly between $\lambda_{i_{min}}$ and $\lambda_{i_{max}}$ for each iteration k, where

$$\lambda_i \in \text{Unif}\,(\lambda_{i_{min}}, \lambda_{i_{max}}) \quad \forall\, i. \tag{4.16}$$

We note that $v^{(k+1)}$ is in expanded space S' but may not be in the original space S. The new points that are outside of S are reflected into S along the boundaries of S. The reflection logic to map $v_{ix} \in S'$ to $V_{ix} \in S$ is elaborated in Equation 4.17:

$$V_{ix} = \begin{cases} 2X_L - v_{ix}, & \text{if } v_{ix} < X_L, \\ v_{ix}, & \text{if } X_L \leq v_{ix} \leq X_U, \\ 2X_U - v_{ix}, & \text{if } v_{ix} > X_U. \end{cases} \tag{4.17}$$

Similarly, v_{iy} and v_{it} are also reflected based on their respective boundaries, $[Y_L, Y_U]$ and $[T_L, T_U]$ into V_{iy} and V_{it} using Equation 4.17, described by

$$V_{iy} = \begin{cases} 2Y_L - v_{iy}, & \text{if } v_{iy} < Y_L, \\ v_{iy}, & \text{if } Y_L \leq v_{iy} \leq Y_U, \\ 2Y_U - v_{iy}, & \text{if } v_{iy} > Y_U, \end{cases} \tag{4.18}$$

and

$$V_{it} = \begin{cases} 2T_L - v_{it}, & \text{if } v_{it} < T_L, \\ v_{it}, & \text{if } T_L \leq v_{it} \leq T_U, \\ 2T_U - v_{it}, & \text{if } v_{it} > T_U. \end{cases} \tag{4.19}$$

The resultant sample distribution converges to a uniform distribution in space S, as shown in Figure 4.1. The plots are the realization of a 2,000-iteration run using the Hit-and-Run step size algorithm implemented with two sensors, one in yellow and one in blue, without optimization. We can observe that the sensor points are approximately evenly distributed within the S space in the left plot of Figure 4.1 using the Hit-and-Run algorithm without the SA algorithm. This is in contrast to the case where SA optimization is used in the right plot of Figure 4.1, where the iteration points of the sensors tend to cluster close to the current best value.

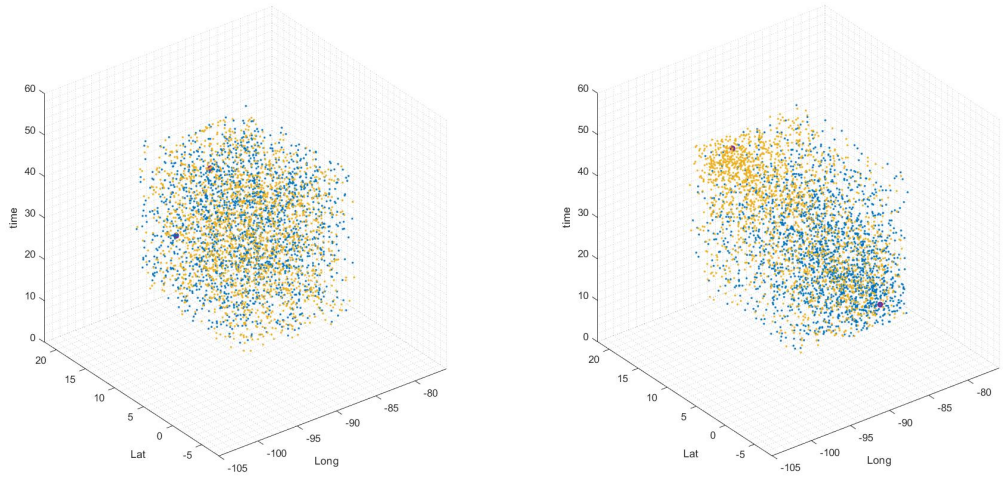


Figure 4.1. Plots of q in S space with pure Hit-and-Run (left) and Hit-and-Run with optimization (right).

Blue and orange points are the sampled q_1 and q_2 respectively, in a 2,000-iteration run in the S space. With optimization, new iteration points tend to cluster close to the current best value.

We also note that, although the points q_i are sampled in the continuous S space, the TBBM heat map does round off the time of the sensor's observation. This discretizes q_{it} before

being fitted into the probability heat map. It has been discussed in Section 3.5 that the discrete-continuous space problem is still able to converge to global optimum.

4.5 Stopping Conditions

The first stopping condition is when the iteration count reaches the total maximum number of iterations, i.e., $k = l$. The maximum number of iterations is also set proportional to the number of sensors, N, such that

$$l = 1000N. \tag{4.20}$$

This is done to allocate more iterations for cases when the degrees of freedom of the problem is large, anticipating that the algorithm will require more time to optimize all $q \in Q$. The relationship between run length and sensor count is chosen to be linear to ensure that the overall run-time of the algorithm is kept reasonably low. Once the algorithm reaches the end of all iterations, the best objective function value obtained so far, $\hat{Z} = f(\hat{Q})$ is the approximate maximum with its argument $\hat{Q}$ being the best sensor deployment configuration.

4.6 Algorithm Summary

In summary, we develop a particular implementation of the SA algorithm for the purpose of target detection. The step-size algorithm is slightly modified to suit the space dimensionality of our specific problem. Also, by modifying the adaptive temperature schedule and implementing a customized stopping condition, we hope to improve the algorithm's run-time and probability of convergence.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 5:
Experimental Analysis

In this chapter, the optimized results obtained from the SA algorithm optimization is analyzed and compared with simple heuristic results. Based on the Central American smuggling scenario described in Section 1.3, an experimental design is proposed to evaluate the performance of the SA algorithm. Inputs such as the number of sensors, sensor size, variance of target movement, and number of paths, are varied to investigate the effects of inputs on SA algorithm performance. Next, we investigate the best sensor configuration to maximize the target detection probability by limiting the total available sensor search area.

5.1 Probability Ratio of the SA Algorithm Results to the Heuristic Algorithm Results

The heuristic sensor placement method can be described as the most naive placement method. It obtains the coordinate of the cell with the highest probability from the heat map, and uses it as the center position of the sensor. This is based on the assumption that the highest valued cell is usually surrounded by cells with a relatively high probability of detection. An illustration of a heuristic sensor placement is shown in Figure 5.1. In this thesis, we assume the sensors deployed in the heuristic case to be active during the mid-section of the scenario, where $15 \leq t \leq 55$ hours. For instance, at the minimum extreme where $N = 1$, the heuristic sensor will be active at $t = 35$, and when $N = 2$, the heuristic sensors will be active at $t = \{35, 40\}$, etc. At the maximum extreme where $N = 5$, the five sensors will be active at $t = \{25, 30, 35, 40, 45\}$, respectively. While this may not be the best placement option, it is a heuristic that can be easily implemented automatically.

The heuristic algorithm results are theoretical lower bounds to the optimization problem. As the solutions are in the feasible space S, the maximum probability of detection has to be at least equal to or higher than the heuristic results. Therefore, they are used as a benchmark for comparison with the SA algorithm results.

We formally define the probability of detection obtained from the SA algorithm as Z_{SA}, and the probability of detection obtained from the heuristic method as Z_h. A probability ratio

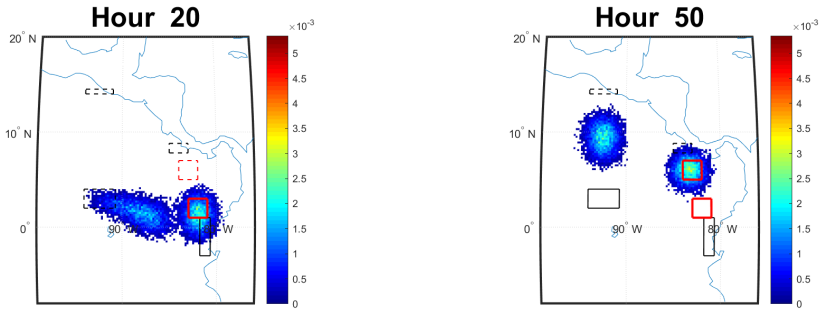


Figure 5.1. Probability heat map with heuristic algorithm sensors activated at $t = 20, 50$ hours.

