To What Extent Can Simulation Optimisation be Used in Wildlife Reserve Design?

Shengjie Zhou, B.Sc.(Hons.), M.Sc.

Submitted for the degree of Doctor of Philosophy at Lancaster University.

15th February, 2025

Abstract

Establishing nature reserves is a key method for preventing biodiversity loss. This thesis addresses the reserve site selection (RSS) problem, which aims to select sites for nature reserves to ensure species survival. Specifically, it examines the extent to which simulation optimisation (SO) can be used in the RSS problem. The applicability and effectiveness of SO are evaluated by applying an SO method and its adaptations across three scenarios of a Grey Wolf (*Canis lupus*) RSS problem.

The problem is formulated as a chance-constrained SO problem, with a deterministic objective that minimises conservation costs, subject to a probabilistic species survival constraint. This probability is estimated using a grey wolf simulation model that simulates the wolves' birth, growth, dispersal and death in discrete time steps. The problem is solved using the sequential feasibility test procedure from Hong, Luo, and Nelson (2015), hereafter CCSB-F.

Three scenarios of the RSS problem, each with different characteristics, are investigated in this research. Scenario 1 demonstrates how CCSB-F can tackle a basic problem. Several observations are made: first, since solution costs are trivial to obtain, computational effort (measured by the number of simulation runs) is required solely for establishing solution feasibility. Second, due to sampling error in simulation results, solution feasibility can only be assured subject to a 'statistical guarantee'. Lastly, the

computational effort required depends on how close the solutions are to the feasibility boundary and the required level of statistical guarantee.

To address likely computational hurdles in more difficult versions of this problem, Scenarios 2 and 3 are designed to demonstrate and evaluate two solution space filtering approaches. The first approach 'temporarily removes' solutions with equivalent alternatives, identified based on the simulation model, without affecting CCSB-F's statistical guarantee. Applying this approach to Scenario 2 (28 solutions) reduces computational effort by approximately 26% compared to using CCSB-F alone. When applied to Scenario 1 (28 solutions), it achieves an estimated savings of 40% while maintaining the same level of statistical guarantee.

The second approach is a heuristic that uses expert knowledge to create solution dominance rules and then removes dominated solutions before applying CCSB-F. It reduces computational effort by approximately 80% in Scenario 3 (210 solutions) compared to using CCSB-F alone. When applied to Scenario 1 (28 solutions), the estimated computational savings is 60%. Even though it cannot guarantee to find the best solution in the entire solution space because it removes solutions, it still provides a statistical guarantee on the filtered solution space and the feasibility of the selected solutions. Although these estimates represent conservative lower bounds (as they do not fully account for the additional reduction in the number of replications per solution), they clearly demonstrate the potential of the proposed approaches to significantly reduce computational effort.

Acknowledgements

First and foremost, I would like to thank my supervisors, David Worthington, Richard Williams, and Luke Rhodes-Leader, for their invaluable support throughout my PhD journey. To Richard, thank you for your support and for generously sharing your knowledge and experiences. To Luke, thank you for providing your insights and for all the 9 a.m. meetings that kept me on track. I am especially grateful to David for your continuous support during both my MSc and PhD. Thank you for all the advice and support you have patiently given to guide and teach me how to become a researcher.

I would also like to thank Roger Brooks for his supervision during the first three and a half years of my PhD (before his retirement). I am grateful for the time and effort he dedicated to supporting me during that period, and for his kindness.

I am also grateful to the Management Science Department. Without its financial support, this work would not have been possible. I would like to extend my thanks to the staff in the department for creating a very friendly environment. Additionally, I would like to thank all the PhD students in the department during my time there. Thank you all for making my PhD journey enjoyable.

I also want to express my gratitude to my partner, Liam, for taking care of me, encouraging me, and engaging in meaningful discussions about life and academics.

Finally and most importantly, I would like to thank my mother for her unconditional and ever-present love, understanding, and support throughout my life. Without you, I would not have achieved any of what I have. I cannot thank you more.

Declaration

I declare that the work in this thesis is my own work. The material has not been submitted in whole for the award of any other degree, in Lancaster or elsewhere. An early version of this work has been presented at two conferences in the form of posters. The abstracts of these posters have been published in:

S. Zhou et al. (2023). "To What Extent Can Simulation Optimisation be Used in Wildlife Conservation?" In: *Proceedings of the Operational Research Society Simulation Workshop 2023 (SW23)*. Ed. by C. Currie and L. Rhodes-Leader. The Operational Research Society. Southampton, UK, p. 285

S. Zhou (2023). "To What Extent Can Simulation Optimization be Used in Wildlife Reserve Design?" In: *Proceedings of the 2023 Winter Simulation Conference (Archive)*. Ed. by C. G. Corlu et al. Texas, USA: IEEE Press. URL: https://informs-sim.org/wsc23papers/doc127.pdf

The approximate word count of this thesis is 28,724 words.

Shengjie Zhou

Contents

List of Figures

List of Tables

List of Abbreviations

RSS reserve site selection

OR operational research

MILP mixed integer linear programme

SO simulation optimisation

RnS ranking and selection

PCS probability of correct selection

IZ indifference zone

CCSB chance-constrained selection of the best

CCSB-F sequential feasibility test of procedure CCSB

Pr probability

MBA model-based approach (MBA)

M-CCSB-F MBA coupled with CCSB-F

RBHA rule-based heuristic approach

H-CCSB-F RBHA coupled with CCSB-F

Chapter 1

Introduction

The International Union for Conservation of Nature (IUCN) reports that over 42,100 species out of the 150,300 evaluated, face extinction (IUCN, 2023). This extensive biodiversity loss threatens not only species' existence but also the stability of ecosystems and the vital services they offer, including climate regulation, water quality maintenance, and wildlife habitat provision (IPBES, 2019).

To counteract biodiversity loss, establishing protected areas (also known as nature reserves, hereafter reserves) has emerged as a key global strategy (Langhammer et al., 2024). Studies have shown that establishing reserves helps to reduce rates of habitat loss (Joppa and Pfaff, 2011), maintain species populations (Taylor et al., 2011), and when appropriately located, may slow the rate at which species are driven towards extinction (Butchart et al., 2012).

However, the establishment of nature reserves is not without challenges. It necessitates significant resources, such as land and financial capital, which are often in limited supply. Additionally, allocating these resources for conservation purposes can conflict with other economic interests like agriculture, housing, industry, and tourism (Billionnet, 2013). Moreover, areas of the highest biological value are typically more financially, socially and politically costly (Luck, 2007). To address these challenges comes the science of reserve design.

1.1 Reserve Design

Reserve design is an interdisciplinary science that combines principles from subjects like ecology, economics, and operational research (OR) (Kingsland, 2002). Simulation and mathematical optimisation (hereafter optimisation) are two primary OR methods used in reserve design.

Simulation models in reserve design are typically used to evaluate management options (e.g. Haight et al. (2002)), understand the ecosystems (e.g. Possingham and Davies (1995)), or to evaluate the solutions selected by an optimisation algorithm (e.g. Meester et al. (2004)). Typical simulation models in reserve design are Monte Carlo (e.g. Haight and Travis (1997)), agent-based (e.g. Miller et al. (2014)), discrete-event (e.g. Gaucherel et al. (2021) and discrete-time (also known among the ecologists as individual-based models, e.g. Fahrig (2001)).

Optimisation is typically used in reserve design to identify which areas of land should be protected to achieve conservation goals, a problem known as the reserve site selection (RSS) problem (e.g. Camm et al. (2002)).

1.2 Reserve Site Selection

RSS is one of the key topics in reserve design. In the context of RSS, the study area is divided into small areas, which are called sites. The RSS problem aims to select a subset of sites amongst a set of potential sites to assemble a reserve to achieve conservation goals (Billionnet, 2013).

The analytical methods to solve this problem borrow heavily from OR (Snyder and Haight, 2016). Optimisation is the primary OR method used in RSS. The RSS problem in optimisation is usually formulated as a mixed integer linear programme (MILP), and aims to select the best (optimal) subset of sites that maximises or minimises certain goals subject to some constraints (Wang, Önal, and Fang, 2018). Two basic formulations are the maximal covering location formulation, where the objective is to maximise the conservation goal subject to a constraint on the number of sites; and the set covering formulation, where the objective is to minimise the number of reserve sites subject to conservation constraints (Williams, ReVelle, and Levin, 2005). The MILP can be solved with either heuristic (does not guarantee to find the global optimum) or exact methods (guaranteed to find the global optimum). Heuristic methods include greedy algorithms (e.g. Clemens, Revelle, and Williams (1999)), simulated annealing (e.g. McDonnell et al. (2002)) and genetic algorithms (e.g. Delmelle, Desjardins, and Deng (2017)). Exact methods include the simplex algorithm and interior point methods (for linear programming formulation), and branch and bound (for integer programming formulation) (see Haight and Snyder (2009) for more detail).

Optimisation is designed to identify the best solutions but often needs to oversimplify the problem. Simulation allows for the evaluation of more detailed representations of ecological systems but is not designed to search for the best solutions. Hence, combining simulation and optimisation seems to be a natural way to model the RSS problem.

1.3 Simulation Optimisation in RSS

The combination of simulation and optimisation is usually referred to as simulation optimisation (SO) in OR. SO is a process of finding the best input variable values to maximise or minimise some performance measure that is estimated using simulation (Jian and Henderson, 2015).

SO has the potential to overcome the limitations of both optimisation and simulation by integrating the strengths of both methods. However, running complex ecosystem simulation models can be time-consuming, and combining optimisation with simulation can increase the computational burden dramatically, making this approach much less common (Fulton et al., 2015).

Haight and Travis (1997) and Haight and Travis (2008) are two attempts to solve RSS using a SO approach. The former minimises the costs of reserve sites subject to a probabilistic species population size constraint, which is estimated with Monte Carlo simulation. The latter maximises species persistence (long-term survival) probability, which is estimated by a discrete-time simulation model, with a constraint on the cost of sites. However, neither solution method guarantees to find the optimal solution, and the solution method used by Haight and Travis (1997) does not guarantee that the optimal solution satisfies the constraints.

Hence, this thesis aims to develop and evaluate SO methods for the RSS problem that ensure solution accuracy (i.e. provide a 'statistical guarantee' of finding the optimal solution). Solving the RSS problem with SO (Haight and Travis (1997) and Haight and Travis (2008)) is already computationally intense, and ensuring solution accuracy could further increase this computational intensity (e.g. Boesel, Nelson, and Ishii (2003)).

The species studied as an example is the grey wolf (*Canis lupus*). Even though the grey wolf is currently listed as a species of "Least Concern" on the IUCN Red List (IUCN, 2023), it was nearly driven to extinction in North America from the 1800s to the mid-1900s due to wolf eradication campaigns by European settlers and to rapid environmental changes (Paquet and Carbyn, 2003). Fortunately, human interventions have facilitated remarkable comebacks for wolves in some states in the USA. For example, Wydeven, Deelen, and Heske (2009) documents the recovery of wolves from diverse perspectives in the western Great Lakes region. However, despite these successes, the grey wolf is still listed as endangered and remains a protected species in most states in the USA (U.S. Fish and Wildlife Service, 2022). Governments continue to develop and implement grey wolf recovery plans (e.g. California Department of Fish and Wildlife (2016)) to support their ongoing conservation.

1.4 Thesis Aims

This thesis explores the extent to which SO can be used to solve RSS problems. As mentioned in Section 1.3, the application of SO in RSS problems is still in its infancy, and existing works face two major challenges: a lack of solution accuracy and high computational intensity.

To address the first challenge, this research begins by asking:

RQ1: How well do current SO methods perform (in terms of solution accuracy and computational effort) when applied to RSS problems? Previous studies have formulated RSS problems as SO problems but did not adequately address the accuracy of the selected solutions. This thesis aims to evaluate the performance of SO in solving RSS problems from both solution accuracy and computational effort perspectives.

Having identified the computational effort challenge, there is a need to explore approaches for reducing the computational intensity associated with SO in solving RSS problems. This thesis proposes two approaches for addressing this issue. The first involves looking into the simulation model itself, leading to the second research question:

RQ2: How and to what extent can computational effort in SO be reduced by leveraging the simulation model? By answering RQ2, this thesis aims to investigate how information from the simulation model can be utilised to reduce computational effort without compromising solution quality when using SO to solve RSS problems.

After considering computational effort reduction from a modelling perspective, this research considers a complementary perspective of incorporating domain-specific insights. In this light, this study asks:

RQ3: How and to what extent can expert opinion be used to reduce the computational effort in SO? By answering RQ3, this thesis aims to explore the role of expert knowledge in reducing the computational effort required for SO.

1.5 Thesis Structure

These research questions are examined through a computational study, where different variations of a SO solution method are applied to solve three scenarios of the RSS problem. Each scenario is designed to address a specific aspect of the research questions: Scenario 1 is designed to evaluate the performance of SO (RQ1), Scenario 2 to investigate how leveraging the simulation model can reduce computational effort (RQ2), and Scenario 3 to investigate how incorporating expert opinion can reduce computational effort (RQ3).

This thesis consists of seven chapters. Chapter 2 primarily focuses on reviewing the literature on RSS and SO. The final section of the chapter fully defines the research questions, outlines the work undertaken to address these questions, and introduces three RSS scenarios to be investigated further in Chapters 4, 5, and 6.

Chapter 3 details the mathematical formulation of the RSS problem and the simulation model used to assess the performance of potential solution methods. This provides the foundation for evaluating solution accuracy and computational effort.

Chapter 4 defines the first scenario (Scenario 1) and addresses RQ1 by analysing the performance of a SO solution method that guarantees solution accuracy when applied to Scenario 1.

Chapter 5 defines the second scenario (Scenario 2) and introduces a model-based approach that leverages information from the simulation model. It examines RQ2 by assessing the effectiveness of this approach in solving Scenario 2. Its generalisability is evaluated by applying it to Scenario 1 and estimating the potential computational effort savings.

Chapter 6 defines the third scenario (Scenario 3) and introduces a heuristic approach based on expert opinion. It investigates RQ3 by analysing the performance of the heuristic approach in solving Scenario 3. To investigate its generalisability, the approach is also applied to Scenario 1, where its potential computational effort savings are estimated.

Chapter 7 summarises the research findings, reflects on the potential of SO for solving the RSS problem, compares the heuristic and model-based approaches, examines the generalisability of the two proposed approaches, and suggests directions for future research.

Chapter 2

Literature Review

2.1 Introduction

This chapter reviews relevant literature on RSS and SO. Section 2.2 reviews the literature on optimisation, simulation and SO in RSS, highlights research gaps, and specifies the RSS problem formulation this thesis focuses on. Section 2.3 reviews relevant SO methods, and identifies a solution method to solve the proposed RSS problem. Section 2.4 summarises this chapter, states the research questions and outlines the work undertaken to address the research question.

2.2 Reserve Site Selection

The RSS problem this thesis focuses on considers a study area, which is divided into smaller areas called 'sites'. The problem aims to select a set of sites in the study area so that some conservation goals can be achieved. These possible sets of sites are the possible solutions to the problem.

To review how simulation and optimisation have been applied to the RSS problem, relevant literature is searched using a systematic literature search method (detailed in Appendix A.1). This section reviews relevant literature on the RSS problem. Subsection 2.2.1 reviews the use of optimisation in RSS. Subsection 2.2.2 reviews the use of simulation in RSS. Subsection 2.2.3 details existing attempts to use SO to solve the RSS problem. Subsection 2.2.4 then highlights the research gaps.

2.2.1 Optimisation in RSS

A substantial body of literature formulates the RSS problem as an optimisation problem. This section provides several examples of these studies.

RSS problems are typically formulated as MILPs. The optimisation models of RSS problems mainly aim to minimise economic costs subject to constraints defined by conservation goals, or vice versa. The economic costs are mainly defined by the number or the cost of sites. The conservation goals include spatial requirements (e.g. compactness and connectedness of sites, and shape of the reserve), species diversity, and species survival. See Billionnet (2013) or Cabeza and Moilanen (2001) for a more detailed discussion on different types of conservation goals.

Two basic examples of such optimisation formulations are the set covering problem (SCP) formulation and the maximal covering problem (MCP) formulation. The SCP is a classic integer programming problem first introduced by Berge (1957). One of the earliest examples of formulating the RSS problem as an SCP is Underhill (1994), where the objective is to minimise the number of reserve sites so that all species are covered.

This problem can be mathematically written as follows:

$$\text{Min} \quad \sum_{j \in J} x_j \tag{2.1}$$

$$\text{s.t.} \quad \sum_{j \in J_i} x_j \geq 1 \quad \forall i \in I \tag{2.2}$$

$$x_j \in \{0, 1\} \quad \forall j \in J_i \tag{2.3}$$

where I is the set of species to be covered in the reserve; J is the set of candidate sites; J_i is the set of candidate sites that contain species i; x_j is a binary variable and $x_j = 1$ if site j is selected, 0 otherwise. Other examples of problems formulated in this way include: Sætersdal, Line, and Birks (1993), Bonneau et al. (2018), and Álvarez-Miranda et al. (2021).

The MCP is also a classic integer programming problem, first introduced by Church and ReVelle (1974). One of the first examples of formulating the RSS problem as an MCP is Camm et al. (1996), where the objective is to maximise the number of species covered, subject to a constraint on the number of sites, mathematically written as:

$$\text{Max} \quad \sum_{i \in I} y_i \tag{2.4}$$

$$\text{s.t.} \quad \sum_{j \in J_i} x_j \geq y_i \quad \forall i \in I \tag{2.5}$$

$$\sum_{j \in J} x_j \leq k \tag{2.6}$$

$$x_j \in \{0, 1\} \quad \forall j \in J \tag{2.7}$$

$$y_i \in \{0, 1\} \quad \forall i \in I. \tag{2.8}$$

Here, I, J, J_i, and x_j are defined in the SCP. The MCP can be seen as an extension of SCP, where there is an added constraint on the total number of sites that may be selected (denoted as k), and a binary variable y_i to ensure species coverage, with $y_i = 1$ if species i is represented in the reserve, 0 otherwise. See Snyder and Haight (2016) for a review of the RSS problem in this formulation and typical solution methods.

The numerical methods to solve the RSS problem formulated as a MILP borrow heavily from OR (Snyder and Haight, 2016). The problem is typically solved with either heuristic methods (which do not guarantee finding the global optimum) or exact methods (which are guaranteed to find the global optimum). Examples of heuristic methods used to solve RSS problems include greedy algorithms (e.g. Clemens, Revelle, and Williams (1999)), simulated annealing (e.g. McDonnell et al. (2002)), genetic algorithms (e.g. Delmelle, Desjardins, and Deng (2017)), and metaheuristics (e.g. Bandara and Weerasena (2016)). Exact methods include the simplex algorithm and interior point methods (for linear programming formulation), and branch and bound (for integer programming formulation) (see Haight and Snyder (2009) for more detail).