Note that the sensors turn from dotted red boxes to solid red boxes at the point of sensor activation.

based on the ratio of the probability of detection obtained from the SA algorithm to that from the heuristic to evaluate the performance of the SA algorithm is

$$ratio = \frac{Z_{SA}}{Z_h}. \tag{5.1}$$

Numerically, a probability ratio value larger than one implies that the SA solution outperforms the heuristic algorithm. Also, a larger value of the probability ratio signifies a more significant improvement of SA over the heuristic algorithm's solution, and a value smaller than one signifies the converse. The ratio is less than one when the SA algorithm under-performs compared to the heuristic, which may be due to non-convergence of the solution within the run-time limits.

5.2 Simulated Annealing Algorithm Performance

We vary several variables and parameters of the model to evaluate the performance of the SA algorithm. There are two main measures of performance; namely, the ratio of the probability of detection obtained from SA algorithm to the probability of detection for the heuristic, and the run-time of the algorithm. We obtain the two measures of performance while varying the number of sensors N, the sensor size, the variance of the BBMM, the number of possible paths made by the target, and "Time Optimized" (whether the time of sensor placement is optimized), as summarized in Table 5.1.

The variance σ^2 is obtained from Equation 2.7, which increases the uncertainty of the target during its travel over the path as σ^2 increases. In the experiments, all the sensors are assumed to be square in shape, and the sensor size varies the length of the sides of the sensor. In addition, all the sensors in a particular experiment run are assumed to be the same size. The number of paths varies the number of possible trajectories that the target can transverse from start to end, and a number larger than one assumes a multipath scenario, not to be confused with the number of simulated BBs. The "Time Optimized" option switches on and off whether the times of sensor placement are optimized relative to the heuristic. When "Time Optimized" is on, q_{it} is varied over all discrete t between $15 \leq t \leq 55$ rounded to the nearest multiples of 5. Conversely, when the option is off, the algorithm assumes a fixed value for q_{it} based on the heuristic sensor timings, and optimizes over q_{ix} and q_{iy} only. For instance, for a $N = 2$ scenario, if the heuristic sensors are active when $t = \{35, 40\}$ and "Time Optimized" is off, then the SA sensors will also be active when $q_{1t} = 35$ and $q_{2t} = 40$, i.e., $t = \{35, 40\}$. When "Time Optimized" is on, then q_{1t}, q_{2t} will be chosen from $\{15, 20, 25, \ldots, 55\}$.

Table 5.1. Experimental design to evaluate performance of the SA algorithm.

Factor	Low	High	Factor Type
N	1	5	Discrete $\mathbb{Z}$
σ^2	0.1	0.5	Continuous $\mathbb{R}$
Sensor Size	1	3	Continuous $\mathbb{R}$
# of paths	1	3	Discrete $\mathbb{Z}$
Time Optimized	0 (off)	1 (on)	Nominal

We employ a Nearly Orthogonal Latin Hypercube (NOLH) (Sanchez 2011) to produce a 65-design point experimental design, with four replications each run, to analyze the effects of the factors described in Table 5.1 on the measures of performance. These experimental designs are structured to allow the space to be tested efficiently using a minimal number of design points as compared to a full factorial model. The current run-time for a single replication of 65 design points is approximately two hours of computational effort, and a conventional 30-repetition experiment will take approximately 2.5 days to complete. Therefore we propose to use a lower repetition count to save computational effort.

To justify the low replication count, we pick an arbitrary case where $N = 1$, $\sigma^2 =$

0.2, Sensor Size = 2, # of paths = 2 with"Time Optimized" off. We perform 30 replications and analyzed the resulting probability of detection obtained from the heuristic and SA, Z_h and Z_{SA}. The results shown in Figure 5.2 indicate that the 95% CI of Z_h is $(0.286, 0.300)$ and of Z_{SA} is $(0.310, 0.315)$. Compared with a four replication result, as shown in Figure 5.3, the corresponding 95 percent CI of Z_h is $(0.296, 0.312)$ and Z_{SA} is $(0.308, 0.318)$, which are sufficiently small considering the small sample size. There are two levels of stochasticity in the simulation due to the randomness in heat map generation and randomness of the SA search algorithm; however, the results obtained remained low in variance relative to the difference between the algorithms. Thus, we are able to run the simulations with fewer replications and still obtain consistent results. More repetitions could be run, but it would take more time without a significant improvement in differentiating the two approaches.

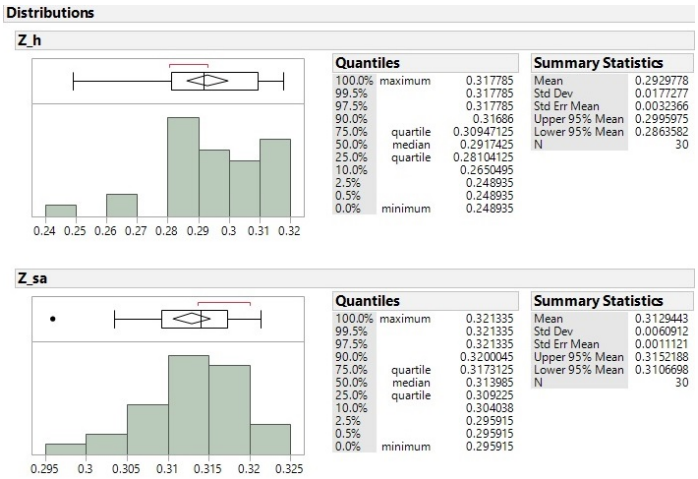


Figure 5.2. Histogram of Z_h (top) and Z_{SA} (bottom) from 30 replications.

5.2.1 SA Algorithm Results Compared to Heuristic Results

After obtaining the probability ratio for all design points, we use a partition tree as shown in Figure 5.4 to evaluate the effects of the factors on the probability ratio. From the analysis, we observe the SA algorithm outperforms the heuristic method in most cases. It is also notable that variance is the most significant variable affecting the ratio. When the variance is less than 1.9, SA results tend to outperform heuristic results by a large margin. In particular, the SA algorithm outperforms heuristic results under certain conditions; one is in the case when $\sigma^2 < 0.19$, and the other case is when $\sigma^2 \geq 0.19$, sensor size ≥ 1.7 and under a

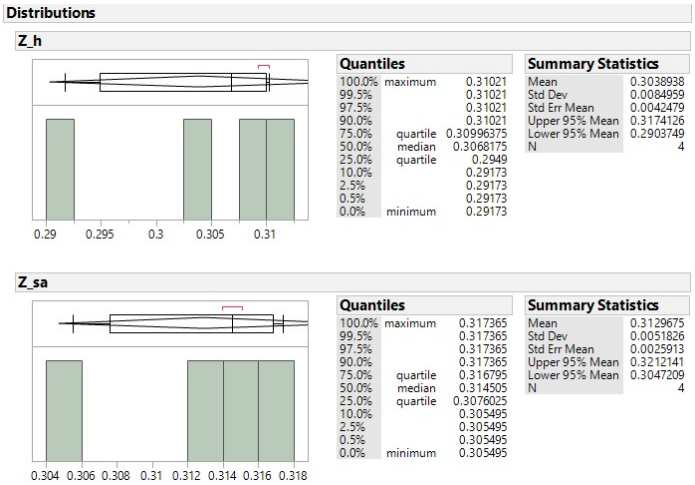


Figure 5.3. Histogram of Z_h (top) and Z_{SA} (bottom) from four replications.

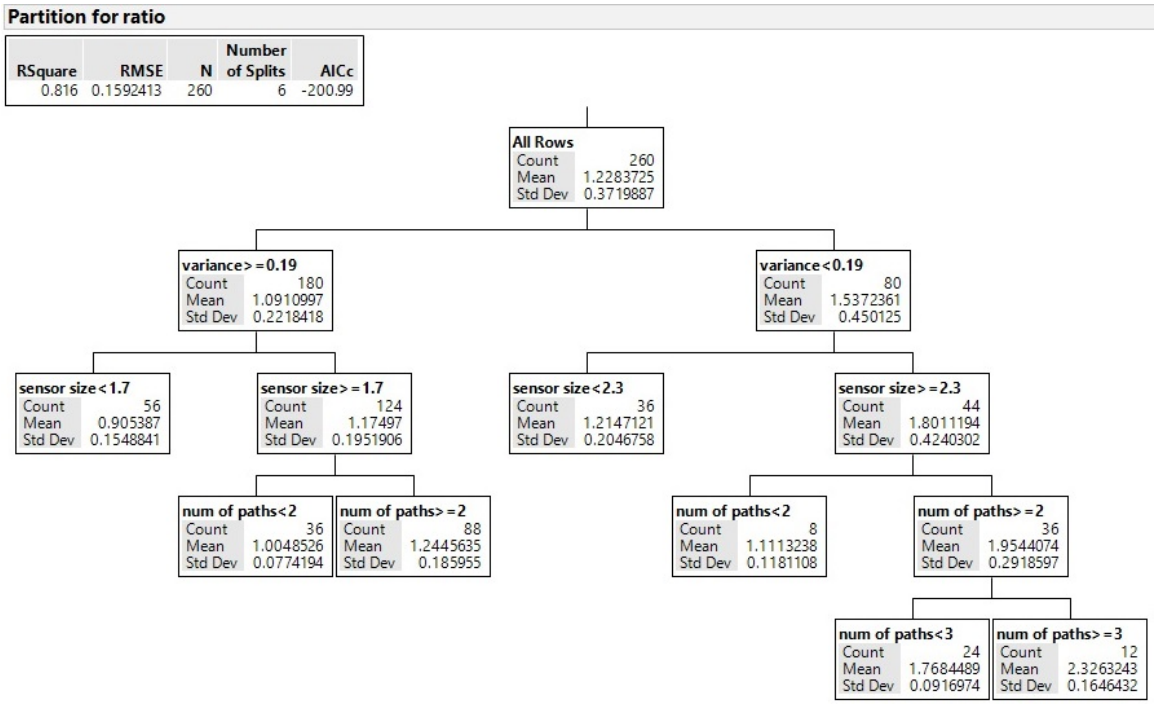


Figure 5.4. Partition tree of the ratio for SA with random initialization.

multipath scenario condition. This is because when large sensors are covering large areas of the heat map that has a small target uncertainty spread, the SA algorithm is able to detect areas with high probability of detection more effectively. There are instances, however, where the heuristic results perform better than the SA algorithm. In the lower extreme cases

when σ^2 is more than 0.19 and sensor size is less than 1.7, the ratio is statistically less than 1, with a 95 percent CI of $(0.885, 0.931)$. This implies that 22 percent of the time, the solution obtained from the SA algorithm is suboptimal as compared to the heuristic, which may be due to non-convergence. The combination of the large variance of the probability density and small sensor size causes the heat map surface in the multidimensional space S to be flatter than the case with lower variance, with values very close to zero. Given that the Hit-and-Run algorithm in SA only approximates to the Boltzmann distribution as the number of iterations approaches infinity, the SA algorithm may not be sensitive enough to detect the small changes when we limit the iteration count.

The partition tree results also show that the "Time Optimized" option is not a dominant factor in improving the optimal solution. Nevertheless, when we evaluate the distribution of"detection ratio" against "Time Optimized," we can see from Figure 5.5 that there is a significant difference between the two results. When the sensors are optimized over time, the 95% CI of the ratio is $(1.22, 1.37)$, which is significantly higher than when "Time Optimized" is off, which has a ratio CI of $(1.11, 1.20)$. This result is expected since optimization over time is a relaxation to the optimization problem when time is fixed; therefore, the optimal objective function value is expected to be at least equal to or higher when "Time Optimized" is turned on versus when it is off. Although the results here show the significance of the "Time Optimized" option, its effect on the probability ratio is dominated by other factors like σ^2 and sensor sizes. Therefore, it did not appear to be significant in the partition tree in Figure 5.4.

5.2.2 Extreme Case Scenarios

We investigate an upper extreme case where the ratio is greater than 2, specifically, when $N = 2$, $\sigma^2 = 0.1$, sensor size = 2.9, three paths, and "Time Optimized" is on. The heat maps for the heuristic and SA results are shown in Figure 5.6, with a mean value of the ratio being 2.497.

When there are more paths that the target can take in the scenario, the SA algorithm generally achieves better optimal estimates as compared to the heuristic. This is intuitive since the SA algorithm is able to explore the solution space to search for better sensor deployment positions and combinations. As shown in Figure 5.6, the two-sensor deployment was able

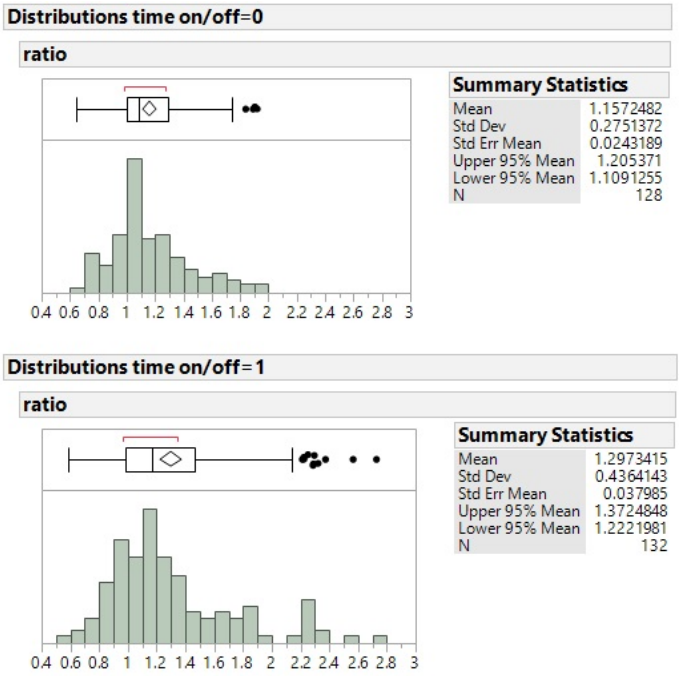
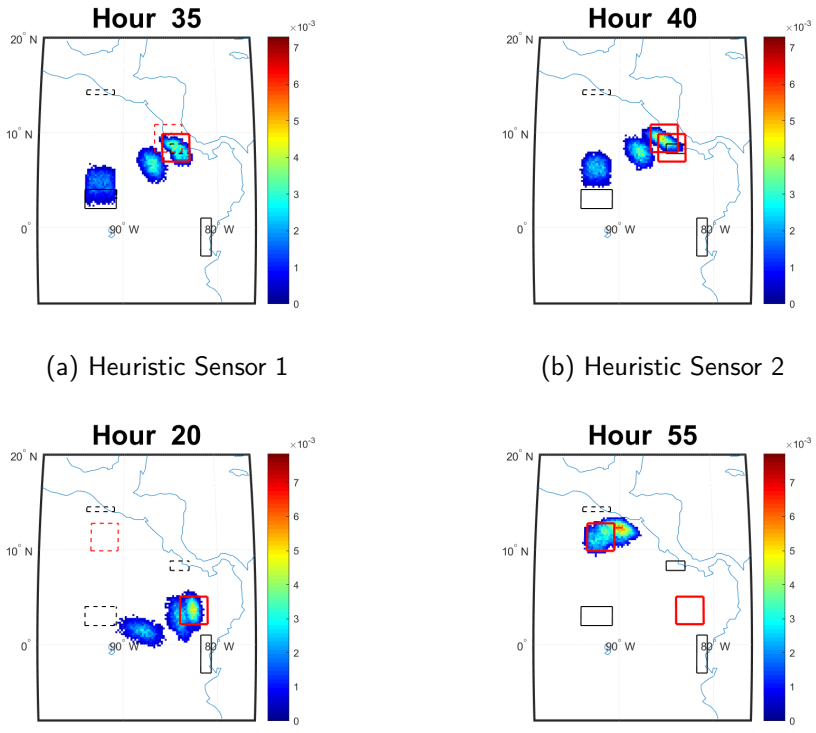