Other ways of formulating the RSS problem include multi-objective optimisation, dynamic programming and nonlinear programming. Snyder, ReVelle, and Haight (2004) present a multi-objective optimisation formulation that aims to maximise the number of unique land systems represented while minimising the total area of the selected sites, subject to constraints on the number of sites and land types coverage, solved with a multi-objective weighting method. One example of the RSS problem formulated as dynamic programming is Costello and Polasky (2004). They aim to maximise the number of species conserved over a given planning horizon subject to a constraint on the number of sites in each period, solved with a backward induction method. Camm et al. (2002) provide an example of nonlinear binary integer programming, aiming to maximise the expected number of species covered (with species coverage probability estimated using a logistic regression model), subject to a constraint on the number of sites. They approximate the nonlinear problem as a mixed integer linear problem and solve it using a branch and bound algorithm.

2.2.2 Simulation in RSS

Simulation models in RSS are typically used to evaluate optimal solutions selected by optimisation algorithms or to estimate some parameter values used in the optimisation formulation. Readers interested in the application of simulation models in the broader area of designing reserves are referred to Drechsler (2020) for a systematic literature review on this topic.

Examples of models used for evaluation include Costello and Polasky (2004), which uses a Monte Carlo model to compare the performance of the solution produced by their method and with a heuristic method in solving a dynamic RSS problem. Another example is Arthur et al. (2002), which uses a Monte Carlo model to compare the performance of two approaches for solving the RSS problem with probabilistic data.

Examples of models used for estimating parameter values include Hof and Raphael (1997), which uses a discrete-time model for the Northern Spotted Owl to estimate the 'connectivity' function (the population of the sites as a function of the connecting sites' population) and the 'carrying-capacity' function (the population of the sites as a function of the sites selected) to incorporate into the optimisation model. Additionally, the solution from the optimisation model is also evaluated by the same simulation model.

Another example is Haight et al. (2004), which develops a discrete-time model for the San Joaquin kit fox. The model simulates the key life events of foxes, including their reproduction, dispersal, and mortality in discrete time. The model is used to estimate the relationship function between extinction risk and the number of sites. This function is then used as the objective function of the optimisation problem that aims to maximise the survival probability of the foxes subject to a cost constraint.

2.2.3 Simulation Optimisation in RSS

Formulating the RSS problem as a MILP oversimplifies important things, such as the movement of species, ecological interactions like predator-prey relationships, and the stochastic nature of species survival. Simulation models allow for the evaluation of more detailed representations of ecological systems but are not designed to search for the best solutions.

SO has the potential to overcome the limitations of both optimisation and simulation by integrating the strengths of both methods. However, running complex ecosystem simulation models can be time-consuming, and combining optimisation with simulation can increase the computational burden dramatically, making this approach much less common. Two existing attempts of applying SO to the RSS problem are Haight and Travis (1997) and Haight and Travis (2008).

Haight and Travis (1997)'s optimisation objective is to minimise the cost of reserve sites for wolf protection. Their constraint is that the probability of the wolf population exceeding the targeted size should reach a specified 'margin of safety' probability. The wolf population is estimated using a Monte Carlo simulation, where population growth is modelled using difference equations that incorporate random environmental variation. The solution method they use is retrospective optimisation. This involves estimating the objective value for each site selection option under one set of random environmental effects and selecting the best-performing solution. This process is repeated with a specified number of different sets of environmental effects to calculate the confidence interval for the selected solution. Additionally, they use the importance sampling method as a variance reduction technique by ensuring a significant portion of the environmental effects considered are extreme cases. One observation they make is that the use of a simulation model limits the size of the optimisation problem that can be solved. Their solution method does not provide guarantees on finding the optimal solution or that the selected solution satisfies the constraint.

Haight and Travis (2008) formulate the RSS problem with a probabilistic objective function. The optimisation objective is to maximise the probability of species persistence (long-term species survival probability) subject to a limit on the total cost of sites. This objective function is evaluated using a discrete-time fox model similar to the one used in Haight et al. (2004). The model is implemented in Arena Professional simulation software. They solved an example problem with 2,002 solutions using a search heuristic in OptQuest (a commercial solver). One of their findings is that computational intensity is a significant issue in solving the RSS problem with SO (the time to solve their example problem with 2,002 solutions required up to 24 hours). As with Haight and Travis (1997), their solution method does not provide a guarantee of finding the optimal solution.

Since the papers by Haight and Travis (1997) and Haight and Travis (2008), no more relevant works on SO to solve the RSS problem have been found (see Appendix A.2 for the literature searched).

2.2.4 Research Gaps

The limited use of SO in RSS and the lack of any solution accuracy guarantee in the two existing attempts highlight the research gaps. This thesis aims to investigate the use of SO with some solution accuracy guarantee in solving the RSS problem.

As explained in Subsection 2.2.1, RSS problems are typically formulated using either the maximal covering location formulation or the set covering formulation. The former aims to maximise a conservation outcome (e.g. the number of species protected) subject to a cost constraint (e.g. the number of sites). The latter aims to minimise the cost subject to a conservation goal. Neither approach is intrinsically superior, and both types of formulations are widely used in RSS literature. This thesis formulates the RSS problem using the set covering formation.

More specifically, it adopts the SO formulation from Haight and Travis (1997), which aims to minimise the cost of reserve sites (deterministic) subject to a probabilistic species survival constraint (stochastic). This type of formulation is also known as a chance-constrained formulation in SO (e.g., Hong, Luo, and Nelson, 2015).

Another reason for focusing on this formulation is that the constrained SO approach allows control over whether the best solution meets minimal conservation thresholds. Specifically, the chance-constrained formulation is chosen for two reasons: first, it aligns with the recommendation of ecologists like Ellingson and Lukacs (2003), who suggest that wildlife populations should be defined "in a probabilistic fashion"; second, it leverages the unique properties of probability constraints, as discussed further in Subsection 2.3.3.

2.3 Simulation Optimisation

This section reviews relevant literature on SO. Subsection 2.3.1 describes the general SO problem. Subsection 2.3.2 focuses on a specific group of SO methods, Ranking and Selection (RnS), which are particularly relevant to the RSS problem addressed in this thesis. Subsection 2.3.3 further narrows the focus to methods designed to solve RnS problems with stochastic constraints and details a solution method used in this thesis to solve the proposed RSS problem.

2.3.1 Simulation Optimisation Background

This section provides a general background on SO. Subsection i defines SO and its general formulations. Subsection ii describes the general classification of SO problems and explains the focus on RnS. Subsection iii describes the errors associated with SO methods and highlights the one most relevant to the RSS problem that this thesis addresses.

i Overview

Generally, Simulation Optimisation (SO) problems can be formulated as:

$$\min_{x \in I} \quad f(x) \tag{2.9}$$

where f represents the objective, x is the decision variable, and I is the solution set. The variable x can be a vector of variables or a single variable, and it can take continuous or discrete values. The objective and constraint functions can be linear or nonlinear, but at least one of them involves randomness and cannot be evaluated exactly (Jian and Henderson, 2015). A simulation model, such as discrete-event simulation, Monte Carlo, agent-based simulation, hybrid simulation, or system dynamics, is used to estimate the value of the objective function and/or the constraints. Minimisation is used for the objective function here, but if the objective is to maximise, then stating (2.9) as min $-f(x)$ is sufficient (Nelson and Pei, 2021, p. 231).

Usually, the SO problems have an objective that involves randomness and cannot be evaluated exactly (i.e. they are stochastic). Extensive reviews on this type of SO problem can be found in Amaran et al. (2016), Fu and Henderson (2017), chapter 9 of

Nelson and Pei (2021), and the handbook by Fu (2015).

For SO problems where both the objective and constraints are stochastic, it is more challenging to solve, as the feasibility of a solution (i.e. if it satisfies the constraints) needs to be verified before (or concurrently with) estimating the objective. One way to formulate such a problem is as follows:

$$\begin{aligned} \min_{x} \quad & f(x) \\ s.t. \quad & g_i(x) \geq 0, \qquad i = 1, ..., n \end{aligned} \tag{2.10}$$

where n is the number of stochastic constraints.

ii Classification

Depending on the nature of the solution space I, the SO problem is usually categorised as a discrete SO or continuous SO problem. Since the RSS problem this research focuses on typically has a finite number of ways to select sites, and its solution space is naturally discrete, this chapter does not review continuous SO. Interested readers are referred to Frazier (2018), Nelson and Pei (2021, pp. 259–267), and Fu (2015) for reviews on different methods in continuous SO.

Discrete SO problems can be further categorised based on the size of I. For problems with a solution space that is finite but very large, where examining all possible alternatives is practically impossible, solution methods such as Ordinal Optimisation, Random Search and Bayesian methods are often used. Ordinal Optimisation selects a subset of solutions from the solution space and spends the computational effort on this subset. It focuses on the probability that at least some top solutions are in the subset. Interested readers are referred to Chen, Jia, and Lee (2013). Random Search explores the solution space by sampling solutions based on some search strategy and evaluating their performance using simulations. See Andradóttir (2006) for more about random search algorithms.

Bayesian methods treat unknown performance values as random variables with prior probability distributions; whenever new data are observed, these distributions are updated using Bayes' rule. One such approach is the Gaussian Markov Random Field (GMRF) of Salemi et al. (2019), which models these unknowns using a Gaussian Markov Random Field (GMRF), where each solution is a node and only neighbouring nodes are directly correlated. This local Markov property makes the precision matrix sparse, allowing for computationally efficient maintenance of a posterior over all solutions in large solution spaces.

This research focuses on the RSS problem where all possible solutions can be listed and evaluated. The method commonly used to solve SO problems where all solutions can be simulated is Ranking and Selection (RnS).

iii Errors

Three fundamental types of errors could happen while solving SO problems (Nelson and Pei, 2021, p. 233):

Error 1 The first error occurs when the optimal solution is never simulated. This error arises when the solution space cannot be exhaustively evaluated, either because it is impossible to list all solutions or because the solution space is so large that it is unrealistic to simulate all solutions.

Error 2 The second error occurs when the best solution, which has already been simulated, is not selected. This can happen regardless of the size or type of the solution space, as it is influenced by the randomness in the simulation model and how small the gap is between the best and second best solution.

Error 3 The third error occurs when the estimate of the value of the selected solution is inaccurate. In the process of selecting the best solution, there is a natural bias towards solutions with better simulated values. Once a solution is selected, the estimation of its value is likely to be lower than its actual value (for minimisation problems).

Since this research focuses on the RSS problem in which all solutions can be listed and simulated, and with a deterministic objective, **Error 1** and **Error 3** are not relevant here.

Error 2 is usually addressed through 'correct selection', which guarantees the probability that when the algorithm stops, the selected solution (which has the best estimated value) is the actual best among all simulated solutions (i.e. a statistical guarantee on solution accuracy). Because of the multiplicity of this probability, the effort required to control it is directly linked to the total number of solutions. Hence, this method of controlling the error is typically used for problems with a relatively small solution space.

2.3.2 Ranking and Selection

Building on the concept of correct selection, RnS is a group of algorithms designed to address **Error 2** through the probability of correct selection (PCS) for problems where every solution can be listed and simulated. For readers who are not familiar with RnS, see Nelson (2022) for an introductory tutorial.

There are several ways to classify RnS procedures. This section follows the fixed-precision and fixed-budget categorisation proposed by Hunter and Nelson (2017) and Hong, Fan, and Luo (2021).

Fixed-budget procedures typically focus on the optimal allocation of the simulation budget. The goal is typically to maximise the PCS subject to a computational budget constraint, or minimise the total computational budget subject to an 'Approximate PCS' constraint (Lee et al., 2010). One of the first examples of such procedures is the Optimal Computing Budget Allocation ($\mathcal{OCBA}$) by Chen et al. (2000). However, this formulation is designed for scenarios where a computational budget is a primary consideration. In the case of RSS, where the existence of wildlife species could be directly impacted, this thesis argues that the priority should be on guaranteeing species survival, with computational efficiency as a secondary concern. Hence, this chapter does not explore the fixed-budget formulation in detail. Interested readers can find more details in the book by Chen and Lee (2011) and review papers by Chen et al. (2015) and Lee et al. (2010).

Fixed-precision procedures aim to achieve an overall PCS by controlling the probability of making Type I and Type II errors in multiple hypothesis tests on different solutions. However, the sample size required to achieve such a PCS guarantee can be computationally infeasible when there is more than one optimal solution or when there are several solutions very close to the optimum. One typical approach to address this issue is to assume that the best solution is at least δ better than the second best, where δ is usually chosen as the smallest difference in solutions' performance that would begin to have a practical impact. This approach is known as the indifference zone (IZ). For other ways of relaxing the PCS condition, details can be found in the review papers by Hunter and Nelson (2017) and Eckman, Plumlee, and Nelson (2020).

Within the IZ formulation, depending on how samples are collected, procedures can be further divided into two-stage procedures and fully sequential procedures.

Two-stage Procedures The general mechanism of two-stage procedures is as follows: in the first stage, an initial set of samples from each solution is collected. Based on the mean and variance of those samples, the number of additional samples needed for the second stage to achieve the overall PCS is calculated. In the second stage, these additional samples are collected for each solution. The performance of each solution is then evaluated, and the solution with the best performance is selected.

One example of such a procedure, and one of the earliest, is Rinott's procedure ($\mathcal{R}$, Rinott, 1978). $\mathcal{R}$ assumes all simulation outputs are independent of each other and follow a normal distribution. The rationale behind this assumption is that as different streams of random numbers are used for different solutions, one can safely assume the simulation outputs are independent. Additionally, because multiple replications are needed for each solution, the mean outputs of those solutions are approximately normally distributed based on the central limit theorem. For more examples of two-stage procedures, see Swisher, Jacobson, and Yücesan (2003) and Kim and Nelson (2006).

Fully Sequential Procedures The idea of fully sequential procedures is similar to that of two-stage procedures. Both approaches take an initial set of samples in the first stage. However, rather than collecting all samples for all solutions at once in the second stage, fully sequential procedures collect samples sequentially, update the information based on the samples collected, and repeat this process until either enough evidence is gathered to select the best solution or the stopping criteria are met.

As fully sequential procedures constantly try to rule out solutions that are not optimal, a clearly poor solution might be eliminated just by collecting the first-stage samples. This formulation may require no more simulation runs than two-stage procedures and may require fewer simulation runs, as it needs fewer samples for solutions identified as unlikely to be the best and does not collect extra samples for solutions identified as the best. However, this does not necessarily mean it is more computationally efficient. For example, when the procedure operates outside the simulation software, switching between the procedure and the software can be computationally expensive (Currie and Monks, 2021).

One of the most famous procedures of this category is the $\mathcal{KN}$ procedure (Kim and Nelson, 2001). $\mathcal{KN}$ focuses on the situation where the $f(x)$ in Eq.(2.9) is defined by the expected performance, i.e. $f(x) = E[Y(x)]$. The procedure records the means and covariances of solutions based on the initial samples in the first stage. Then, the screening stage narrows down these solutions based on the mean and threshold. The screening stage iterates by collecting one extra sample for all solutions still in contention and stops when only one solution is left or the maximum number of samples has been reached. Interested readers can find more about fully sequential procedures in Hong, Fan, and Luo (2021) and Kim and Nelson (2007).

2.3.3 RnS with Stochastic Constraints

The methods discussed in Subsection 2.3.2 assume that the feasibility of solutions is known. However, when dealing with stochastic-constrained SO problems such as the RSS problem, the feasibility of a solution must also be evaluated accordingly. Furthermore, the general SO formulation focuses on optimising the objective function but does not provide control over whether the value of the best solution meets a minimal threshold.

One way to formulate such a problem is the Constrained Selection of the Best (CSB) formulation, where one optimises the objective (primary performance measure) while satisfying all constraints (secondary performance measures). Since the RSS problem this thesis focuses on has one stochastic constraint, the rest of this section focuses on CSB formulations with a single constraint.

Subsection i reviews two CSB formulations: one with an expectation constraint and one with a chance constraint. Since the RSS problem this thesis focuses on is chance-constrained, Subsection ii describes the idea behind a solution method of the chance-constrained formulation. Subsection iii further details the characteristics of the solution method. Subsection iv then provides other solution methods for the CSB problem.

i CSB Formulations

When both the objective and constraint are defined by expected performance, one solution method is the Expectation Constrained Selection of the Best (ECSB) by Andradóttir and Kim (2010). The formulation is written as:

$$\begin{aligned} \min_{i=1,2,\ldots,k} \quad & E(X_i) \\ s.t. \quad & E(Y_i) \geq 0 \end{aligned} \tag{2.11}$$

where X_i denotes the primary performance measure and Y_i denotes the second performance measure.

When the secondary performance measure is probabilistically constrained, Hong, Luo, and Nelson (2015) formulates the Chance Constrained Selection of the Best (CCSB) as:

$$\max_{i=1,2,\ldots,I} \quad E(X_i) \tag{2.12}$$

$$s.t. \quad Pr\{Y_i \geq N\} \geq 1-\gamma \tag{2.13}$$

Following the authors' explanation, Y_i can be understood as a quality of service constraint, where the constraint $Y_i \geq N$ means the outcome is satisfactory, and $Y_i < N$ means the outcome is unsatisfactory. The chance constraint in Eq.(2.13) means the probability of the outcome of the secondary performance measure being satisfactory is at least $1 - \gamma$, where γ is the violation probability and $0 < \gamma < 0.5$. Since the event $\{Y_i \geq N\}$ is either true or false, it can be formulated as a Bernoulli random variable.

Both CCSB and ECSB determine a solution's feasibility using the IZ concept (see Subsection 2.3.2). However, they approach the problem differently. ECSB assumes that the outputs of the primary and secondary performance measures are jointly normally distributed with an unknown mean and covariance. Meanwhile, CCSB takes advantage of the probabilistic constraint and the Bernoulli random variable in its sample size calculation, thereby avoiding the need to assume the distribution of the secondary measure's output.

ii Feasibility Determination

As explained in Subsection 2.2.4, the RSS problem this thesis focuses on has a deterministic objective function. Therefore, the samples required are purely for determining the feasibility of solution i, i.e. whether i satisfies Eq.(2.13). The feasibility of i can be determined by performing a hypothesis test to check if $Pr\{Y_i \geq N\}$ is greater than $1 - \gamma$. Hong, Luo, and Nelson (2015) write the hypothesis test as follows:

$$\begin{aligned} &H_0 : p_i > \gamma \quad \text{(i.e. solution } i \text{ is infeasible)} \\ &H_1 : p_i \leq \gamma \quad \text{(i.e. solution } i \text{ is feasible)} \end{aligned} \tag{2.14}$$

where p_i denotes the probability of failing to meet the constraint ($p_i = Pr\{Y_i < N\}$). As the authors explain, this formulation adopts a conservative view towards solution feasibility, meaning that declaring an infeasible solution as feasible (i.e. rejecting H_0

when $p_i > \gamma$, Type I error), is more harmful than declaring a feasible solution infeasible (i.e. not rejecting H_0 when $p_i \leq \gamma$, Type II error).