Figure 5.5. Histogram of the ratio against "Time Optimized" off (top) and on (bottom).

to cover different paths to maximize overall target detection probability.

Multiple sensors in the heuristic placement tend to have large overlapping coverage that does not fully utilize the available detection area when the sensor size is large. As the heuristic placement is determined by placing the center of the sensors at the highest probability without considering the boundaries of each individual sensor, the sensors tends to overlap more when the peaks are close. With the same argument, the heuristic placement also tends to focus on placing all the sensors on a single path that has the highest probability given a multipath scenario. From the results, the SA algorithm produces results with generally smaller overlapping areas, which increases the probability of detection. In contrast to the heuristic placement, the SA algorithm is able to explore the whole feasible region and place sensors over multiple paths in a multipath scenario. The uncertainty of the target in this scenario is much smaller, with a higher concentration of probability density peaked near the center of path when σ^2 is small. These distinct peaks allow the objective function value surface to be more distinct over the S space.



(a) Heuristic Sensor 1

(b) Heuristic Sensor 2

(c) SA Sensor 1

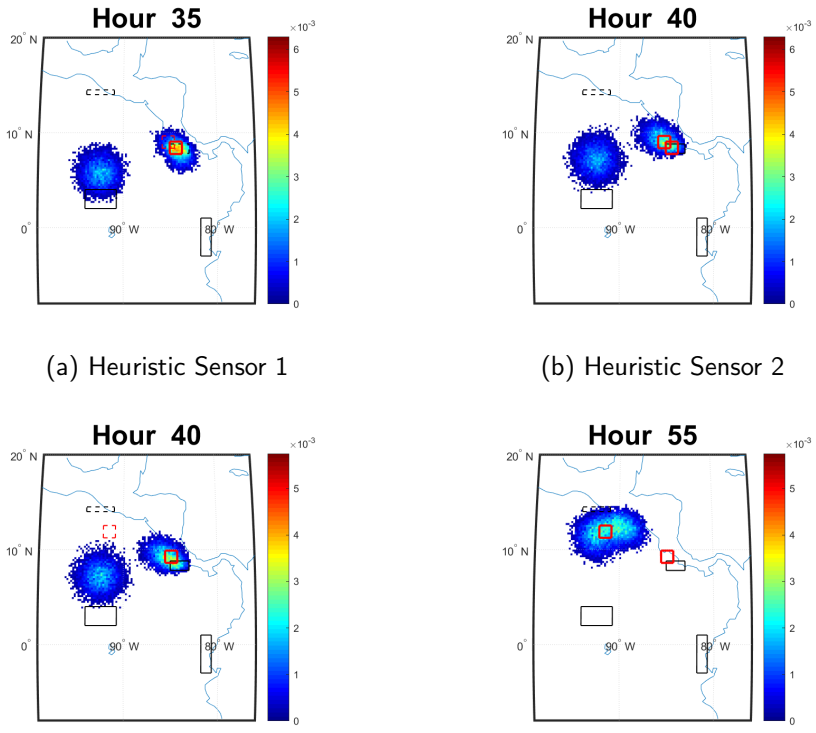
(d) SA Sensor 2

Figure 5.6. Locations of sensors from heuristic (a,b) and SA optimization (c,d) obtained from the maximum ratio extreme case, ratio = 2.497.

Heuristic sensors are active at $t = 35, 40$, as shown by the change of dotted red boxes to solid red boxes. Similarly, SA sensors are turned on at $t = 20, 55$.

Subsequently, we investigate a lower extreme case where the ratio is less than 1, specifically, when $N = 2$, $\sigma^2 = 0.38$, sensor size is 1.3, number of paths is two, and "Time Optimized" is on. The heat maps for the heuristic and SA results are shown in Figure 5.7. We observe that although the sensors are placed on distinct paths and visually well-placed, the ratio obtained has an average value of 0.765.

The large variance produces a probability density profile with a larger spread around its expected location, and combined with the small sensors, we can deduce that the objective function value surface in the multidimensional space S is smoother than the lower variance case, with its probability values closer to zero. With closer investigation of the individual cases with extremely low ratios, we see the probability of detection from the heuristic is less



(a) Heuristic Sensor 1

(b) Heuristic Sensor 2

(c) SA Sensor 1

(d) SA Sensor 2

Figure 5.7. Locations of sensors from heuristic (a,b) and SA optimization (c,d) obtained from the minimum ratio extreme case, ratio= 0.765.

Heuristic sensors are active at $t = 35, 40$, as shown by the change of dotted red boxes to solid red boxes. Similarly, SA sensors are turned on at $t = 40, 55$.

than 0.25 in all cases. A smooth objective function value surface implies that the algorithm has to be sensitive enough to detect and accept small changes of improvement. Given that the objective function values are low and the lower bound of the objective function value is zero, however, it is easier for the algorithm to accept poorer objective values close to zero with a higher probability based on the Metropolis Criterion described by Equation 4.5. As the Metropolis acceptance criterion is based on $f(V^{(k+1)}) - f(Q^{(k)})$, when $f(Q^{(k)})$ is small and close to the lower bound value of zero, the difference becomes sensitive to small variations of $f(V^{(k+1)})$. The actual difference in objective function value between the old and new point may be overwhelmed by the randomness of the simulation.

In this case, the algorithm may converge to a more precise solution with a better objective

function value given a larger number of iterations, whereby the algorithm accepts only improving objective values and samples more frequently around its current best known solution as the sampling distribution converges to the Boltzmann distribution.

5.2.3 Algorithm Run-Time

The run-time of the SA algorithm is also used to evaluate its performance. We recorded the algorithm run-time, measured in seconds, to investigate the effects of the five parameters during the same runs described in Table 5.2. Since our maximum run length l is proportional to the number of sensors N as described in Equation 4.20, the run-time is normalized accordingly by dividing the recorded times by their respective N.

A stepwise algorithm is used to find the effects of the factors, up to polynomial degree 2 and factorial level of 2 to the normalized run-time. The regression results show that the normalized run-time increases linearly with the number of sensors, as shown in Figure 5.8. The other factors did not significantly affect the run-time of the algorithm significantly. This can be explained by looking at the step-size algorithm of SA. During each iteration, the algorithm searches for a new point q_i (for each i) by simulating the direction vector d_i, scale factor λ_i, and going through the reflection logic in each subroutine. The number of processes increases proportionately with N and is relatively intensive computationally, thus dictating the main processing time of the algorithm. We therefore can also conclude that the run-time of the normalized SA algorithm (relative to the linear stopping time) is at least of order N.