Similar to the IZ approach described in Subsection 2.3.2, to avoid the issue of being unable to determine the feasibility of a solution when the constraint is tight, CCSB controls the Type I and II errors by introducing a feasibility tolerance level δ_γ. The Type I and II error controls in Hong, Luo, and Nelson (2015) are defined as:

$$\text{Type I error control: } \Pr\{\text{reject } H_0 \mid p_i > \gamma\} \leq \beta_1 \tag{2.15}$$

$$\text{Type II error control: } \Pr\{\text{do not reject } H_0 \mid p_i \leq \gamma - \delta_\gamma\} \leq \beta_2 \tag{2.16}$$

where β_1 and β_2 are the target probability of making Type I and II errors, respectively.

Based on these definitions of error controls, there are three possible situations based on the value of p_i as explained by the authors:

Situation 1: If a solution is infeasible (i.e. $p_i > \gamma$), then the probability of it being declared as a feasible solution is less than β_1.

Situation 2: If a solution is clearly feasible (i.e. $p_i \leq \gamma - \delta_\gamma$), then the probability of declaring the solution as infeasible is less than β_2.

Situation 3: If a solution is close to the feasibility boundary (i.e. $\gamma - \delta_\gamma < p_i \leq \gamma$), then there is no explicit control of the probability of Type II error.

The calculation of the sample size n required for achieving the defined selection errors control is detailed in Hong, Luo, and Nelson (2015). This section highlights the main ideas they used.

Suppose n simulation runs are performed on solution i, and outputs $\{Y_{i1}, Y_{i2}, ...Y_{in}\}$

have been collected. Let $Z_n = \sum_{j=1}^{n} 1_{\{Y_{ij}<N\}}$ be the total count of the runs that do not satisfy the requirement (referred to as 'failure count' for simplicity).

To achieve the Type I error control in Eq.(2.15), there exists an integer $m_{\beta_1}(n)$ such that the probability of Type I error is less than β_1 when $Z_n \leq m_{\beta_1}(n)$, written as:

$$m_{\beta_1}(n) = \max\{m \in \{0, 1, ..., n\} : Pr\{Z_n \leq m \,|\, p_i = \gamma\} \leq \beta_1\} \tag{2.17}$$

This means that to achieve a probability of making a Type I error of at most β_1, out of n samples collected, the maximum number of failure counts a solution can have is $m_{\beta_1}(n)$ before being declared as infeasible.

Also, for Type II error control in Eq.(2.16), there will be a sample size $n(\beta_2)$, such that the probability of Type II error is less than β_2 when $Z_n \geq m_{\beta_1}(n) + 1$, written as:

$$n(\beta_2) = \min\{n \in \{0, 1, ...\} : Pr\{Z_n \geq m_{\beta_1}(n) + 1 \,|\, p_i = \gamma - \delta_\gamma\} \leq \beta_2\} \tag{2.18}$$

This means that to achieve a probability of making a Type II error of at most β_2 given β_1, if the failure counts out of $n(\beta_2)$ samples collected has not reached $m_{\beta_1}(n)$, then the solution is declared feasible.

Since $\{Y_{ij} < N\}$ is a Bernoulli random variable, Z_n follows a binomial distribution. Then, the value of $m_{\beta_1}(n)$ and $n(\beta_2)$ can be calculated by solving both Eq.(2.17) and Eq.(2.18). For simplicity, Hong, Luo, and Nelson (2015) assume $\beta_1 = \beta_2 = \beta$ and that the value of β is the Bonferroni corrected value of the overall significance level of the test (denoted as α). They use the normal approximation to the binomial to find the values of $m_\beta(n)$ and $n(\beta)$. The equations they provide to calculate $n(\beta)$ and $m_\beta(n)$

are:

$$\tilde{n}(\beta) = \frac{Z_{1-\beta}^2}{\delta_\gamma^2}\left(\sqrt{(\gamma-\delta_\gamma)(1-\gamma+\delta_\gamma)} + \sqrt{\gamma(1-\gamma)}\right)^2$$

$$n(\beta) = \lceil \tilde{n}(\beta) \rceil \tag{2.19}$$

$$\tilde{m_\beta}(n) = n\gamma - Z_{1-\beta}\sqrt{n\gamma(1-\gamma)}$$

$$m_\beta(n) = \lfloor \tilde{m_\beta}(n) \rfloor \tag{2.20}$$

where $\tilde{n}(\beta)$ and $\tilde{m_\beta}(n)$ are the continuous approximation of $n(\beta)$ and $m_\beta(n)$ respectively.

The feasibility check process becomes straightforward after the values of $m_\beta(n)$ and $n(\beta)$ are calculated. One approach is to collect $n(\beta)$ samples and count the number of failures for each solution. Solutions with a failure count less than $m_\beta(n)$ can be declared feasible, with a probability of making a selection error less than β when the solution's violation probability is outside the $(\gamma-\delta_\gamma, \leq \gamma]$ range. This is the "Fixed-Sample-Size Feasibility Test" in the Hong, Luo, and Nelson (2015) paper.

Another procedure in their paper – the "Sequential Feasibility Test" – follows the idea of the fully sequential procedures described in Subsection 2.3.2. Instead of collecting all $n(\beta)$ samples for each solution, it uses the sequential nature of the simulation sample generation process, collecting one sample at a time and checking if the output is a failure or success. It declares a solution infeasible once the failure count reaches $m_\beta(n)+1$ and stops collecting further samples for this solution. If, after collecting all $n(\beta)$ samples for a solution, the failure count does not reach $m_\beta(n)+1$, the solution is declared feasible.

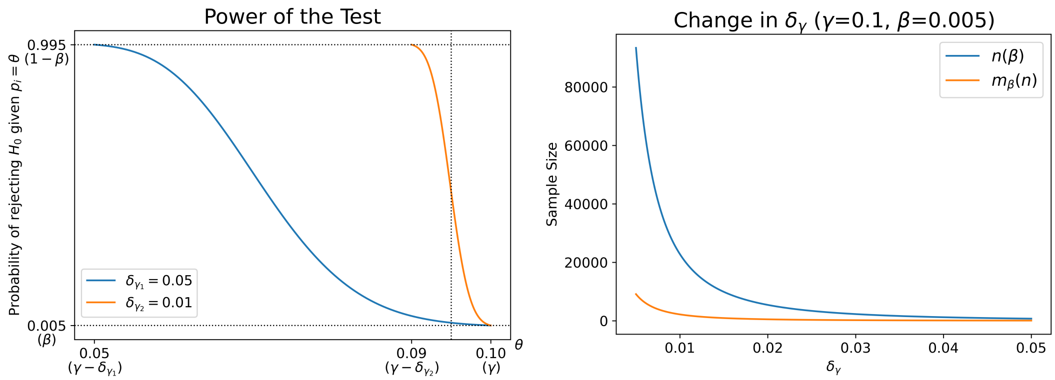
iii Feasibility Tolerance Level

As pointed out by Hong, Luo, and Nelson (2015), when a solution's violation probability is in the $(\gamma-\delta_\gamma, \gamma]$ range, there is no explicit control over the Type II error. As a result, a feasible solution may be declared as infeasible with a probability larger than β, meaning the test may reject more feasible solutions than desired. The approach they suggest to address this problem is to reduce the tolerance level δ_γ, which requires an increase in the sample size n. This compromise plays an important role in the analysis of the solution method performance for the RSS problems discussed in Chapters 4, 5 and 6. Hence, this section uses an example to show the scale of the sample size change with changing δ_γ. The example has 10 solutions and aims to achieve an overall significance level α of 5%. The error control rate β for each hypothesis test is Bonferroni corrected to $\frac{\alpha}{10} = 0.005$.

Figure 2.1a shows how reducing δ_γ increases the probability of declaring a solution feasible given a $p_i \in (\gamma - \delta_\gamma, \gamma]$ and a β of 0.005. This probability is represented by the power curve, which charts the probability of rejecting H_0 when $p_i = \theta$. Since this section only considers solutions that are feasible and close to the feasibility boundary, rejecting H_0 means correctly selecting the solution as feasible, which is equivalent to the PCS.

For the example problem with a γ of 0.10, if the violation probability of a solution is 0.095 (vertical dotted line in Figure 2.1a), and the δ_γ value is set to 0.05 (the blue curve), the test has a very low PCS (0.0146). However, if $\delta_\gamma = 0.01$ (the orange curve), the PCS increases to 0.4758. The increase in the PCS is a result of the narrowing of the range of $(\gamma - \delta_\gamma, \gamma]$, as shown in Figure 2.1a (from the blue curve to the orange curve).

Figure 2.1: Effect of changing the Sequential Feasibility Test parameter values on the sample size.



(a) (b)

Figure 2.1b presents the scale of sample size increases required for reducing δ_γ in the same example problem. The growth rate of ${\delta_\gamma}^{-2}$ for the maximum sample size required $n(\beta)$ is anticipated due to Eq.(2.19). With such a growth, the sample size goes to infinity as δ_γ approaches 0. In the example mentioned above, reducing δ_γ from 0.05 to 0.01 caused $n(\beta)$ to increase from 712 to 22,799 per solution.

As the Sequential Feasibility Test stops collecting more samples for a solution when its failure count reaches the threshold $m_\beta(n)$, the overall increase in total sample size depends on the characteristics of the solution space. Four types of solution spaces need to be considered: those with solutions that are infeasible and far from the feasibility boundary, those with solutions that are infeasible but not far from the feasibility boundary, those with solutions that are feasible and close to the feasibility boundary, and those with solutions that are all clearly feasible.

Assuming a solution i is infeasible and far from the feasibility boundary, i.e. every simulation output is $Y_i < N$ (referred to as clearly infeasible here), then the procedure will collect $m_\beta(n)$ runs' results for i and declare it as infeasible. Hence, if a problem has a solution space that contains only this type of solution, the total sample size required will be $m_\beta(n)$ multiplied by the number of solutions.

On the contrary, assuming a solution i is clearly feasible, i.e. every simulation output is $Y_i \geq N$, the procedure will collect $n(\beta)$ samples on this solution and declare it as feasible. Hence, if a problem has a solution space that contains only this type of solution, the total sample size required will be $n(\beta)$ multiplied by the number of solutions.

For a problem with solutions that are infeasible but not far from the feasibility boundary or feasible and close to the feasibility boundary, the total sample size required will be close to $n(\beta)$ multiplied by the number of solutions, as the procedure requires more samples to reach or not reach the failure threshold $m_\beta(n)$ for each solution. The actual sample size required in this case cannot be analytically calculated.

iv Other Stochastic Constrained RnS Procedures

Currie and Monks (2021) provide a practical and user-friendly two-stage selection procedure for chance-constrained SO problems. It differs from CCSB in that it is based on subset selection, which aims to identify the best m designs out of a total of k possibilities, rather than using the indifference zone approach. The key idea is to use bootstrapping to evaluate the performance of each design and assess the likelihood of violating the chance constraints. They also provide a ready-to-use Python package in their paper.

Lee, Park, and Park (2018) propose a fully sequential RnS framework for self adjusting δ_γ for the ECSB formulation. Their idea is to run two RnS procedures simultaneously and adjust the δ_γ at the end of each iteration until both procedures give the same set of feasible solutions.

Another approach for the ECSB formulation comes from Zhou et al. (2022), who propose a fully sequential RnS algorithm for constraints with multiple threshold values. This procedure is suitable for RSS problems with constraints that have multiple thresholds, for example, one threshold for the total population of the wildlife species and one threshold for the number of alpha pairs.

If the problem needs to be solved in a limited time, readers are directed to Han, Kim, and Park (2021) and Szechtman and Yücesan (2016), where they adapted an OCBA framework with Bayesian statistics to solve the stochastic constrained RnS problem.

Another way to deal with multiple performance measures is to formulate the problem as a multi-objective RnS problem. See Yoon and Bekker (2020) for a review of such methods.

If the solution space is too large (i.e., not all solutions can be evaluated), solution methods for such a problem include Probabilistic Branch and Bound (e.g., Tsai et al., 2018), reformulating a stochastic constraint to a non-linear function (see Lam and Qian (2019) for a review), and heuristics (e.g., Horng and Lin, 2023).

2.4 Summary and Research Question

This chapter reviews the use of simulation and optimisation methods in the RSS problem and relevant SO methods. The literature on the reserve selection problem primarily focuses on optimisation, which can evaluate a large number of options but is limited in its ability to capture the inherent stochasticity and complexity of ecosystems. Simulation allows for the study of complex ecosystems, but its ability to evaluate a large number of options is limited. Combining the two methods provides a more balanced approach to the problem.

However, as highlighted in Subsection 2.2.4, the application of SO in reserve selection is very limited. Existing studies that formulate the RSS problem as an SO problem

(Haight and Travis (1997) and Haight and Travis (2008)) do not provide any accuracy guarantee on the selected solutions. This leads to the first research question of this thesis:

RQ1: How well do current SO methods perform (in terms of solution accuracy and computational effort) when applied to RSS problems? More specifically, this thesis investigates the performance of several adaptations of the Sequential Feasibility Test procedure from Hong, Luo, and Nelson (2015) on solving variations of a chance-constrained RSS problem (see Subsection 2.2.4). The problem has a deterministic objective, which is to minimise the cost of reserve sites, subject to a probabilistic species survival constraint (the species survival is estimated using the proposed simulation model). The solution procedure is specially designed for solving chance-constrained SO problems, and it ensures solution accuracy by providing a guarantee on the PCS (see Subsection 2.3.3).

As explained in Section 2.3, computational intensity can present a significant challenge, particularly when solving chance-constrained SO problems. This thesis aims to reduce computational effort from two perspectives: leveraging insights from the simulation model itself and incorporating expert knowledge. This leads to the following two research questions:

RQ2: How and to what extent can computational effort in SO be reduced by leveraging the simulation model? This thesis aims to propose an approach that utilises model information to reduce computational effort while ensuring that the selected solution meets a specified PCS. It investigates the performance of such approaches when applied to a variation of the chance-constrained RSS problem.

RQ3: How and to what extent can expert opinion be used to reduce the computational effort in SO? This thesis aims to propose a heuristic approach that uses expert knowledge to reduce computational effort. It investigates the performance of this approach when applied to a variation of the chance-constrained RSS problem.

To answer these research questions, this thesis adopts a computational study approach. The computational study involves solving different versions of the RSS problem using variations of the solution method. The example species this thesis focuses on is the grey wolf. To create different versions of the RSS problem, a model capable of simulating various grey wolf reserve scenarios is required. Such scenarios enable the evaluation of SO for problems with different challenges. The simulation model is detailed in Chapter 3.

Three scenarios of the grey wolf RSS problem, each with different characteristics, are studied in Chapters 4, 5, and 6. Scenario 1 is a basic reserve problem. In solving this scenario, Chapter 4 aims to answer RQ1 by evaluating the effectiveness of the Sequential Feasibility Test procedure in terms of both selection accuracy and computational intensity. Scenarios 2 and 3 are designed to demonstrate cases where two different types of solution space filtering approaches (proposed to answer RQ2 and RQ3, respectively) can be applied. Chapters 5 and 6 explain these computational intensity reduction approaches. To answer RQ2 and RQ3, the effectiveness of these approaches is evaluated in Chapters 5 and 6 by solving Scenarios 2 and 3, respectively. To assess their generalisability, the approach in Chapter 5 is applied to Scenario 1 towards the end of Chapter 5, and likewise for Chapter 6.

Chapter 3

Problem Formulation and Simulation Model

3.1 Introduction

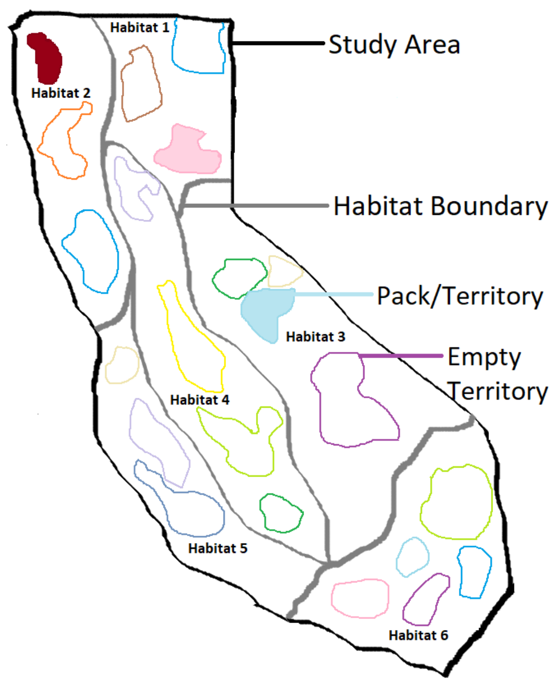
To explore the extent to which SO can be applied to the RSS problem, a model capable of simulating different scenarios of an RSS problem is required. These scenarios will enable the evaluation of SO methods under various reserve design challenges.

This chapter lays the foundation for Chapters 4, 5, and 6. It provides the essential background for addressing the research questions detailed in Chapter 2. Section 3.2 explains the grey wolf RSS problem that this research focuses on (Chapters 4, 5, and 6 explore three scenarios of this problem). Section 3.3 presents the chance-constrained formulation of the problem. Section 3.4 explains the modifications made to the solution method to better suit the RSS problem (Chapter 4 directly applies this method to solve the problem, while Chapters 5 and 6 provide adaptations of the method). Section 3.5 describes the conceptual model of the grey wolf model. Section 3.6 explains the computer model built based on this conceptual model. The computer model is used in the next three chapters with varying input parameters to create different scenarios. Section 3.7 explains these input parameters. Section 3.8 provides a summary of this chapter.

3.2 Grey Wolf RSS Problem

The grey wolf chance-constrained RSS problem studied in this research aims to minimise the cost of reserve sites (deterministic) subject to a 'species persistence' constraint (stochastic). Figure 3.1 provides an abstract representation of the study area.

Figure 3.1: Illustrative visualisation of the study area of a hypothetical grey wolf RSS problem.



To mimic the variety of terrestrial environments (e.g., deserts or forests) in which grey wolves realistically live, this thesis adopts Haight and Travis (2008)'s approach and assumes the study area is divided into regions called habitats (outlined by grey lines in Figure 3.1). These habitats are further divided into smaller areas called sites. As grey wolves are territorial animals, these sites represent their territories. For ease of understanding the grey wolf problem, the term 'territory' is used instead of 'site' hereafter.

Territories can be protected territories or not protected territories. This thesis assumes wolves can only live in protected territories. Protected territories are represented by zones marked with different colours in Figure 3.1. The protected territories can be either occupied by wolf packs (such as the blue-shaded territory in Habitat 3) or empty (such as the purple-outlined territory in Habitat 3). Although wolf territory sizes can vary, this thesis adopts the idea from Mech and Boitani (2003) that assumes territories' sizes are the same and fixed. Both occupied and empty territories can be either strictly protected (referred to as core territories) or less strictly protected (referred to as peripheral territories).

The grey wolf RSS problem this thesis focuses on is subject to a constraint on the species persistence. The species persistence in this thesis is measured through viable packs. A viable pack is defined as a pack that includes a pair of alpha wolves (i.e. able to reproduce). Species persistence is considered achieved if the number of viable packs at the end of the planning horizon (i.e. the timeframe over which the RSS is planned and assessed) is at least equal to the initial count. Based on examples from two government grey wolf conservation plans (U.S. Fish and Wildlife Service (1987) and California Department of Fish and Wildlife (2016)), the planning horizon for this thesis is set at five years.

The aim of the grey wolf RSS problem is to select a set of additional territories from available territories in each habitat to minimise the cost of territories subject to a constraint on the 'species persistence'. The problem assumes there is a limit on the number of additional territories that can be selected (hereafter budget), existing territories cannot be removed, and the unit cost of additional territories is one. Hence, the problem becomes finding the optimal allocation of additional territories between habitats. Incorporating the monetary cost of territories into the problem is straightforward and does not alter the problem structure.

3.3 Problem Formulation

The chance-constrained grey wolf RSS described in Section 3.2 can be written as follows:

$$\min_{i \in I} \sum_{h=1}^{H} X_{hi} \tag{3.1}$$

$$s.t. \ \Pr\{Y_i \geq N\} \geq 1 - \gamma \tag{3.2}$$

In the objective function Eq.(3.1), i is the solution index, I is the solution space, X_{hi} is the number of additional territories selected in Habitat h under solution i. As this thesis assumes that existing territories cannot be removed, X_{hi} should be greater than or equal to 0. Since the unit territory cost is assumed to be the same for all habitats, the objective function becomes minimising the total number of additional territories across all H habitats.

The constraint Eq.(3.2) ensures the probability of at least sustaining the number of viable wolf packs over a five-year planning horizon is at least $1 - \gamma$. The term γ is referred to as the violation probability. The variable Y_i denotes the simulated number of viable packs after five years under solution i, and N represents the number of viable wolf packs at the start of the planning horizon.

The solution space I includes solutions that satisfy the constraint (i.e. feasible solutions) and those that do not (i.e. infeasible solutions). The number of solutions, $|I|$, is naturally limited by the budget for territories (denoted as b) and can be calculated analytically.

Mathematically, I is a set containing all possible distributions of additional b territories across H habitats. Finding $|I|$ is equivalent to the combinatorial problem of distributing $b+1$ balls (representing territories) into H boxes (representing habitats).

This thesis assumes all additional territories are indistinguishable. In scenarios where all habitats are distinguishable, enumerating I becomes finding all possible distributions of up to b indistinguishable territories across H distinguishable habitats. The formula for calculating the number of possible combinations, $C(\cdot\,,\cdot)$, for a given number of territories j is (Rosen, 2007, p.377):

$$C(H+j-1, H-1) = \frac{(H+j-1)!}{j!(H-1)!} \tag{3.3}$$

As the objective is to minimise the number of additional territories selected, all j such that $0 \leq j \leq b$ should be considered. Thus, the total number of combinations is:

$$|I| = \sum_{j=0}^{b} C(H+j-1, H-1) = \sum_{j=0}^{b} \frac{(H+j-1)!}{j!(H-1)!} \tag{3.4}$$

Eq.(3.4) naturally structures the solution space from the smallest cost (i.e. $j = 0$ additional territories) to the largest cost (i.e. $j = b$ additional territories).

3.4 Solution Method

An algorithm specifically designed for the chance-constrained SO problem is Hong, Luo, and Nelson (2015)'s CCSB, as explained in Subsection 2.3.3. However, CCSB is intended for problems where both the objective and the constraint require simulation. In the chance-constrained grey wolf RSS problem that this research focuses on, the objective function (Eq.(3.1)) is deterministic. Therefore, solving the problem essentially involves determining the feasibility of each solution. As detailed in Subsection 2.3.3, CCSB offers two procedures for this purpose: the 'Fixed-Sample-Size Feasibility Test' and the 'Sequential Feasibility Test'. Given the authors' suggestion that the sequential test never takes more samples than the fixed sample test and may take fewer samples, this thesis exclusively focuses on the application of the sequential test.

Since the objective function, Eq.(3.1), is deterministic and aims to minimise the number of additional territories (i.e. cost), with the solution space structured from the smallest to the largest cost, once a solution that satisfies the constraint is found, it automatically becomes one of the optimal solutions. Given that multiple combinations are typically under the same cost, there might be more than one optimal solution.

The feasibility test can be terminated once a feasible solution is found and all other solutions with the same cost have been simulated. Any remaining solutions would require a greater cost, meaning that even though they might be feasible, they are not optimal. Hence, the feasibility test can be terminated, and all feasible solutions corresponding to the minimum cost are returned.

Intuitively, once a feasible solution is found, the number of solutions that remain to be checked is reduced, which means the sample size required by the feasibility test can be further reduced. However, doing so might jeopardise the error control and introduce randomness and dependency on the sample size. Therefore, this research takes a conservative view and does not change the sample size.

The Sequential Feasibility Test with this added terminating condition is abbreviated as CCSB-F in this thesis. Algorithm 1 summarises CCSB-F.

Algorithm 1: Adapted Sequential Feasibility Test (CCSB-F).

Initialisation: Input I, B, α, β, δ_γ, $n(\beta)$ and $m_\beta(n)$. Set $\beta = \alpha/|I|$. Define I_j as the set of all solutions that selects exactly j additional territories. Initialise Θ as an empty set to store optimal solutions.

for $j = 0$ **to** B **do**
- **for** *each solution* i *in* I_j **do**
  - Set sample size counter $\tau = 0$ and failure count $z = 0$
  - **while** $\tau \leq n(\beta)$ **do**
    - $\tau \leftarrow \tau + 1$
    - Run one simulation on solution i, get output $Y_{i\tau}$
    - **if** $Y_{i\tau} < N$ **then**
      - $z \leftarrow z + 1$
      - **if** *failure count* $z \geq m_\beta(n)$ **then**
        - Declare solution i as infeasible
        - Break
    - **if** $\tau = n(\beta)$ **then** Add i to Θ
- **if** $\Theta \neq \emptyset$ **then**
  - Terminate and output all solutions in Θ as optimal solutions

In the initialisation, I is the set that contains all solutions, B is the budget, α is the total error allowance ($0 < \alpha < 1 - 1/|I|$), δ_γ is feasibility tolerance level, and $n(\beta)$ and $m_\beta(n)$ are calculated using Eq.(2.19) and Eq.(2.20)) respectively. The solution set I should be structured with solutions ordered from the smallest to the largest cost.

For the grey wolf RSS problem, α represents the overall significance level of the CCSB-F test, γ is the maximum tolerable probability of not achieving the targeted wolf population persistence, and δ_γ is the tolerance level for ruling out clearly feasible solutions, which can be chosen based on the maximum differences the decision maker believes would begin to make a practical impact.

Algorithm 1 guarantees that for the RSS problem defined in Eq.(3.1) and Eq.(3.2), when the algorithm stops, the selected solutions are the optimal solutions and satisfy the constraint with a significance level of α given a tolerance level of δ_γ (i.e. the statistical guarantee).

This thesis acknowledges that decision makers may have additional factors to consider when deciding on territory allocation plans. Therefore, all optimal solutions are presented without bias, allowing decision makers to select the most suitable option.

Since the problem is stochastically constrained with a budget, there is a possibility that no solution will satisfy the constraint after using the entire budget. In such circumstances, the decision maker would be advised to either increase the budget or explore additional measures to aid species conservation, such as relocating wolves to other locations.

3.5 Conceptual Modelling

The fundamental modelling belief this research adopts is that "all models are wrong but some are useful" (Box, 1979). Consequently, the grey wolf model developed in this research is not intended to replicate every detail of the grey wolf's life history. Instead, the model abstracts the important aspects relevant to the RSS problem. This process of abstraction is known as conceptual modelling (Robinson et al., 2010). This section explains the conceptual model of the grey wolf developed in this research.

Subsection 3.5.1 describes the life history and behaviours of grey wolves to provide a background for the wolf model. Subsection 3.5.2 outlines the requirements of the model. Based on these requirements, certain simplifications and assumptions are made during the conceptual modelling process. Subsections 3.5.3 to 3.5.6 detail the conceptual model from the perspective of an individual wolf's life history, categorised by the wolf's age group. Subsection 3.5.7 specifies the modelling of breeding as it is conceptualised as a pack-level activity rather than an individual wolf activity. Subsection 3.5.8 provides an abstracted view of the possible states and activities of a wolf.

3.5.1 Grey Wolf

Wolves live in packs, which inhabit territories. Pack sizes generally vary from 3 to 11 members (Fuller, Mech, and Cochrane, 2003). A typical pack comprises a mated pair and one or more generations of their offspring (Mech and Boitani, 2003). Most packs produce only a single litter of pups per year by the dominant female, with breeding occurring annually under favourable conditions (Harrington et al., 1982).

Generally, the earliest reproduction age is two years (Peterson, Woolington, and Bailey, 1984). The average litter size is six pups, with litter sizes ranging from one to 13 pups (Mech, 1970). Typically, pups begin travelling with adults on hunts at about six months old (Jimenez et al., 2008). Most pups disperse from their natal pack between 9 and 36 months.

Wolves disperse throughout the year, but autumn and winter tend to be the peak seasons (Mech, 1970) (Mech and Boitani (2003) argue peak seasons are autumn and spring). Pairs form during dispersal. If two dispersers find each other and pair up, they will seek suitable land and establish their territory together (Packard and Mech, 1980). If they are unable to establish a territory locally, they may travel some distance before settling (Mech and Boitani, 2003).

Wolves die from two primary causes of mortality. One is human-caused mortality, which includes legal and illegal killing, intentional wolf population control, and car strikes (Fuller, Mech, and Cochrane, 2003). The other is natural mortality, such as starvation, accidents, injuries during travelling, hunting, territorial conflicts, conflicts with other species, old age, and disease (Peterson et al., 1998). Wolves between 6 and 12 months old die primarily due to malnutrition in winter, whereas human-caused mortality is the primary reason for adult mortality in winter (Mech (1997) and Fuller (1989)).

3.5.2 Simulation Model Design Requirements

The RSS problem aims to find the optimal allocation of additional territories across different habitats within a study area while maintaining a targeted probability of at least sustaining the number of viable wolf packs over a five-year planning horizon. The simulation model is used to estimate the number of viable packs, Y_i, in constraint Eq.(3.2), under solution i. Hence, the model's inputs should include the potential allocations of territories within different habitats (i.e. solutions).

The main requirement of the model is to reflect the impact of different territory allocation plans on the number of viable packs. To achieve this, the model should simulate individual wolf activities that affect the number of viable packs, including birth, growth, dispersal, and death. The model should also include interactions both within and across habitats (including pack formation and breeding), as these behaviours influence the overall number of viable packs.

The modelling of the wolf population within habitats borrows heavily from an individual-based (simulating the behaviour of individual wolves), stage-wise (simulating wolves at the same life stage collectively), stochastic (considering events as probabilistic), and discrete-event (events happen at discrete times) model developed by Haight, Mladenoff, and Wydeven (1998).

The key differences between the wolf model used in this study and their model lie in how dispersal, environmental variability, and the simulation run length are handled. For dispersal, their model considers only a single habitat, where wolves dispersing beyond their natal habitat (i.e. long dispersers) are assumed to be lost from the population. In contrast, this model accounts for multiple habitats, allowing long dispersers to move across habitats (adapting ideas from the fox model of Haight and Travis (2008)). Additionally, in their model, the survival of local dispersers (i.e. wolves dispersing within their natal habitat) is determined by a function, whereas in this wolf model, dispersal outcomes depend on the outcomes of other events. For environmental variability, one variation of their model calculates winter mortality rates as a function of the previous year's wolf population. However, for simplicity, this model does not include this environmental variability.

Neither Haight, Mladenoff, and Wydeven (1998) nor this model incorporates spatial aspects. However, incorporating such aspects into this model would be straightforward; for instance, by assigning distinct movement probabilities for long dispersers travelling to different habitats. This is discussed in more detail in Chapter 7.

The scale of the model is the study area, which is divided into different habitats to mimic the spatial aspect of wolf habitats. The model should simulate the wolf population over a defined time horizon and should be able to incorporate realistic starting conditions, including details of existing individual wolves, packs, territories, and habitats.

The model's output should be the count of total viable packs at the end of the planning horizon. Since the focus of this research is to evaluate the use of CCSB-F in solving the RSS problem, the model should have a fast run speed to facilitate experimentation with CCSB-F. Moreover, the model needs to be flexible enough to allow the simulation of various wolf problem scenarios to evaluate the applicability of CCSB-F.

3.5.3 Pup (0 to 6 Months)

At the start of spring, a pup is born to pack P_a at territory T_a in Habitat H_a. In the real world, pups may be born at any point from the beginning of spring to the end of summer. However, the model simplifies this by assuming all pups are born at the start of spring. Reflecting seasonal patterns, the model operates on discrete three-month intervals, assuming events occur either at the beginning or end of each season.

If this pup is female, she is denoted as age 0 months, P_a, T_a, H_a, female. If the pup is male, he is denoted as 0, P_a, T_a, H_a, male. The model does not account for gender differences except during breeding. Therefore, female and male wolves follow an identical path before and after breeding.

In reality, not all pups survive their first 3 months. However, for simplicity, this model assigns all pup mortality to the summer season. As a result, during spring, this pup will age and become three months old at the end of spring. Figure 3.2 visualises the timeline of a wolf pup's age transition from birth at the start of spring to three months old at the end of spring.

Figure 3.2: Pup in spring.

During summer, two events are considered: ageing and pup mortality. The pup born in spring could die with a fixed probability of pup mortality. If it survived, it is described as age 6, P_a, T_a, H_a (Figure 3.3).

Figure 3.3: Pup in summer.

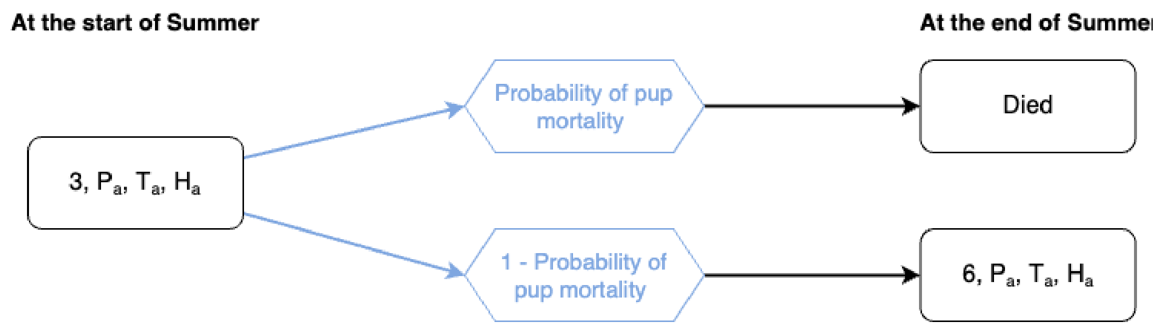


Table 3.1 summarises the possible states for pups in each season. Since all pups are born at the start of spring, by autumn, they are more than 6 months old. Wolves aged between 6 and 18 months are categorised as juveniles in this research, so there are no pups in the model during autumn or winter. Therefore, pups in autumn or winter are marked as 'N/A' in Table 3.1.

Table 3.1: States and transitions of pups.

Pups (0 to 6 months)			
Autumn	**Winter**	**Spring**	**Summer**
N/A	N/A	Age	Age or die

3.5.4 Juvenile (6 to 18 Months)

i Juvenile in Autumn

For the juvenile wolf, it remains at P_a, T_a, H_a at the start of autumn. It ages during this season to become 9 months old, represented as 9, P_a, T_a, H_a (Figure 3.4). For simplicity, the model assigns all non-pup wolf mortalities to the winter and does not differentiate mortality rates by age.

Figure 3.4: Juvenile in autumn.

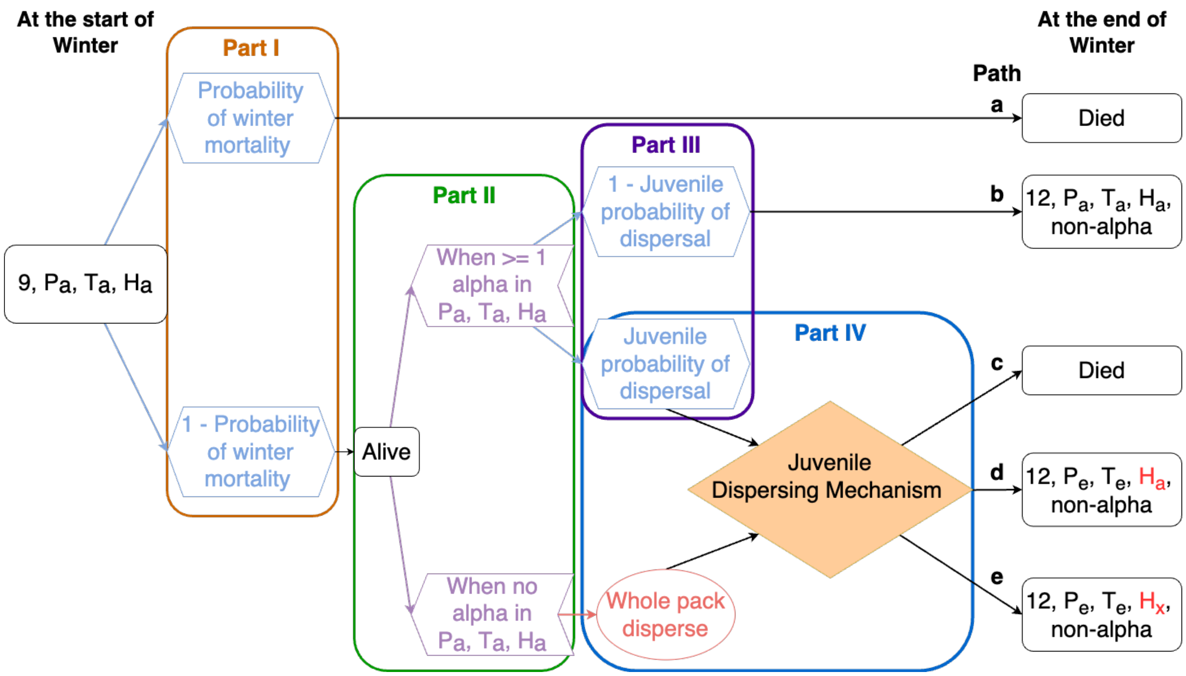
ii Juvenile in Winter

When winter arrives, this juvenile wolf could either die (Path a, Figure 3.5) or survive, with survival being determined by a probability of winter mortality (Part I, bounded by the orange line in Figure 3.5). Adapting Treves et al. (2017)'s idea, this probability is lower if T_a is a core territory compared to if it is a peripheral territory. If it survives the winter, the model then checks if its parents are still alive (Part II, bounded by the green line).