5.2.4 SA Algorithm Initialized with Heuristic Results

In the previous sections of this chapter, we initialized the SA algorithm by assigning the initial point $Q^{(0)}$ approximately uniformly in S. Since the heuristic results are the lower bounds of the optimal solution value, we initialize the SA algorithm with the heuristic results in this experiment. This is done by assigning $Q^{(0)} = Q_h$, $\hat{Z} = Z_h$ and $\hat{Q} = Q_h$ in the initialization step of the SA algorithm described in Section 4.1. The same measure of performance, the probability ratio described in Equation 5.1, is used to evaluate the performance of the SA algorithm while varying the factors described in Table 5.1 using the same design points obtained from the NOLH.

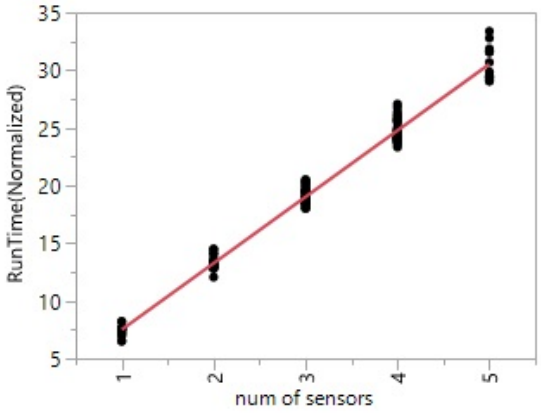


Figure 5.8. Normalized run-time regression results.

From the partition tree in Figure 5.9, we observe that the splits are almost similar to the case when the initialization is done arbitrarily. In the cases where the ratio value is statistically higher than 1, the performance of SA after heuristic initialization is comparable to the previous results. This implies that the algorithm performance is insensitive to initialization inputs and is sufficiently robust to reach the estimated optimal value in these cases. We note that, in this case, the minimum value of the detection ratio obtained is expected to be 1 for the cases where SA algorithm can find no better solution than the heuristic results, i.e., $Z_{SA} = Z_h$. Therefore, we should not be expecting any probability ratio value smaller than 1. There are two scenarios in this experiment when the ratio is very close to 1. The first case is when $\sigma^2 \geq 0.19$ and the sensor size ≤ 1.9, with a 95% CI for the ratio of (1.002, 1.037). The other is when $\sigma^2 \geq 0.19$ and the sensor size ≤ 1.9 in a single-path scenario, where the 95% CI of the ratio is (1.017, 1.056). These results suggest that the algorithm was not able to find a better solution than the heuristic results even when the algorithm is initialized with the heuristic solution. Therefore, instead of the SA algorithm, it is more efficient in these cases to use the heuristic results that can be obtained with a shorter computational run-time.

The performance of the SA algorithm with heuristic initialization performed equally well in terms of optimizing with and without time optimization. Figure 5.10 shows the histogram results of the probability ratio against "Time Optimized" off and on, respectively, and there is no significance difference compared to the case when initialization is done arbitrarily in Figure 5.5. Comparing the 95% CI of (1.14, 1.22) for the off cases with initialization and (1.11, 1.20) without initialization, the performance without time optimization is similar

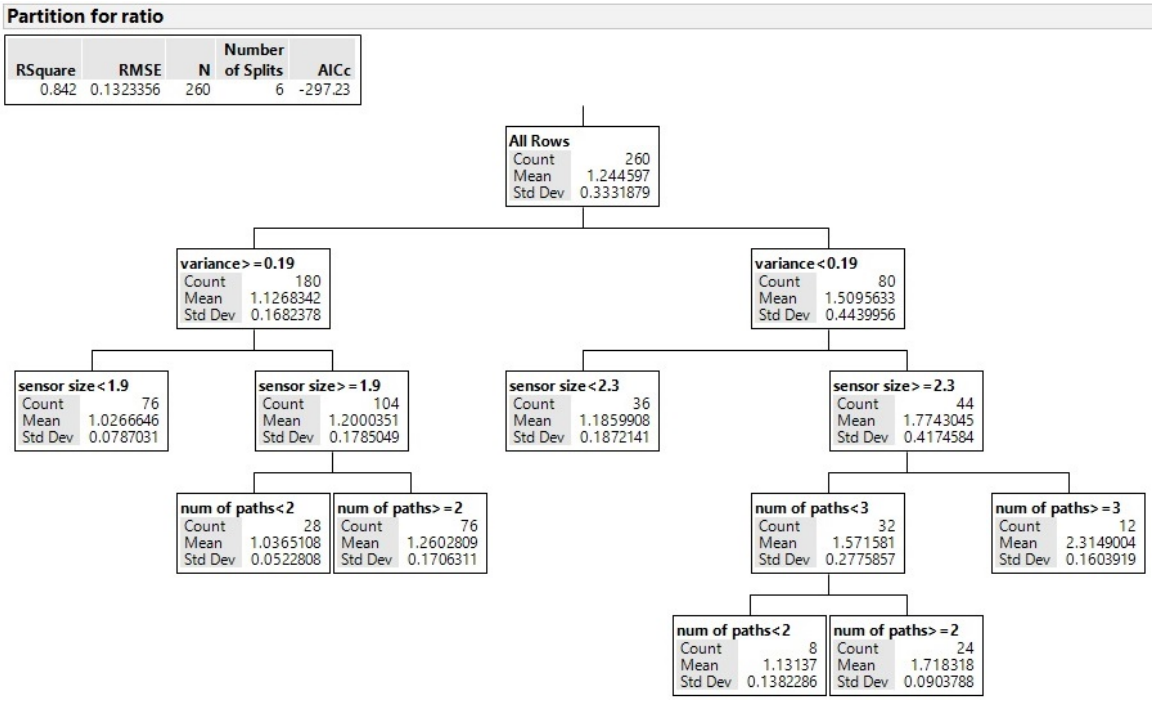


Figure 5.9. Partition tree of ratio after initializing SA with heuristic results.

with overlapping CIs. In this case, when the sensor placement is optimized over time, the SA algorithm performs 30 percent better than the heuristic results on average, with an overlapping 95% CI of (1.24, 1.37) with initialization and (1.22, 1.37) without initialization.

5.3 Scenario Experimental Design

Based on the Central American narcotics smuggling scenario described in Section 1.3, we look for a patrol team deployment configuration that can maximize the detection probability while minimizing resources. The deployment configurations are sea patrol routes that can be employed to detect the traffickers. In the experiment, we assume a static deployment strategy, which implies that the patrol teams are deployed prior to the start of the mission and will keep their positions regardless of future intelligence input. It is also assumed that the sensors on the patrol ships will only turn on for one instance in the assigned time period.

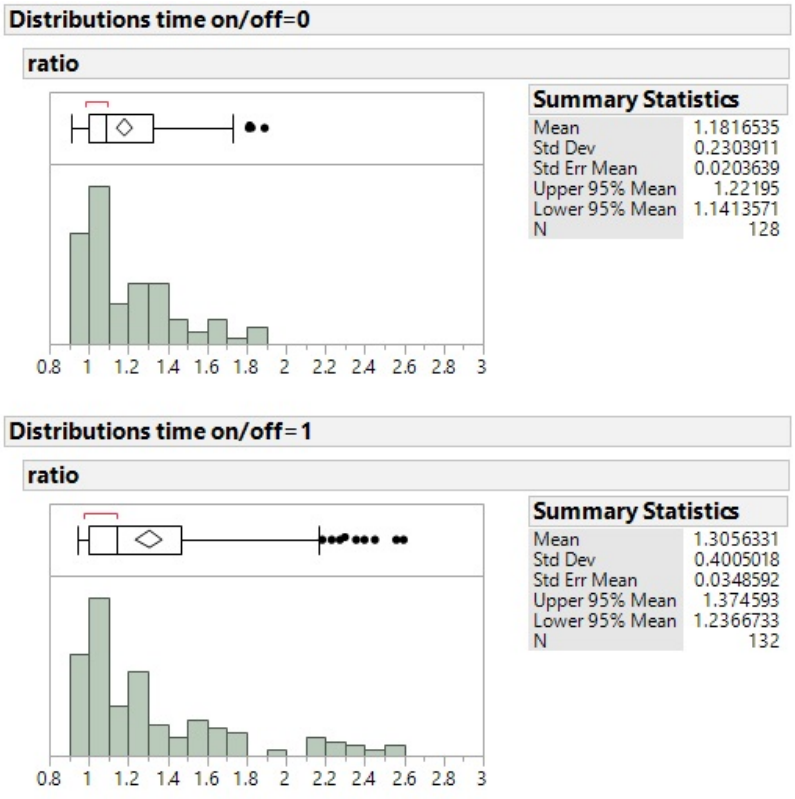


Figure 5.10. Histogram of the ratio with Time Optimization off (top) and on (bottom) after initializing the SA algorithm with heuristic results.

5.3.1 Scenario Modeling

The trafficker location is modeled using the TBBM to produce its temporal probability heat map. Two instances of the heap map are shown in Figure 2.3. Based on intelligence, we know the traffickers will be moving from the borders of Ecuador and Colombia at around $t = 0$ hours and reach the borders of Costa Rica or Mexico at $t = 70$ hours approximately. We assume a choice of one, two, three or four patrol teams that can be deployed for a particular mission, between $15 \leq t \leq 55$ hours. The patrol troop's search area is modeled as square sensors, keeping a constraint of total patrol area of 9 square degrees with varying sensor sizes. The four deployment strategy alternatives consist of four sensors of 2.25 square degrees each, three sensors of 3 square degrees each, two sensors of 4.5 square degrees each, and one sensor of 9 square degrees. To simulate information uncertainty regarding target movement and weather conditions, the variance parameter for the TBBM acts as a noise factor to the simulation and is varied between $0.1 \leq \sigma^2 \leq 0.3$. Four replications are

performed on each of the 44 experimental design points, summarized in Table 5.2.

Table 5.2. Experimental design to evaluate best sensor deployment for Central American narcotics smuggling scenario.

Design #	N	Sensor Size	Noise σ^2
1	1	3	$0.1 \leq \sigma^2 \leq 0.3$ at 0.02 interval
2	2	$\sqrt{4.5}$	
3	3	$\sqrt{3}$	
4	4	1.5	

5.3.2 Best Sensor Deployment

The simulated data is analyzed and the box plot is as shown in Figure 5.11. Although we observe that the median and quantiles are higher in the two sensor deployment strategy, we are unable to draw conclusions based on visual inspection.

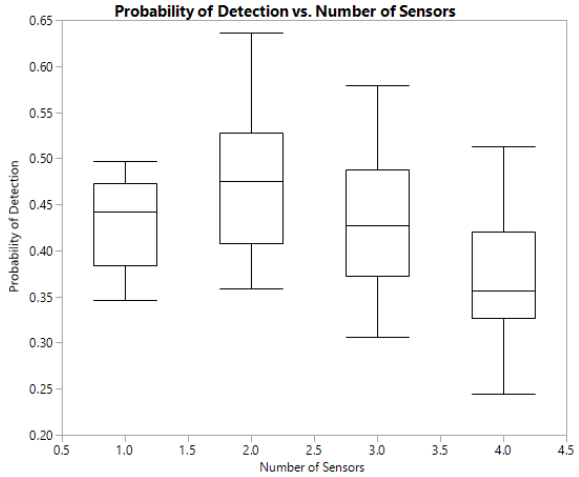


Figure 5.11. Box plot of probability of detection against number of sensors.

We employ a mathematical methodology to select the best sensor deployment strategy. Sanchez (2018) discussed the use of the quadratic loss function that provides a quantitative measurement for selecting the best option given the mean and standard deviation of the data distribution, given by minimizing

$$E_{\Omega}[l(y_x)] = c\left[\sigma_{y_x}^2 + (\mu_{y_x} - \tau)^2\right] \tag{5.2}$$

over the noise space Ω, where y_x is the performance measure, μ_{y_x} is the mean of y_x, $\sigma^2_{y_x}$ is the variance of y_x, c is a constant cost factor, and τ is the target value. The target value is the criterion value that the user tries to achieve and is used as a benchmark to compare the precision and accuracy of the observed results. The equation defines loss as the penalty incurred to the system when the performance y_x deviates from the target value τ. We can observe that the penalty is high when the y_x obtained lacks accuracy, which increases $(\mu_{y_x} - \tau)^2$, or lacks precision, which increases $\sigma^2_{y_x}$. In this thesis, the concept of this criterion is adopted to measure the robustness of the estimated optimum. We can think of the variability in the heat map probability due to the variance as the noise factor affecting the detection probability, and the objective is to maximize the detection probability while considering the variability.

For each design point in Table 5.2, we generate four replications to obtain the mean, μ_Z and standard deviation, σ_Z of the probability of detection. The expected loss at the design point is evaluated as follows:

$$E[l(Z)] = {\sigma_Z}^2 + (\mu_Z - 1)^2, \tag{5.3}$$

where the target is set to 1 as we want to achieve 100 percent probability of detection. The lowest loss value out of the four design points in Table 5.2 has the best sensor deployment configuration amongst them.

Table 5.3. Experimental results for best sensor deployment over $0.1 \leq \sigma^2 \leq 0.3$ using expected loss values.

N	μ_Z	σ_Z	Lower 95% CI	Upper 95% CI	E(Loss)
1	0.435	0.0436	0.348	0.522	0.321
2	0.488	0.0799	0.328	0.648	0.269
3	0.434	0.0732	0.287	0.580	0.326
4	0.379	0.0662	0.246	0.511	0.390

Using Equation 5.3, the loss values of the respective test points are tabulated in Table 5.3. We note that the means of the results are close; therefore, we are unable to find the best configuration using CIs. For instance, the 95% CI of the probability of detection is $(0.348, 0.522)$ for $N = 1$ and $(0.328, 0.648)$ for $N = 2$. Since the CIs overlap, we therefore cannot conclude that both cases are significantly different from each other. However,

using the loss function method, we can conclude from the loss values that the two-sensor deployment strategy having the smallest loss value has the most robust solution to achieve a good probability of detection consistently, with a mean probability of detection at 0.488. A snapshot of the optimized two-sensor deployment configuration is shown in Figure 5.12, at $\sigma^2 = 0.2$.

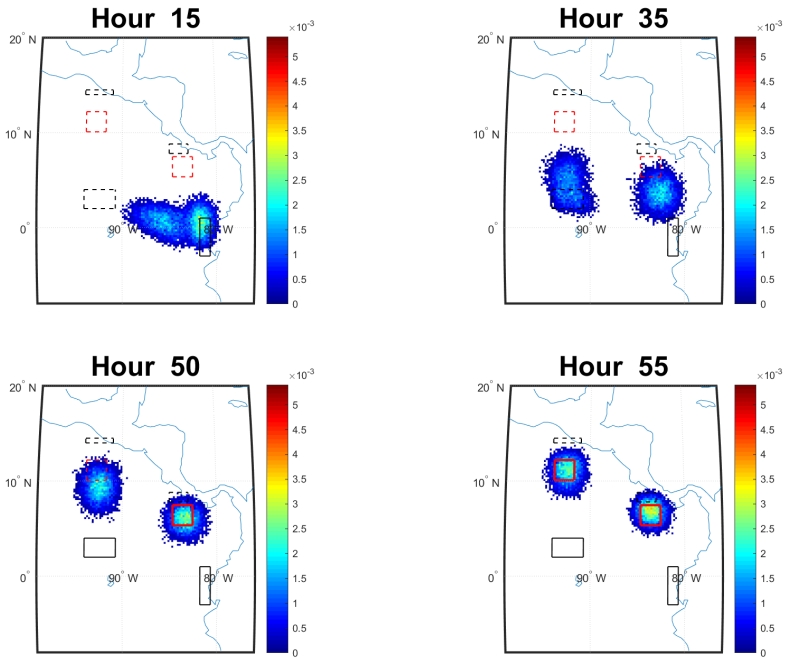


Figure 5.12. Snapshot of best sensor deployment for the narcotics smuggling scenario using two sensors, at $\sigma^2 = 0.3$.