If both parents have died due to winter mortality, the model assumes the pack will disassemble due to the absence of a leader. However, if at least one parent survives, the pack remains intact, and the juvenile wolf may either disperse or stay with the pack based on a probability of juvenile dispersal (Part III, bounded by the purple line). The likelihood of a juvenile dispersing from its pack is considered to be relatively low. Following the conclusions of Kojola et al. (2006) and Blanco and Cortés (2007), all dispersal rates in the model are assumed to be independent of gender. For simplicity, the model assumes that dispersal occurs only during winter.

Should the juvenile disperse, either due to the absence of an alpha or the probability of dispersal, it goes through a dispersal process, defined as the 'Juvenile Dispersing Mechanism'. This dispersal process can result in several outcomes for the wolf: it may die (Path c), successfully occupy an empty territory within its original Habitat H_a (Path d), or settle in an empty territory in a different Habitat H_x (Path e). This dispersing mechanism is detailed in the next subsection.

Figure 3.5: Juvenile in winter.



iii Juvenile Dispersing Mechanism

The model assumes the minimum age for a wolf to reproduce at the start of spring is 21 months. Therefore, a wolf must be at least 18 months old at the start of winter to be able to reproduce in spring. Juvenile wolves are only 9 months old. Hence, dispersing juvenile wolves can only find and occupy an empty territory, and they cannot join another pack or find a mate.

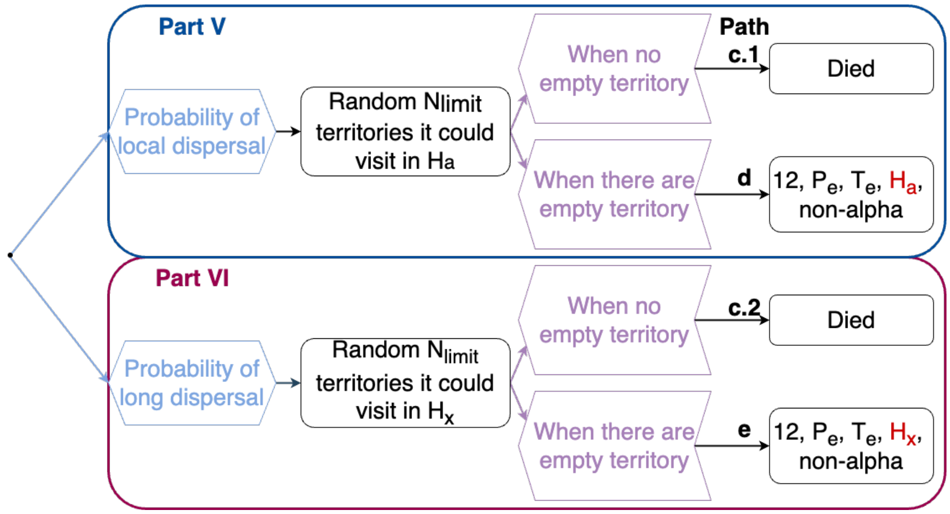
For the juvenile wolf, if it disperses, it could disperse within its original habitat (referred to as local dispersal, Part V in Figure 3.6) based on a local dispersal probability, or to another habitat (long dispersal, Part VI) based on a long dispersal probability (i.e. 1 - probability of local dispersal). For simplicity, the model assumes all local dispersers share one local dispersal probability, and all long dispersers share one long dispersal probability.

If the wolf disperses locally, there will be a few territories it could travel to, and the number of such territories is limited in the model (denoted as N_{limit}). The reason is to mimic the physical limitations on travelling distances, and to account for death caused by battling other wolves or hunger. These territories are randomly selected (excluding the wolf's original territory) under the assumption that wolves have the same level of pre-knowledge about all other territories.

Within these selected territories, if there are empty territories (denoted as T_e), the disperser is randomly allocated to one of them. This random allocation captures the stochastic nature of wolves selecting areas to live in reality. Even though this juvenile wolf cannot find a mate and form a pack at this point, it might do so in the future. Hence, the situation of a lone wolf occupying a territory is simplified as establishing a pack P_e that has only one wolf.

In essence, for a juvenile wolf dispersing within its habitat H_a, it could die due to no suitable territory (Path c.1 in Figure 3.6), or it could establish its own pack P_e at territory T_e in H_a (Path d). If the wolf disperses to another habitat (for example habitat H_x), it goes through the same allocation process in H_x as in H_a, potentially ending up in P_e, T_e, H_x (Path e) when there is an empty territory available.

Figure 3.6: Juvenile dispersing mechanism.



iv Juvenile After Winter

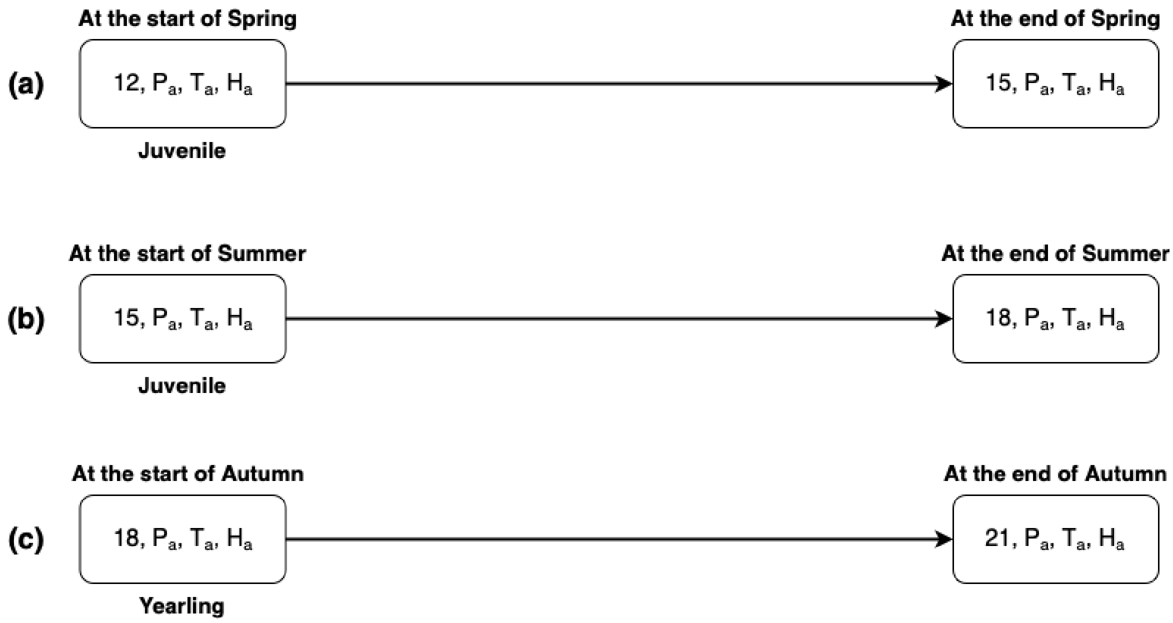
Assuming the juvenile wolf survived winter and remained with its original pack, it ages in the coming spring, summer, and autumn. It becomes 15 months old at the end of spring (Figure 3.7 (a)), 18 months old at the end of summer (Figure 3.7 (b)), at which point the wolf is classified as a yearling, and 21 months old at the end of autumn (Figure 3.7 (c)).

Table 3.2 summarises the possible states and transitions of all juveniles at each season at any point in time.

Table 3.2: States and transitions of juveniles.

Juveniles (6 to 18 Months)			
Autumn	**Winter**	**Spring**	**Summer**
Age	Age or die or juvenile disperse	Age	Age

Figure 3.7: Juvenile in spring (a), summer (b) and autumn (c).



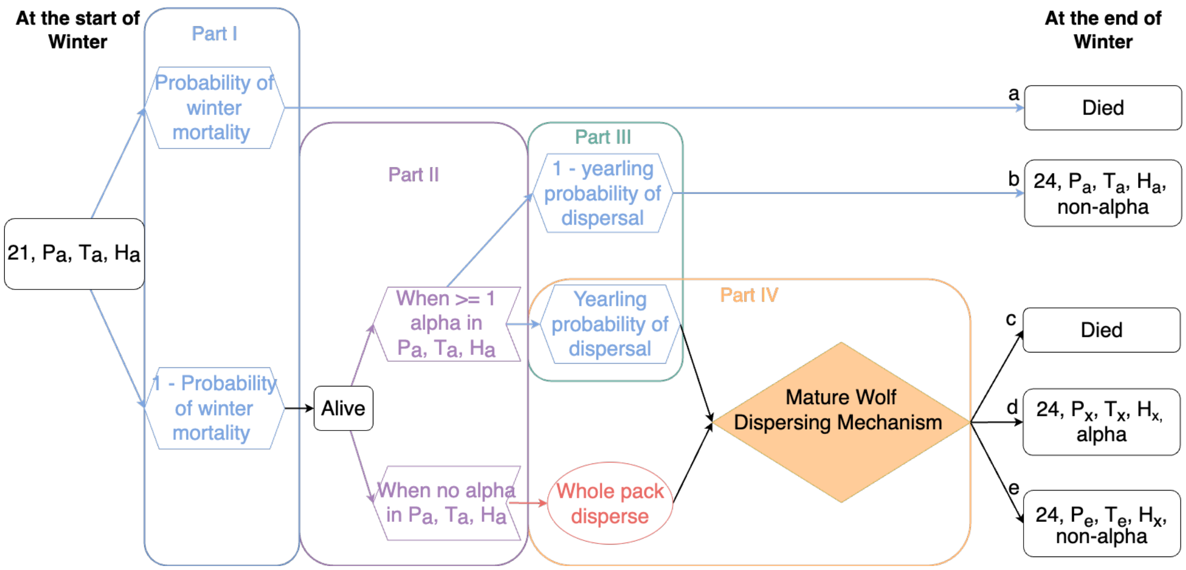
3.5.5 Yearling (18 to 30 Months)

i Yearling in Winter

At the start of winter, when this wolf is 21 months old, it goes through a process similar to that of the juvenile wolf during the winter season. This includes area-based winter mortality (Part I, Figure 3.8), the alpha wolf check (Part II), the dispersal decision (Part III), and the dispersal process (Part IV) if it follows this trajectory. It could die due to winter mortality (Path a), become a 24-month old, non-alpha wolf in P_a, T_a, H_a if it remains with its current pack (Path b), die during dispersal (Path c), or successfully disperse. If it successfully disperses, it might become a 24-month old, alpha wolf in P_x, T_x, H_x (Path d), or a non-alpha wolf in an empty territory at Habitat H_x.

The dispersal process for yearlings differs from that of juveniles. The dispersing mechanism for mature wolves (i.e. wolves over 21 months old by the start of winter) is detailed in the next subsection.

Figure 3.8: Yearling in winter.



ii Mature Wolf Dispersing Mechanism

This wolf will be 24 months old at the start of the next spring, meaning it will be old enough to reproduce. If it disperses, it could take over other packs and become the alpha wolf (Path d, Figure 3.8). For alpha wolves in general, the model assumes that each pack can have only one pair of alphas, with only the alpha female able to reproduce. For computational simplification, breeding longevity is not included in the model. When a pair of alpha wolves are present in a pack, the model assumes they will not accept new wolves and does not allow a disperser to challenge such a pack. However, if only one alpha remains, given that wolves mate for life, the pack loses its reproducibility. In this scenario, if the pack contains a suitable mate for the disperser, the model allows this disperser to join the pack and become the new alpha pair with a suitable mate. If a dispersing wolf cannot find a suitable mate, it is assumed to settle in an empty territory (Path e, Figure 3.8). Another disperser may later join it to form a pack. Such an event would result from other events in the simulation, mimicking the formation of a pair during dispersal in reality (e.g. Morales-González et al. (2022)).

Habitat allocation - Figure 3.9a The dispersal mechanism for mature wolves follows a pattern similar to juvenile dispersal in deciding between local and long dispersal. A mature wolf could disperse within its original Habitat H_a based on local dispersal probability, or to another Habitat H_x based on long dispersal probability (Figure 3.9a). Once the destination habitat for the wolf is determined, the events it may go through within H_a or H_x are identical. To simplify the narrative, H_ε is used as a generic representation for both habitats H_a and H_x in the rest of Subsection ii.

Pack allocation - Figure 3.9b In Habitat H_ϵ, there are up to N_{limit} territories to which a mature disperser could travel, similar to the process for juvenile dispersers (Figure 3.9b). The model assumes that a mature disperser always prioritises finding a suitable pack if one is available. If there is more than one suitable pack, the model assumes the dispersing wolf randomly chooses one to join. If no suitable pack is found within its travelling limits and empty territories are available, it is allocated to one of those empty territories (Path e.1 or e.2, Figure 3.9b).

A suitable pack, denoted as P_ε, is defined as a pack that has less than one alpha and a wolf of the opposite gender who is old enough to mate. The gender of the wolf is considered here, as a female wolf looks for a pack with a suitable male mate, and vice versa. Under the model assumption that wolves do not inbreed (i.e., mating with close kin), the randomly selected territories exclude the original territory from which the wolf came.

If there is a suitable pack, the wolf settles in the pack (Figure 3.9b, Path d). If the suitable pack has one alpha, the model assumes the original alpha wolf goes through a De-alphalise process (Figure 3.9c). Then, the successful disperser go through an Alphalise process (Figure 3.9d). These two processes are explained in more detail in the following section.

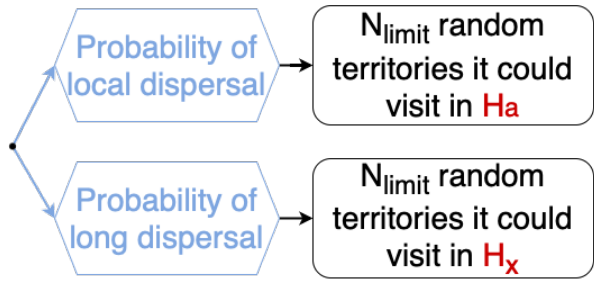
If there are packs with only one alpha among the territories the disperser can visit, but none has a suitably aged wolf of the opposite gender (Path e.1, Figure 3.9b), or if there are no packs with only one alpha (Path e.2, Figure 3.9b), and an empty territory T_e is available, the disperser moves to T_e. It then becomes a non-alpha, aged 24 months, located at P_e, T_e, H_ε by the end of winter. If none of the above scenarios applies, it dies (Figure 3.9b, Path c.1 and c.2).

Pack allocation: De-Alphalise and Alphalise - Figure 3.9c and 3.9d De-alphalise and Alphalise processes occur at the pack level when a dispersing wolf finds a suitable pack with a potential mate (Figure 3.9b, Part VI). The De-alphalise process is based on the assumption that alpha wolf pairs are loyal to each other. Therefore, if only one alpha remains in the pack, it will not form a new pair with any other wolf. The de-alphalised wolf is considered to no longer be part of the pack and is removed from the model (Figure 3.9c, Path da.1). After the De-alphalise process, no alpha remains in the pack. If the pack is without an alpha before the De-alphalise process occurs (i.e., both alphas have died or a previous disperser has occupied an empty territory), then the pack remains unchanged (Figure 3.9c, Path da.2).

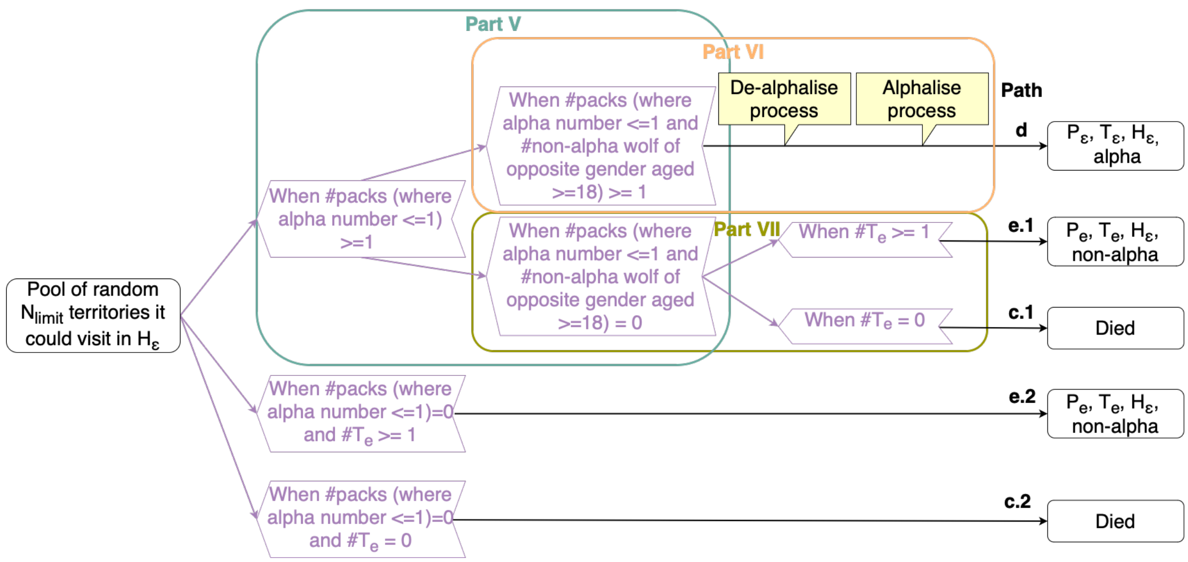
The Alphalise process occurs right after the De-alphalise process (Figure 3.9d). Through this process, the wolf who successfully joins the pack changes its status to 'alpha'. Its paired wolf also becomes an alpha after the process. If the pack contains more than one potential mate, the model randomly selects one.