In this scenario, $Z_{SA} = 0.596$. Sensors are turned on at $t = 50, 55$, as shown by the change of dotted red boxes to solid red boxes.

To verify that the loss function is a robust solution for sensor deployment selection, we investigate the cases where $\sigma^2 = 0.1, 0.2, 0.3$ separately with 30 replications each, obtaining 30 distinct sensor deployment solutions for each variance value. The box plot as shown in Figure 5.13 has a more obvious trend where the expected probability of detection peaks near the points when two sensors are deployed.

The numerical results are tabulated in Table 5.4. By comparing the 95% CIs of the results by each variance level, we can deduce the following observations. When σ^2 is low at 0.1, we observe that $N = 2, 3$ has a better probability of detection than when $N = 1, 4$. When

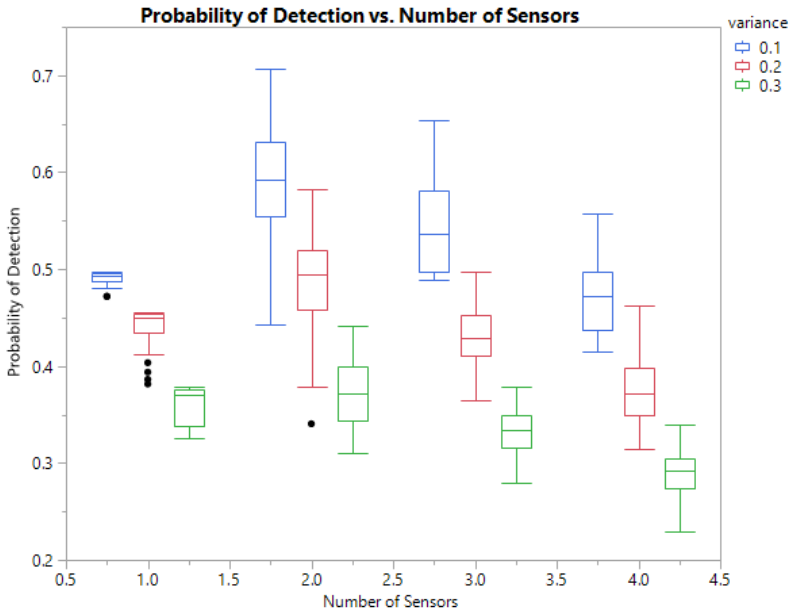


Figure 5.13. Box plot of probability of detection against number of sensors and variance.

$\sigma^2 = 0.2$, the two sensor configuration has the best probability of detection. Finally when σ^2 is at 0.3, using either a one or two sensor configuration in this scenario will yield better results than deploying three or four smaller sensors. In all cases for σ^2, the probability of detection is highest when the two-sensor configuration is deployed. This conclusion is analogous to the case when we used fewer replications and evaluated the most robust solution using the loss function methodology.

5.3.3 Observations

Throughout the experiments, we observe variability in the results obtained using SA optimization. We expect the variability since there are multiple levels of stochasticity in this simulation-based optimization process. The high variability of the SA algorithm is notable from Figure 5.13, especially in the case where $N = 2, 3, 4$. In this case, the increase in the number of sensors also increases the dimensionality and complexity of the problem. Although the algorithm is able to approach the global optima using the iteration count as a stopping condition, the algorithm may not have sufficient time to converge to a precise solution. As discussed in Chapter 3, the SA algorithm has proven convergence properties when the number of iterations approaches infinity. Therefore, the use of iteration count

Table 5.4. Experimental results for best sensor deployment for $\sigma^2 = 0.1, 0.2, 0.3$ with 30 replications at each design point.

σ^2	N	μ_Z	σ_Z	Lower 95% CI	Upper 95% CI
0.1	1	0.491	0.00672	0.488	0.493
	2	0.591	0.0620	0.569	0.613
	3	0.543	0.0444	0.527	0.559
	4	0.470	0.0366	0.456	0.483
0.2	1	0.440	0.0218	0.432	0.447
	2	0.484	0.0587	0.463	0.505
	3	0.432	0.0307	0.421	0.443
	4	0.375	0.0360	0.362	0.388
0.3	1	0.360	0.0194	0.353	0.366
	2	0.373	0.0377	0.360	0.389
	3	0.332	0.0250	0.323	0.341
	4	0.288	0.0274	0.279	0.299

as a stopping condition may be the cause of the reduction in precision. By altering the stopping condition to some other criterion, we can increase the number of maximum iterations to further reduce the variability of the solution without significantly increasing the overall computational time. As such stopping conditions based on algorithm output are not well-defined for simulated annealing, this is a field for future exploration and improvement.

CHAPTER 6:
Conclusion and Recommendations

Given the incomplete information about moving targets, an effective way to deploy sensors can be determined using a simulation approach. Simulation methods are able to generate probability heat maps that represent the randomness of target movement and location, and are able to incorporate algorithms to find an approximate optimal sensor placement configuration. This thesis focuses on the Simulated Annealing (SA) algorithm, a random search algorithm that is able to analyze probability heat maps generated using the Temporal Brownian Bridge Model (TBBM) and to optimize sensor placement to maximize the overall target detection probability.

6.1 Summary

This thesis discussed the basics of modeling target movement using the assumption of Brownian motion, and using the TBBM to generate probability heat maps given uncertain intelligence. With the TBBM as the underlying stochastic model and given that the sensors are placed prior to the beginning of the target's journey without input based on ongoing information, this thesis employed the SA algorithm to determine the optimal sensor placement strategy that maximizes the overall probability of target detection.

The SA algorithm was inspired by the annealing process of metal, where the metal starts with a high initial temperature and is slowly cooled to form a new lattice structure. Similarly, the temperature parameter of the SA algorithm is the key difference between this algorithm and other random search algorithms, in that it can accept non-improving points during its search for the optimal objective value. During its initial iterations when the temperature parameter is high, the algorithm readily accepts non-improving points to explore the whole sample space. The temperature parameter is slowly decreased and the algorithm reduces the probability of accepting non-improving points. This thesis reviewed the basics of the SA algorithm, and described the temperature cooling schedules, point generators, and point acceptance criteria used in the algorithm. Next, the thesis elaborated on the modifications made to the SA algorithm that can be applied to the TBBM to optimize the sensor deployment strategy for a target detection scenario. In the modified SA algorithm, the adaptive cooling

schedule included a geometric reduction factor to ensure convergence, and a step-size algorithm with a point generator incorporated a reflection generator to simulate uniformly distributed candidate points in the solution space.

Based on the modified SA algorithm, the second portion of the thesis evaluated the performance of the algorithm against a heuristic approach, and analyzed the computational run-time of the algorithm. The heuristic approach provides a baseline for comparison with the SA algorithm, where the heuristic sensors are centered at the peaks of the probability heat map. The experimental results showed that during cases where the variance parameter is small and sensor size is large, the SA algorithm tends to outperform the heuristic approach. In the contrary cases, the results of the SA algorithm are comparable, or sub-optimal, compared to the heuristic approach. In these cases, the heuristic approach should be implemented to obtain similar results with less computational effort. In terms of computational effort, given the same number of iterations per run, it was found that the normalized computational run-time for the SA algorithm is linearly proportional to the number of sensors during optimization.