Figure 3.9: Mature wolf dispersing mechanism.

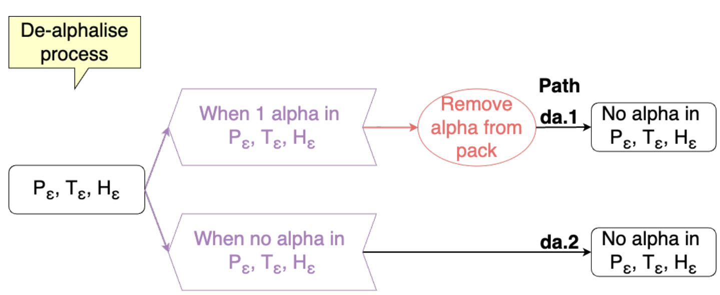
(a) Dispersers habitat allocation.



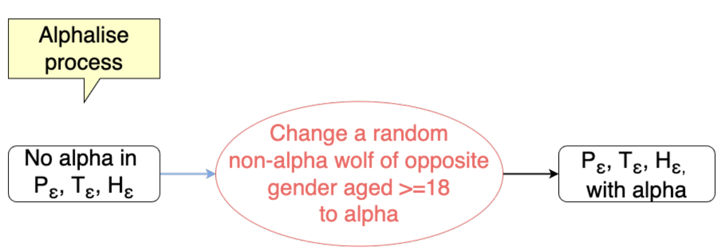
(b) Disperser pack allocation.



(c) De-alphalise process.



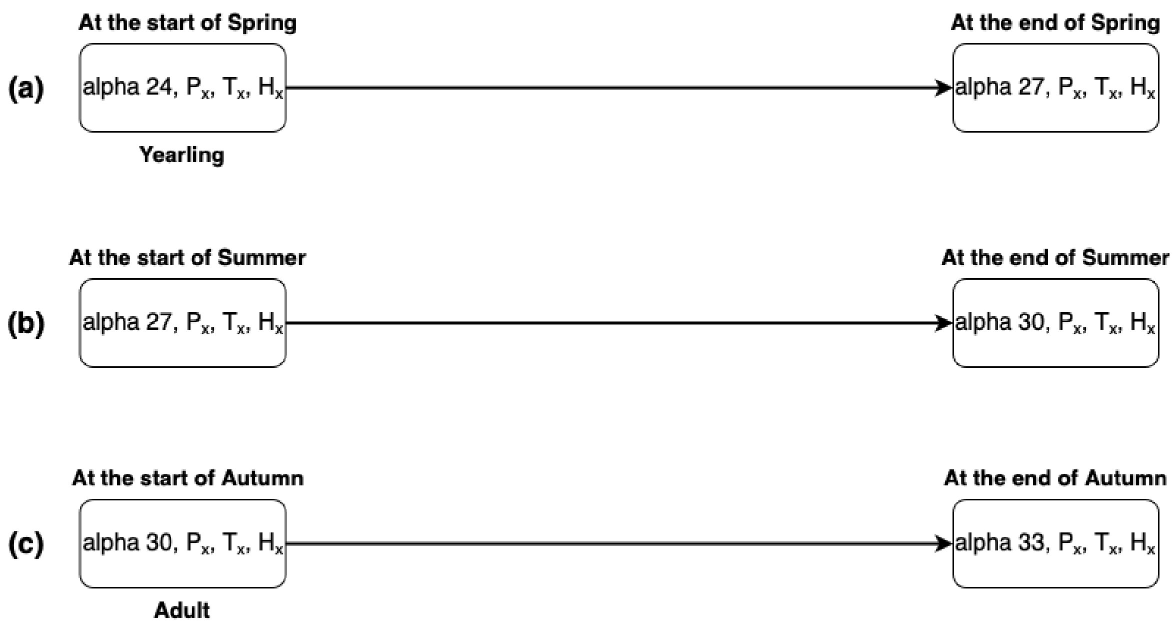
(d) Alphalise process.



iii Yearling Alpha

Suppose the yearling wolf survives the winter and finds a mate (Figure 3.8, Path d). In spring, it engages in breeding. Breeding occurs at the pack level and is detailed in Subsection 3.5.7. For this individual wolf, it ages through the upcoming spring, summer, and autumn (Figure 3.10).

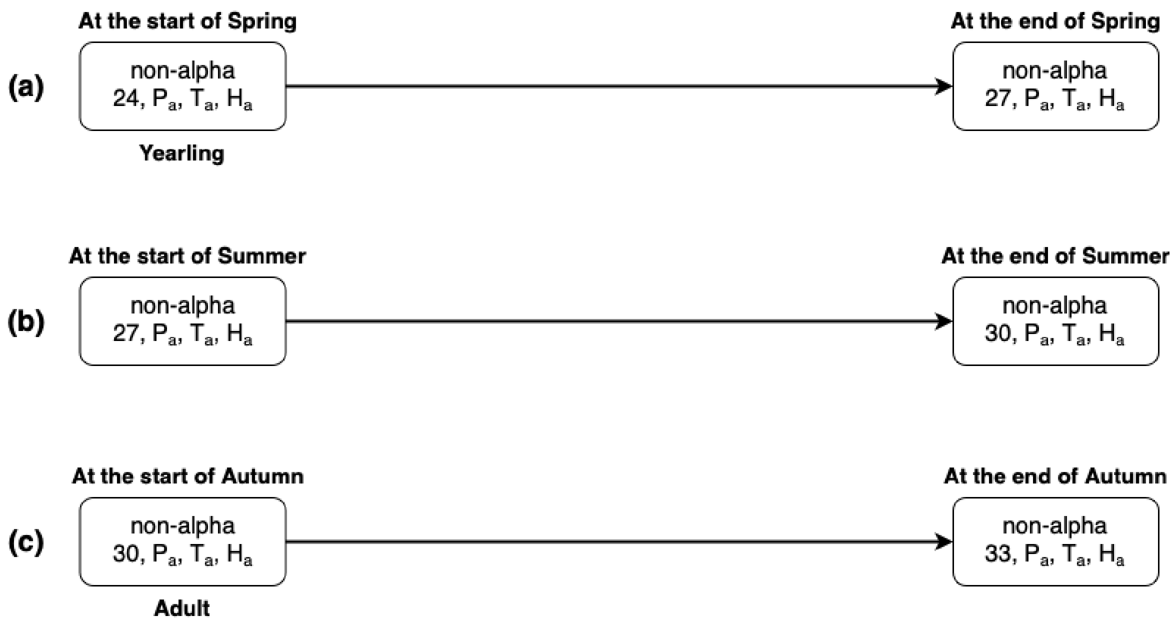
Figure 3.10: Yearling alpha in spring, summer and autumn.



iv Yearling Non-Alpha

Suppose this wolf does not disperse during the winter when it is 21 months old. If it chooses to stay with its original pack during that winter (Path b in Figure 3.8), it ages through the upcoming spring, summer, and autumn as a non-alpha (Figure 3.11).

Figure 3.11: Yearling non-alpha in spring, summer and autumn.



By the end of summer, all wolves born at the same time as this wolf are labelled as adults, as they are now 30 months old. Table 3.3 summarises the possible state transitions for all yearlings at each season and at any point in time.

Table 3.3: States and transitions of yearlings.

Yearlings (18 to 30 Months)				
	Autumn	**Winter**	**Spring**	**Summer**
Alpha	N/A	N/A	Age and breed	Age
Not alpha	Age	Age or die or mature wolf disperse	Age	Age

3.5.6 Adult (More than 30 Months)

The model, following Haight, Mladenoff, and Wydeven (1998), assumes that adult wolves have a higher probability of dispersing than yearling wolves during winter. Adult wolves go through the same process as yearling wolves, with the only distinction being the possibility of having alpha status during autumn and winter. No yearling alpha wolves exist during autumn and winter, as they were too young to successfully disperse in the previous winter. However, if yearling wolves successfully disperse and become alpha at the beginning of spring, their alpha status remains unchanged upon reaching adulthood.

Table 3.4 summarises the possible alpha status for wolves of each age group.

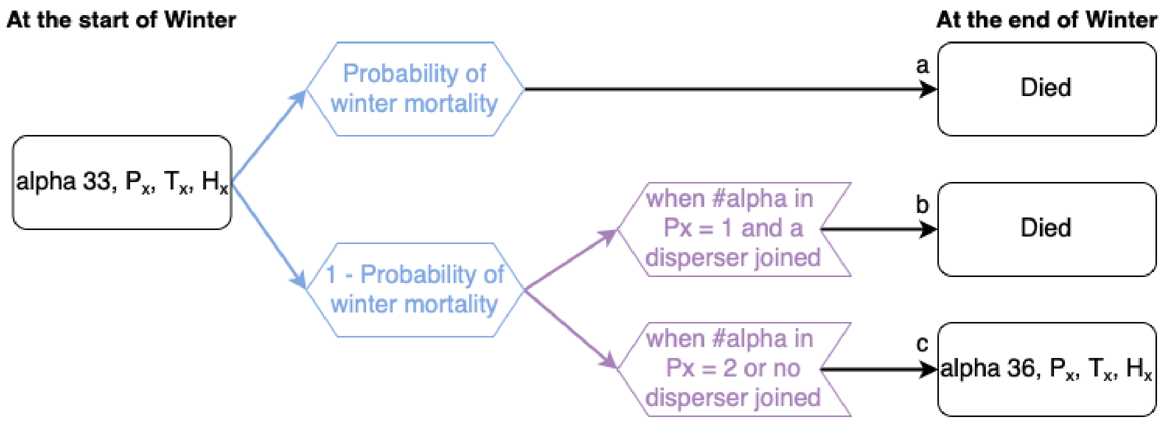
Table 3.4: Possible alpha status for different age groups.

	Autumn	**Winter**	**Spring**	**Summer**
Pup	N/A	N/A	Non-Alpha	Non-Alpha
Juvenile	Non-Alpha	Non-Alpha	Non-Alpha	Non-Alpha
Yearling	Non-Alpha	Non-Alpha	Non-Alpha or Alpha (Figure 3.8)	Non-Alpha or Alpha (Figure 3.10, 3.11)
Adult	Non-Alpha or Alpha (Figure 3.10,3.11)	Non-Alpha or Alpha (Figure 3.12,3.13)	Non-Alpha or Alpha	Non-Alpha or Alpha

i Adult Alpha

At the end of winter, alpha wolves may die due to area-based winter mortality (Figure 3.12, Path a). They might also be de-alphalised and die if their mate died during winter and a new wolf joined their pack (Figure 3.12, Path b, and also explained in Figure 3.9c). If their mate dies but no new wolf joins the pack, or if their mate survives, they will continue as the alpha, aged 36 months, with their pack P_x, on their territory T_x, within their habitat H_x (Figure 3.12, Path c).

Figure 3.12: Adult alpha in winter.



ii Adults Non-Alpha

When winter comes, a 33-month old non-alpha wolf at P_a, T_a, H_a goes through the same process as it did during the winter when it was 21 months old (Figure 3.8). It could die due to area-based winter mortality (Figure 3.13, Path a); age without dispersing (Figure 3.13, Path b); die from unsuccessful dispersal (Figure 3.13, Path c); age, find a suitable mate, and become alpha (Figure 3.13, Path d); or age and settle in an empty territory as a non-alpha (Figure 3.13, Path e).

For all mature wolves that survived winter mortality and did not disperse during the previous winter, they go through the same processes every winter. Therefore, apart from age and the probability of dispersal, there is no difference between Figure 3.13 and Figure 3.8.

Table 3.5 summarises the possible states and transitions of all adult wolves at each season at any point in time.

Figure 3.13: Adult non-alpha in winter.

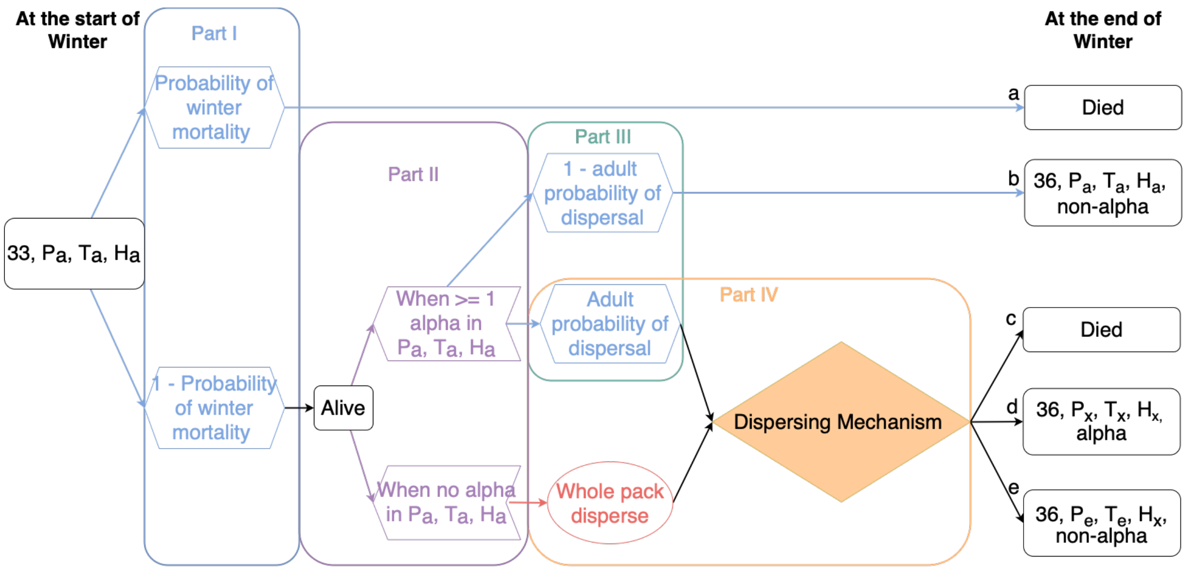


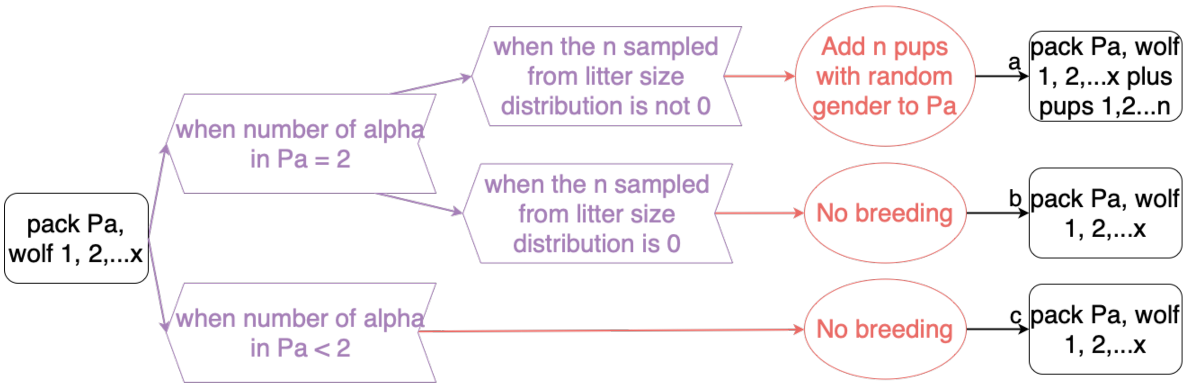
Table 3.5: States and transitions of adults.

Adults (More than 30 Months)				
	Autumn	**Winter**	**Spring**	**Summer**
Alpha	Age	Age or die	Age and breed or de-alphalise and not breed	Age
Not alpha	Age	Age or die or adult disperse	Age	Age

3.5.7 Breeding

Breeding is modelled as a pack-level activity. It is assumed to occur every year at the start of spring when both alpha wolves are alive (Figure 3.14). For every pack with both alpha wolves alive at the start of spring, a litter of pups is added to the pack (Figure 3.14, Path a). The size of each litter is determined based on a litter size distribution, and the gender of each pup is assumed to be equally likely. The model assumes a litter size of 0 is possible (Figure 3.14, Path b). If the pack has only one alpha wolf (regardless of gender), no new pups are added to the pack, and the pack remains unchanged (Figure 3.14, Path c).

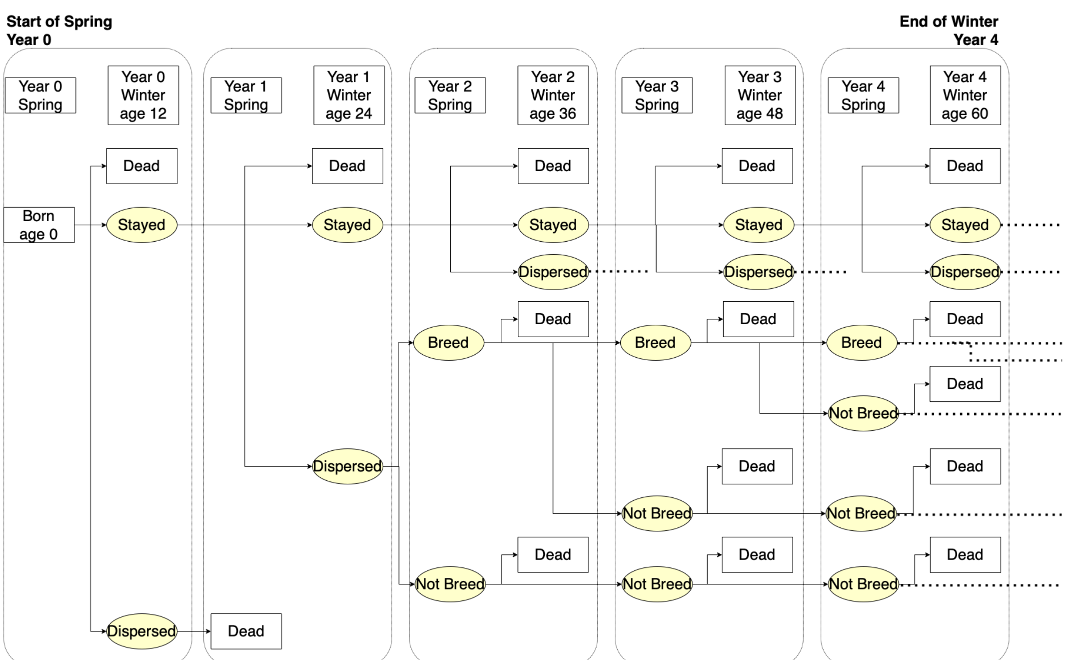
Figure 3.14: Breeding.



3.5.8 Summary Chart

Figure 3.15 summarises the possible states and activities of a single wolf from age 0 to 4 years old.

Figure 3.15: Possible survival, dispersal, and breeding states for a wolf over 4 years.



3.6 Computer Model

Section 3.5 details the conceptual model by tracing the life history of individual wolves. In modelling the entire wolf population within the study area, for computational efficiency, individuals are grouped based on their life stage and wolves at the same stage are simulated collectively. In this chapter, the term 'stage' refers collectively to a wolf's age group, location, and status. As explained in Section 3.5, age groups are categorised as pups (0 to 6 months), juveniles (6 to 18 months), yearlings (18 to 30 months), and adults (over 30 months). The location specifies the specific habitat and territory each wolf is in. The status of a wolf is either living in a group (non-alpha), or leading a group (alpha), or dispersing. The simulation advances in three-month intervals for computational efficiency, also to reflect the seasonal cycle.

The model is coded in Python using a functional programming approach. Two flowcharts explaining the logic of the computer model are provided in Appendix B.1. The model assumptions are summarised in Appendix B.2. Simplifications made for computational efficiency are listed in Appendix B.3. Methods applied for model verification are explained in Appendix B.4.

3.7 Model Parameters

This section explains the values of the model parameters. It is impractical to examine the effect of all model parameters on the solution to the RSS problem. Therefore, the parameters are divided into two categories: Subsection 3.7.1 details the fixed parameters whose values remain unchanged throughout the research, and Subsection 3.7.2 explains the parameters that are varied to create different scenarios of the RSS problem (input parameters).

3.7.1 Fixed Parameters

Below is the list of the fixed parameters and their corresponding values. Unless specified otherwise, the values of these parameters are derived from Haight, Mladenoff, and Wydeven (1998).

- Number of habitats: 6 (Haight and Travis, 2008)
- Probability of dispersal:
  - probability of juvenile wolves dispersing = 25%
  - probability of yearling wolves dispersing = 50%
  - probability of adult wolves dispersing = 90%
  - Long dispersal probability for dispersing wolves = 20%
- Maximum number of territories a dispersing wolf is allowed to visit = 6
- Litter size distribution = uniform distribution from 0 to 12, mean of 6 (Haight, Mladenoff, and Wydeven (1998) uses a model with an empirical distribution that is not detailed in their paper. According to Mech (1970), the average litter size is 6. Therefore, this model uses a uniform distribution with a mean of 6 as the litter size distribution).

3.7.2 Input Parameters

The input parameters defining a study area in the grey wolf simulation model this research developed include: the starting number of territories, the starting packs, the maximum number of core territories, mortality rates for wolves in core and peripheral territories, and mortality rates for pups. Changing these input parameters allows the generation of varied study areas, leading to varied scenarios for the RSS problem. Table

3.6 presents a generalised set of starting conditions for a study area.

Table 3.6: Input table.

		H_1	H_2	...	H_6	
Starting number of territories	[	t_1	t_2	...	t_6	]
Starting packs	[	p_1	p_2	...	p_6	]
Maximum number of core territories	[	c_1	c_2	...	c_6	]
Core territories winter mortality rate	[	cm_1	cm_2	...	cm_6	]
Peripheral territories winter mortality rate	[	prm_1	prm_2	...	prm_6	]
Pup mortality rate	[	ppm_1	ppm_2	...	ppm_6	]

The starting number of territories t is the number of existing territories within each habitat at the beginning of the simulation. $t = 0$ means there is no territory in the corresponding habitat. A habitat without any existing territory is referred to as an empty habitat in this thesis.

The starting packs p are the initial pack configurations in each habitat. This includes the count of packs in each habitat and the constitution of each pack. Pack constitution includes the number of wolves, their ages, genders, and alpha statuses. Starting packs are classified into three categories based on the number of wolves they contain: big packs (denoted as p_b, with 11 or more wolves), medium packs (denoted as p_m, with 5 to 10 wolves), and small packs (denoted as p_s, with 0 to 4 wolves). A value of $p = 0$ means there is no pack in a territory (i.e. empty territory).

The maximum number of core territories, c, represents the number of core territories a habitat can support. For simplicity, the model assumes this number is fixed and is provided by the decision maker. This number means: the first c territories in a habitat are assumed to be core territories, and any additional territories beyond this capacity are categorised as peripheral territories. For example, if $c = 2$ and there is initially only one territory in the habitat, it is a core territory. If one additional territory is selected, it also becomes a core territory. However, if a second additional territory is selected, it becomes a peripheral territory since the maximum number of core territories that the habitat can support has already been reached.

The mortality rates are self-explanatory. The base mortality rates this research adopts are from Haight, Mladenoff, and Wydeven (1998): 20% for core territories' winter mortality rate, 40% for peripheral territories' winter mortality rates, and 40% for pup mortality.

The solution to the problem is expressed in the form of a vector like $[X_{1i}, X_{2i}, X_{3i}, X_{4i}, X_{5i}, X_{6i}]$, where each X_{hi} represents the number of additional territories in Habitat h in solution i (Section 3.3).

3.8 Summary

This chapter provides a foundation for Chapters 4, 5, and 6 by presenting the essential background needed to address the research questions detailed in Chapter 2. Sections 3.2 to 3.4 explain the grey wolf RSS problem that this research focuses on and detail the formulation and solution method for the problem. Sections 3.5 to 3.7 detail the grey wolf model developed in this thesis and highlight the input parameters that Chapters 4, 5, and 6 focus on.

Variations in the input parameters enable the generation of different scenarios for the RSS problem. Applying CCSB-F to these scenarios allows for the evaluation of its performance under various characteristics. Chapters 4, 5, and 6 explore the three scenarios. Scenario 1 uses real-world wolf data to assess the applicability and effectiveness of CCSB-F. Scenario 2 introduces specific characteristics designed to evaluate the use of a model-based approach with CCSB-F. Scenario 3 has more potential solutions than

Scenario 2, and is designed to evaluate the use of a heuristic approach with CCSB-F.

One of the benefits of solving the RSS problem with SO comes from the simulation model used to estimate grey wolf persistence. There are two main advantages of using a simulation model compared to analytical functions in solving the RSS problem. First, the model can accommodate individual behaviours, interactions, dynamic changes in habitat, and spatial aspects of the habitat, which allows for more realistic estimations of species persistence. Second, the model is very accommodating for various performance measures of interest.

This thesis looks for lessons that can be extrapolated beyond the specific examples in Chapters 4, 5 and 6. Although this thesis focuses particularly on species persistence in the constraint, other measures, such as the location of the packs, the number of pups, or the compactness of the sites, can be easily incorporated into the formulation. The same applies to the objective. This thesis focuses on the number of sites in the objective, but other measures, such as the individual costs of each territory or habitat, can be easily incorporated by amending the objective function.

Chapter 4

Data-Informed Wolf Reserve Site Selection Problem

4.1 Introduction

Chapter 2 identifies the lack of use of SO in the RSS problem, and explains the choice of CCSB-F as the representative SO solution method this research focuses on. Chapter 4 addresses **RQ1: How well do current SO methods perform (in terms of solution accuracy and computational effort) when applied to RSS problems?** To investigate this, this chapter introduces a hypothetical RSS problem, referred to as Scenario 1, which uses real-world information from California wolf packs.

The specifics of Scenario 1 are presented in Section 4.2. The chance-constrained formulation of Scenario 1 and its parameters' settings are detailed in Section 4.3. To assess CCSB-F's effectiveness in solving Scenario 1, multiple replications of CCSB-F are performed. The results are presented in Section 4.4. An analysis of the results is provided in Section 4.5. To examine the efficiency of CCSB-F, it is compared to a standard hypothesis testing procedure for solving probabilistically constrained problems. The comparison and its result are explained in Section 4.6. Section 4.7 provides a summary of this chapter.

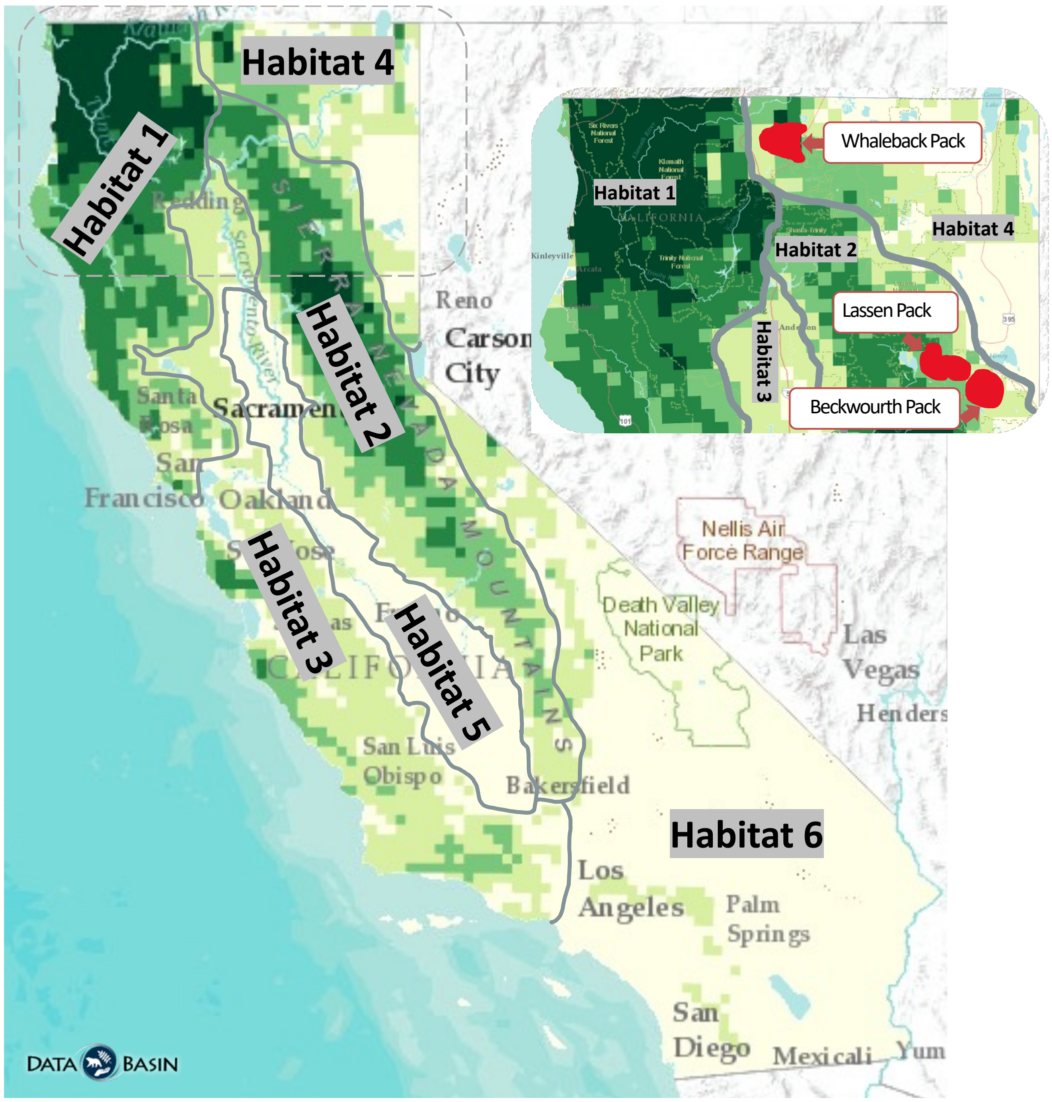
4.2 Scenario 1: Problem Description

This section describes the Scenario 1 problem. As previously discussed in Section 3.2, to define an RSS problem, the high-level information required is: a study area, a budget, a conservation goal, and a desired probability of achieving the conservation goal. The more detailed information required on the study area is the number of habitats, the number of territories in each habitat, the maximum possible number of core territories within each habitat, mortality rates associated with each habitat, and information about existing packs, including pack location, each member wolf's gender, age, and alpha status.

The study area of Scenario 1 is based on the state of California, USA. Scenario 1 assumes the government has a budget for protecting up to two additional territories, with a conservation goal of maintaining at least the starting number of viable packs at the end of a five-year planning horizon, with at least a 75% probability of achieving that goal. The budget is intentionally limited to two additional territories to ensure the problem can be solved within a reasonable timeframe. The objective of Scenario 1 is to identify the minimal number of additional territories required to reach this probability, and to determine the optimal allocation plans for these additional territories.

The study area, California, is divided into six distinct habitats, as shown in Figure 4.1. This division is informed by a map of Antonelli et al. (2016) (downloaded from Boysen (2016)), which illustrates the likelihood of areas being suitable for wolves to establish their pack in. The amplified area on the top right of Figure 4.1 shows the current wolf packs and their location based on CDFW (2023).

Figure 4.1: Map of California with 6 habitats and the current locations of wolf packs.



The starting conditions for Scenario 1 are summarised in Table 4.1. This scenario assumes there are five existing territories in each of Habitats 1 to 4 and none in Habitats 5 and 6. For the maximum number of core territories (explained in Subsection 3.7.2), it assumes there is a maximum of two territories in each of Habitats 1 to 4 that can be core. No territories in Habitats 5 and 6 can be core territories. The mortality rates are assumed to be the same as the base mortality rates in Subsection 3.7.2. The information on existing packs is based on the real conditions in California. Three existing packs in California have been officially documented by the California Department of Fish and Wildlife (CDFW), the detailed packs' constitutions are (CDFW, 2023):

- Lassen pack (Habitat 2): one pair of alphas (the female is around 39 months old, and the male's age is unknown, assumed to be 30 months), five yearlings (assumed to be two males and three females), and five pups (assumed to be three males and two females) - 12 wolves in total. The pack is assumed to be in a core territory.

- Beckwourth pack (Habitat 2): two wolves in total, assumed to be a pair of alphas (the male is around 36 months old, the female's age is unknown, assumed to be 30 months). The pack is assumed to be in a core territory.

- Whaleback pack (Habitat 4): one pair of alphas (the male is around 48 months old, the female's age is unknown, assumed to be 30 months), five yearlings (assumed to be three males and two females), and eight pups (assumed to be four males and four females) – 15 wolves in total. The pack is assumed to be in a core territory.

Table 4.1: Scenario 1: starting condition.

		H_1	H_2	H_3	H_4	H_5	H_6	
Starting number of territories	[	5	5	5	5	0	0	]
Starting packs	[	0	Lassen Beckwourth	0	Whaleback	0	0	]
Maximum number of core territories	[	2	2	2	2	0	0	]
Core territories winter mortality rate	[	0.2	0.2	0.2	0.2	0.2	0.2	]
Peripheral territories winter mortality rate	[	0.4	0.4	0.4	0.4	0.4	0.4	]
Pup mortality rate	[	0.4	0.4	0.4	0.4	0.4	0.4	]

4.3 Scenario 1: Problem Formulation

This section provides the formulation for the Scenario 1 problem. Subsection 4.3.1 explains the formulation. Subsection 4.3.2 details the CCSB-F parameters used and the solution space of Scenario 1.

4.3.1 Chance-Constrained Formulation

Given that there are three viable packs at the beginning of the planning horizon, the conservation goal of maintaining at least the starting number of viable packs is $Y_i \geq 3$, where Y_i is the number of viable packs at the end of the planning horizon, estimated by the simulation model under an additional territories selection plan i (i.e. solution i). The requirement of achieving the conservation goal with at least 75% probability is denoted as $\Pr\{Y_i \geq 3\} \geq 75\%$.

The aim of finding the minimal number and allocation of additional territories in six habitats is written as $\min_{i \in I} \sum_{h=1}^{6} X_{hi}$, where I is the solution space, and X_{hi} is the number of additional territories selected in Habitat h under solution i. The size of the solution space, calculated from Eq.(3.4), is:

$$\sum_{j=0}^{2} C(6+j-1, 6-1) = \sum_{j=0}^{2} \frac{(6+j-1)!}{j!(6-1)!} = 28. \tag{4.1}$$

A chance-constrained formulation of Scenario 1 can be written as:

$$\begin{aligned} &\min_{i=1,2,\ldots,28} \sum_{h=1}^{6} X_{hi} \\ &s.t.\ \Pr\{Y_i \geq 3\} \geq 75\% \end{aligned} \tag{4.2}$$

4.3.2 CCSB-F Initialisation

To calculate CCSB-F's required sample size $n(\beta)$ (for declaring a solution feasible) and the failure count threshold $m_\beta(n)$ (for declaring a solution infeasible), the parameter values for the violation probability γ, the overall error allowance for the procedure α, and the feasibility tolerance level δ_γ need to be defined (detailed in Subsection 2.3.3)

Since the desired probability of achieving the conservation goal is 75%, the violation probability γ is 0.25. The α (significance level for the CCSB-F procedure) is set to 0.05 in Scenario 1. The total number of possible solutions for the Scenario 1 problem is 28 (Eq.(4.1)). The value of the error allowance β is set to $\frac{0.05}{28}$ (explained in Subsection 2.3.3). The full list of all these 28 solutions is presented in Table 4.2.

Table 4.2: Scenario 1: solution space.

Additional Territories	**0**	**1**	**2**
Solutions	[5, 5, 5, 5, 0, 0]	[5, 5, 5, 5, 0, 1]	[5, 5, 5, 5, 0, 2]
		[5, 5, 5, 5, 1, 0]	[5, 5, 5, 5, 1, 1]
		[5, 5, 5, 6, 0, 0]	[5, 5, 5, 6, 0, 1]
		[5, 5, 6, 5, 0, 0]	[5, 5, 6, 5, 0, 1]
		[5, 6, 5, 5, 0, 0]	[5, 6, 5, 5, 0, 1]
		[6, 5, 5, 5, 0, 0]	[6, 5, 5, 5, 0, 1]
			[5, 5, 5, 5, 2, 0]
			[5, 5, 5, 6, 1, 0]
			[5, 5, 6, 5, 1, 0]
			[5, 6, 5, 5, 1, 0]
			[6, 5, 5, 5, 1, 0]
			[5, 5, 5, 7, 0, 0]
			[5, 5, 6, 6, 0, 0]
			[5, 6, 5, 6, 0, 0]
			[6, 5, 5, 6, 0, 0]
			[5, 5, 7, 5, 0, 0]
			[5, 6, 6, 5, 0, 0]
			[6, 5, 6, 5, 0, 0]
			[5, 7, 5, 5, 0, 0]
			[6, 6, 5, 5, 0, 0]
			[7, 5, 5, 5, 0, 0]

There is no straightforward way to set the feasibility tolerance level δ_γ in this thesis. Therefore, two values of δ_γ, 0.025 and 0.010, are used. For $\delta_\gamma = 0.025$, $n(\beta)$ and $m_\beta(n)$ (Eq.(2.19) and Eq.(2.20)) are:

$$\begin{aligned} n(\beta) &= \left\lceil \frac{Z^2_{1-0.05/28}}{0.025^2} \cdot \left(\sqrt{0.25 \cdot (1-0.25)} + \sqrt{(0.25-0.025)\cdot(1-(0.25-0.025))}\right)^2 \right\rceil \\ &= 9,828, \\ m_\beta(n) &= \left\lfloor 9,828 \cdot 0.25 - Z_{1-0.05/28}\sqrt{9,828 \cdot 0.25 \cdot (1-0.25)} \right\rfloor \\ &= 2,331. \end{aligned}$$

Similarly, for $\delta_\gamma = 0.010$, $n(\beta) = 62,805$, and $m_\beta(n) =$ 15,385.