The final portion of this thesis concluded by optimizing the sensor placement for the narcotics smuggling scenario in Central America, given a fixed amount of total search area allocated. The base TBBM model split the possible paths of the target into two distinct paths to simulate the given heat map information. For the same overall search effort, the search areas were broken into various numbers of smaller sensors ranging from one to four sensors. Next, the loss function was employed as an evaluation tool to obtain the most robust solution. It was shown that the best sensor deployment strategy in the drug trafficking scenario is to deploy two separate sensors to maximize the overall probability of detection of the target. This could be because there were only two paths in the scenario chosen, and thus it was most efficient to place one sensor on each path, but there are diminishing returns to having additional sensors.

6.2 Future Work

Subsequent research can modify and improve the convergence of the SA algorithm when the objective function value is small for this particular implementation of target detection. Given unlimited computational power, more analysis can be done to analyze the convergence

properties of the modified SA algorithm. Further studies can implement other stopping criteria, thus allowing the algorithm to have longer runs for possible optimal estimates with better precision.

With the SA algorithm established for static sensor placement, future improvements can work to increase accuracy based on live intelligence input. A dynamic sensor placement strategy would optimize individual sensors in real-time as information from past sensors is collected. Previous work from Nunez (2017) and Cheng (2016) have already improved the TBBM to incorporate intelligence updates in the heat maps. Using simulated annealing to determine a dynamic sensor placement strategy would potentially allow for an improved probability of detection given information from prior sensor observations.

THIS PAGE INTENTIONALLY LEFT BLANK

List of References

Atkinson MP, Singham DI (2015) Multidimensional hitting time results for Brownian bridges with moving hyperplanar boundaries. *Statistics & Probability Letters* 100:85–92.

Ben-Yoash R (2016) *Anti-submarine warfare search models*. Master's thesis, Department of Operations Research, Naval Postgraduate School, Monterey, CA.

Borderland Beat (2014) The US is only capable of stopping 20% of illegal drugs from Latin America. Accessed April 7, 2018, http://www.borderlandbeat.com/2014/03/the-us-is-only-capable-of-stopping-20.html.

Branch S (2012) Design and evaluation of new intelligent sensor placement algorithm to improve coverage problem in wireless sensor networks. *Journal of Basic and Applied Scientific Research* 2(2):1431–1440.

Bunnefeld N, Börger L, van Moorter B, Rolandsen CM, Dettki H, Solberg EJ, Ericsson G (2011) A model-driven approach to quantify migration patterns: Individual, regional and yearly differences. *Journal of Animal Ecology* 80(2):466–476, http://www.jstor.org/stable/41059078.

Campos SI (2014) *An analysis of mathematical models to improve counter-drug smuggling operations*. Master's thesis, Department of Operations Research, Naval Postgraduate School, Monterey, CA, https://calhoun.nps.edu/handle/10945/43885.

Chen TS (1990) *Simulated annealing in sonar track detection*. Master's thesis, Department of Electrical and Computer Engineering, Naval Postgraduate School, Monterey, CA, https://calhoun.nps.edu/handle/10945/27563.

Cheng CC (2016) *A Brownian bridge movement model to track mobile targets*. Master's thesis, Department of Operations Research, Naval Postgraduate School, Monterey, CA, https://calhoun.nps.edu/handle/10945/50522.

Chew MC (1973) Optimal stopping in a discrete search problem. *Operations Research* 21(3):741–747, http://www.jstor.org/stable/169382.

Goldberg DE (1989) *Genetic Algorithms in Search, Optimization and Machine Learning* (Boston: Addison-Wesley Longman Publishing Co.), 1st edition.

Henderson D, Jacobson SH, Johnson AW (2003) *The Theory and Practice of Simulated Annealing* (Boston: Springer US), ISBN 978-0-306-48056-0, https://doi.org/10.1007/0-306-48056-5_10.

Horne JS, Garton EO, Krone SM, Lewis JS (2007) Analyzing animal movements using Brownian bridges. *Ecology* 88(9):2354–2363, http://www.jstor.org/stable/27651373.

Isacson A (2017) Trafficking routes over time. Accessed April 7, 2018, https://adamisacson.com/trafficking-routes-over-time/.

Jawahar N, Gunasekaran A, Balaji N (2012) A simulated annealing algorithm to the multi-period fixed charge distribution problem associated with backorder and inventory. *International Journal of Production Research* 50(9):2533 – 2554.

Kindl MR, Rowe NC (2012) Evaluating Simulated Annealing for the Weighted-Region Path-Planning Problem. *2012 26th International Conference on Advanced Information Networking and Applications Workshops*, 926–931, https://doi.org/10.1109/WAINA.2012.30.

Kroese DP, Porotsky S, Rubinstein RY (2006) The Cross-Entropy Method for Continuous Multi-Extremal Optimization. *Methodology & Computing in Applied Probability* 8(3):383 – 407, http://libproxy.nps.edu/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=bth&AN=22987404&site=ehost-live&scope=site.

Liu SJ, Chen CY, Tsai CW (2017) An effective simulated annealing for influence maximization problem of online social networks. *Procedia Computer Science* 113:478–483.

Macwan A, Nejat G, Benhabib B (2011) Target-motion prediction for robotic search and rescue in wilderness environments. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 41(5):1287–1298.

Mooshegian MS (2013) *A probabilistic model of illegal drug trafficking operations in the Eastern Pacific and Caribbean Sea*. Master's thesis, Department of Operations Research, Naval Postgraduate School, Monterey, CA, https://calhoun.nps.edu/handle/10945/37681.

Nunez JA (2017) *Particle filtering methods for incorporating intelligence updates*. Master's thesis, Department of Operations Research, Naval Postgraduate School, Monterey, CA, https://calhoun.nps.edu/handle/10945/53026.

Nunn L (1981) *An Introduction to the Literature of Search Theory* (Alexandria, VA: Center for Naval Analyses).

Pietz J, Royset JO (2013) Generalized orienteering problem with resource dependent rewards. *Naval Research Logistics* 60:294–312.

Raichlen DA, Wood BM, Gordon AD, Mabulla AZP, Marlowe FW, Pontzer H (2014) Evidence of Lévy walk foraging patterns in human hunter–gatherers. *Proceedings of the National Academy of Sciences* 111(2):728–733, http://www.pnas.org/content/111/2/728.

Sanchez S (2011) NOLH design spreadsheet. Accessed May 20, 2018, http://harvest.nps.edu/.

Sanchez S (2018) Robust design philosophy. Lecture, Simulation Analysis, Feb 27, Department of Operations Research, Naval Postgraduate School, Monterey, CA.

United States Senate Caucus on International Narcotics Control 102nd Congress 2nd session (2012) Preventing a Security Crisis in the Caribbean. Https://www.drugcaucus.senate.gov/sites/default/files/caribbean%20drug%20report.pdf.

Washburn AR (2014) *Search and Detection* (CreateSpace Independent Publishing Platform).

Wei L, Zhang Z, Zhang D, Leung SC (2018) A simulated annealing algorithm for the capacitated vehicle routing problem with two-dimensional loading constraints. *European Journal of Operational Research* 265(3):843–859.

Yu VF, Lin SY (2015) Solving the location-routing problem with simultaneous pickup and delivery by simulated annealing. *International Journal of Production Research* 54(2):1–24.

Zabinsky ZB (2003) *Stochastic Adaptive Search for Global Optimization* (Boston: Springer).

THIS PAGE INTENTIONALLY LEFT BLANK

Initial Distribution List

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California