To analyse the performance of CCSB-F in solving Scenario 1, multiple replications of the entire CCSB-F procedure (hereafter macro-replications) need to be performed due to the sampling errors embedded in the simulation experiment. For this chapter, 100 macro-replications are carried out.

4.4 Scenario 1: Results

Table 4.3 presents the feasible solutions and their selection counts from 100 macro-replications of CCSB-F, with feasibility tolerance levels δ_γ of 0.025 and 0.010. Other solutions not listed in the table were not selected in any of the macro-replications. The full result table is in Appendix C.1.

Table 4.3: Scenario 1: result table for 100 macro-replications of CCSB-F.

Additional Territories	**Solutions identified as feasible**	**Selection count**	
		$\delta_\gamma = 0.025$	$\delta_\gamma = 0.010$
2	[5, 5, 5, 5, 1, 1]	20	100
2	[5, 6, 5, 5, 0, 1]	5	74
2	[5, 6, 5, 5, 1, 0]	8	71
	Average number of runs	252,522	1,664,037
	Standard deviation	788	2206

From both δ_γ of 0.025 and 0.010 results, the minimal additional territories required to ensure a 75% probability of achieving the conservation goal is two territories for the Scenario 1 problem (with a 5% significance level). The additional territories selection plans that can achieve this are: one additional territory in both Habitats 5 and 6 ([5, 5, 5, 5, 1, 1]), one in Habitat 2 and one in Habitat 6 ([5, 6, 5, 5, 0, 1]), or one in Habitat 2 and one in Habitat 5 ([5, 6, 5, 5, 1, 0]).

All optimal solutions involve protecting a territory in at least one empty habitat (Habitat 5 or 6). This is largely dependent on the modelling assumption that long dispersers (those who disperse to a different habitat, see Section 3.5) have a uniform probability of moving into any other habitat, and cannot survive if there are no territories in the habitat they move into. Therefore, having more non-empty habitats increases their likelihood of survival and, in turn, increases the overall chance of achieving the conservation goal. This aligns with the findings of Haight, Mladenoff, and Wydeven (1998).

Another observation is that two of the three optimal solutions involve allocating an additional territory to Habitat 2. This result is closely linked to the wolf model's assumption that dispersers can only survive if they find either a suitable pack or an empty territory. As Habitat 2 is the most densely populated among the six habitats, dispersers (both local and long) in Habitat 2 face a higher likelihood of failing to find a suitable pack or empty territory compared to other habitats. Hence, allocating an additional territory to Habitat 2 is likely to reduce this risk, thereby enhancing disperser survival more effectively than same allocations in other habitats.

4.5 Scenario 1: CCSB-F Performance Analysis

The performance of the CCSB-F is analysed in this section based on the level of computational effort required and the probability of correct selection. Subsection 4.5.1 explores the selection accuracy of CCSB-F, i.e. the probability of correctly identifying any optimal solutions at the end of the procedure. Subsection 4.5.2 provides an analysis of the computational effort (measured by the number of simulation runs) required by CCSB-F.

4.5.1 Selection Accuracy

Since the cost of a solution is deterministic in the problem studied in this thesis, once the feasibility of the solution is determined, a correct selection is defined as selecting any feasible solution that uses the least additional territories. Therefore, to evaluate the selection accuracy of CCSB-F, the feasibility of all solutions needs to be checked. However, due to the sampling error embedded in simulation experiments, the true feasibility of solutions cannot be known with certainty. One way to estimate this is by running the simulation multiple times and calculating a confidence interval for the true probability of the solutions.

i Feasibility Estimation

To estimate the feasibility, each solution is run 100,000 times. Table 4.4 lists the solutions that are identified as feasible (at a 5% significance level). Figure 4.2 is a visual representation of the full result. The full result table, including each solution's test statistics, is in Appendix C.2.

Table 4.4: Scenario 1: feasibility estimation for solutions identified as feasible.

Additional Territories	**Solutions**	**Pr**$\{Y_i \geq 3\}$	**Lower 90% CI**	**Upper 90% CI**	**Pr**$\{Y_i < 3\}$
2	[5, 5, 5, 5, 1, 1]	0.75999	0.75605	0.76393	0.24001
2	[5, 6, 5, 5, 0, 1]	0.75818	0.75423	0.76213	0.24182
2	[5, 6, 5, 5, 1, 0]	0.75483	0.75087	0.75879	0.24517

Figure 4.2: Scenario 1: feasibility estimation result plot.

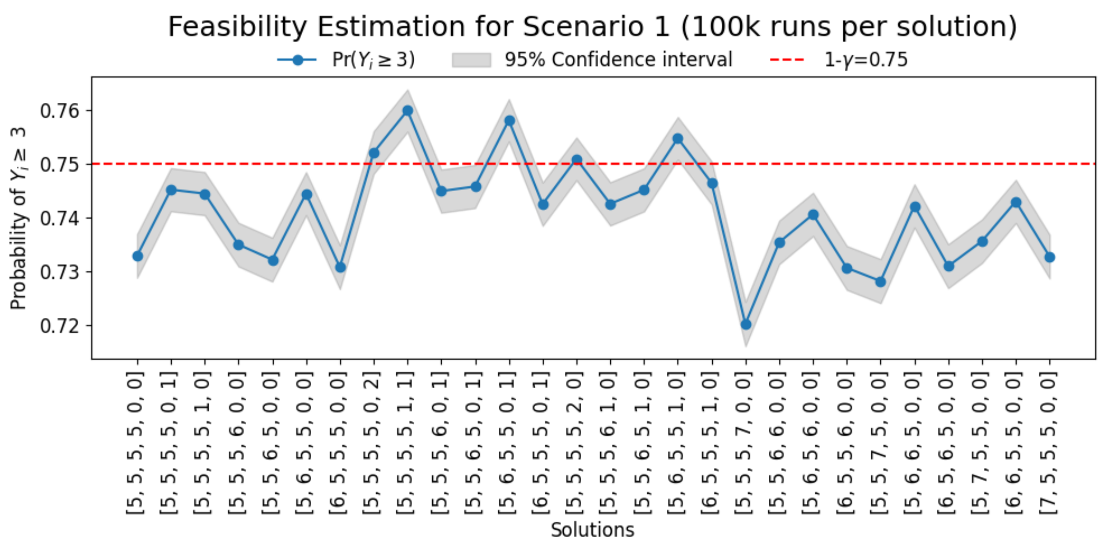


Figure 4.2 shows that five solutions have Pr$\{Y_i \geq 3\}$ above the 75% line, yet only three are listed as feasible in Table 4.4. This is because the t-tests, based on 100,000 simulation runs for each solution, find insignificant evidence to confirm the feasibility of two solutions at a 5% significance level ([5,5,5,5,0,2] with a mean of 0.75218 and

[5,5,5,5,2,0] with a mean of 0.751). Consequently, these two solutions are not considered feasible in the following analyses.

ii Probability of Correct Selection

Based on the feasibility estimation results, solutions [5, 5, 5, 5, 1, 1], [5, 6, 5, 5, 0, 1] and [5, 6, 5, 5, 1, 0] are feasible at 5% significance level. As the minimal cost of any feasible solution is 2 territories, these three solutions are also the optimal solutions. Recall from Table 4.3 that the only solutions CCSB-F selects are these three solutions. The total number of times CCSB-F has made any selection is 31 when $\delta_\gamma = 0.025$, and 100 when $\delta_\gamma = 0.010$. Since all solutions selected by CCSB-F are optimal, the probability of correct selection based on 100 macro-replications for $\delta_\gamma = 0.025$ is estimated at 31%, and for $\delta_\gamma = 0.010$ it is 100%. This result aligns with the theory that with an appropriate feasibility tolerance level, CCSB-F can have a very high level of accuracy in selecting an optimal RSS plan.

The result also shows the extent to which selection accuracy depends on the feasibility tolerance level δ_γ, as explained in Subsection 2.3.3. In Scenario 1, with $\delta_\gamma = 0.025$, CCSB-F does not have explicit control over the Type II errors for solutions whose violation probability $\Pr\{Y_i < 3\}$ falls within the range of (0.225, 0.25] (Subsection 2.3.3). This range shifts to (0.24, 0.25] when δ_γ decreases to 0.010. According to the feasibility estimation results (Table 4.4), the $\Pr\{Y_i < 3\}$ of all three optimal solutions (Table 4.3) falls within these intervals for both δ_γ values. Hence, a decrease in δ_γ is expected to result in an increase in selection counts for all three solutions.

4.5.2 Computational Effort

The computational effort for the Scenario 1 problem, based on the 100 macro-replications results collected (Table 4.3) is the following: with $\delta_\gamma = 0.010$, the average number of runs per CCSB-F macro-replication required is 1,664,037, the 95% confidence interval is 1,664,469 – 1,663,604. When $\delta_\gamma = 0.025$, the average is 252,522 runs, and the 95% confidence interval is 252,677 - 252,368.

As expected with CCSB-F (see Subsection 2.3.3), a smaller δ_γ requires an increase in the number of simulation runs. In Scenario 1, for a 60% decrease in δ_γ (from 0.025 to 0.010), CCSB-F requires a 558.97% increase in the number of runs on average (from 252,522 to 1,664,037) to solve the Scenario 1 problem.

4.6 Benchmark and CCSB-F Comparison

Since there is no other established solution method for probabilistically constrained SO problems with a deterministic objective, a standard statistical hypothesis testing procedure (e.g. Anderson et al. (2014)) that uses power analysis to control the probabilities of making both Type I and II errors is used as a benchmark. Subsection 4.6.1 describes the benchmark procedure and compares its computational effort with CCSB-F. Subsection 4.6.2 compares the selection accuracy of the benchmark and CCSB-F in solving the Scenario 1 problem.

4.6.1 Computational Effort

A standard hypothesis test for testing the population mean μ against the hypothesised value of the population mean μ_0 can be expressed as:

$$H_0 : \mu > \mu_0 \text{ vs. } H_1 : \mu \leq \mu_0 \tag{4.3}$$

$$\text{Type I error rate: } \Pr\{\text{reject } H_0 \mid \mu > \mu_0\} \tag{4.4}$$

$$\text{Type II error rate: } \Pr\{\text{do not reject } H_0 \mid \mu \leq \mu_0\} \tag{4.5}$$

The Type I error may be controlled by setting $\Pr\{\text{reject } H_0 \mid \mu > \mu_0\} \leq \beta_1$. The Type II error cannot be controlled for all values, but suppose one would like to control it at the value of μ less than (or equal to) a specific μ_1 such that:

$$\Pr\{\text{do not reject } H_0 \mid \mu \leq \mu_1\} \leq \beta_2 \tag{4.6}$$

Then, the sample size n should satisfy Eq.(4.7).

$$\mu_0 - \mu_1 = Z_{\beta_1} \frac{\sigma_1}{\sqrt{n}} + Z_{\beta_2} \frac{\sigma_2}{\sqrt{n}} \tag{4.7}$$

Recall from Subsection 2.3.3, the above hypothesis test and the selection errors in CCSB-F are denoted as:

$$H_0 : p_i > \gamma \text{ vs. } H_1 : p_i \leq \gamma \tag{4.8}$$

$$\text{Type I error control: } \Pr\{\text{reject } H_0 \mid p_i > \gamma\} \leq \beta_1 \tag{4.9}$$

$$\text{Type II error control: } \Pr\{\text{do not reject } H_0 \mid p_i \leq \gamma - \delta_\gamma\} \leq \beta_2 \tag{4.10}$$

Comparing the CCSB-F hypothesis testing in Eq.(4.8) with the standard hypothesis testing in Eq.(4.3), it is clear that γ is μ_0. Comparing the CCSB-F Type II error in Eq.(4.10) and the Type II error control in Eq.(4.6), it is clear $\gamma - \delta_\gamma$ is μ_1. Then, σ_1 is the same as σ_γ and σ_2 is $\sigma_{\gamma-\delta_\gamma}$. Note that $\mu_0 - \mu_1$ in Eq.(4.7) in CCSB-F is $\gamma - (\gamma - \delta_\gamma) = \delta_\gamma$.

Because the outcome of $\{Y_i \geq N\}$ follows a Bernoulli distribution, $\sigma_\gamma = \sqrt{\gamma(1-\gamma)}$ and $\sigma_{\gamma-\delta_\gamma} = \sqrt{(\gamma-\delta_\gamma)(1-\gamma+\delta_\gamma)}$. Also, β_1 and β_2 are set to $\beta_1 = \beta_2 = \beta$ in CCSB-F. Therefore, the Eq.(4.7) becomes:

$$\delta_\gamma = Z_\beta \frac{\sigma_\gamma}{\sqrt{n}} + Z_\beta \frac{\sigma_{\gamma-\delta_\gamma}}{\sqrt{n}} \tag{4.11}$$

Rearranging the terms in Eq.(4.11) and substituting the values of σ_γ and $\sigma_{\gamma-\delta_\gamma}$ gives the formula to calculate the sample size n:

$$n = \frac{Z_\beta^2}{\delta_\gamma^2} \cdot \left(\sqrt{\gamma(1-\gamma)} + \sqrt{(\gamma-\delta_\gamma)(1-(\gamma-\delta_\gamma))}\right)^2 \tag{4.12}$$

This sample size is the same as the sample size $n(\beta)$ CCSB-F requires to declare a solution feasible. Nevertheless, because of the sequential feature of CCSB-F, it can declare a solution as infeasible when the solution fails $m_\beta(n)$ times before collecting all $n(\beta)$ samples. Hence, for some clearly infeasible solutions, CCSB-F does not need to collect all $n(\beta)$ samples. As soon as the fail counts reach $m_\beta(n)$, it can declare the solution as infeasible and move to the next solution. However, regardless of feasibility, the benchmark needs to collect n samples for every solution. Therefore, the benchmark is expected to be more computationally intensive (i.e. requiring more number of runs) than CCSB-F.

For the Scenario 1 problem, with 28 solutions and a tolerance level of $\delta_\gamma = 0.025, \beta = 0.05/28$, the sample size the benchmark needs to collect for each solution is calculated by Eq.(4.12) as:

$$n = \frac{Z^2_{0.05/28}}{0.025^2} \cdot \left(\sqrt{0.25 \cdot (1 - 0.25)} + \sqrt{(0.25 - 0.025) \cdot (1 - (0.25 - 0.025))}\right)^2$$
$$= 9,828.$$

The benchmark requires a total sample size of 275,184 ($9,828 \cdot 28$) for 28 solutions. Compared to this, CCSB-F requires an average of 252,522 samples (Table 4.3), which is 8% fewer runs per macro-replication compared to the benchmark. For a smaller $\delta_\gamma = 0.01$, the benchmark requires 62,805 runs per solution (1,758,540 per macro-replication). CCSB-F requires 5% fewer runs on average (1,664,037 per macro-replication) compared to the benchmark.

As explained in Subsection 2.3.3, the benefit of the CCSB-F depends on the solution space characteristics of the problem. For Scenario 1, as shown in Subsection 4.5.1, most solutions are not far from being feasible, which means CCSB-F needs to collect more samples (close to $n(\beta)$) to determine the feasibility of a solution. Thus, the total sample size is close to $n(\beta) \cdot 28$. On the other hand, if all solutions in Scenario 1 are clearly infeasible (i.e. all simulation outputs have $\Pr\{Y_i < N\}$), then the total number of samples CCSB-F needs to collect is $m_\beta(n) \cdot 28$ (65,268 for $\delta_\gamma = 0.025$, and 430,780 for $\delta_\gamma = 0.01$). In such an extreme case, compared to the benchmark, CCSB-F's computational effort savings will be 76.28% (275,184 to 65,268) for $\delta_\gamma = 0.025$, and 75.50% (1,758,540 to 430,780) for $\delta_\gamma = 0.010$.

4.6.2 Selection Accuracy

To compare the selection accuracy between the benchmark and CCSB-F, two sets of 100 macro-replications are performed with $\delta_\gamma = 0.025$ and 0.010 to compare with the previous two sets of CCSB-F results. The full result table and the experiment settings for the benchmark are in Appendix C.3.

For $\delta_\gamma = 0.025$, the benchmark identified the optimal solutions 31 times out of 100 replications, and 100/100 when $\delta_\gamma = 0.01$. Table 4.5 compares the number of correct selections of the benchmark and CCSB-F made. Neither method selected any infeasible solutions.

Table 4.5: Benchmark and CCSB-F selection count comparison with two δ_γs.

δ_γ	$\delta_\gamma = 0.025$		$\delta_\gamma = 0.01$	
Solutions	**Benchmark**	**CCSB-F**	**Benchmark**	**CCSB-F**
[5, 5, 5, 5, 1, 1]	19	20	98	100
[5, 6, 5, 5, 0, 1]	7	5	78	74
[5, 6, 5, 5, 1, 0]	12	8	74	71
Total correct selection count	31	31	100	100

There is no difference between the benchmark and CCSB in the total selection count based on 100 macro-replications. However, two noticeable differences were observed. The first is in solution [5, 6, 5, 5, 1, 0] when $\delta_\gamma = 0.025$, which the benchmark selected 12 times, but CCSB-F selected 8 times. The second is in solution [5, 6, 5, 5, 0, 1], which the benchmark selected 78 times, but CCSB-F selected 74 times for $\delta_\gamma = 0.01$. Two Chi-squared tests were conducted to test the differences at a 5% significance level. Both tests suggest that there is insufficient statistical evidence to say that the benchmark and CCSB perform differently. Details of the Chi-squared tests are in Appendix C.4.

The findings from the 100 macro-replications indicate that CCSB-F requires fewer runs than the benchmark, and there is no statistical evidence suggesting that the benchmark and CCSB-F differ in selection accuracies.

4.7 Summary

To address **RQ1: How well do current SO methods perform (in terms of solution accuracy and computational effort) when applied to RSS problems?**, Chapter 4 uses a SO solution method, CCSB-F, to solve a hypothetical grey wolf RSS problem that is informed by real-world data (referred to as Scenario 1). Sections 4.2 to 4.5 describe the Scenario 1 problem, its chance-constrained formulation, and the results of experiments designed to study the performance of CCSB-F in solving the problem.

As expected, the results from solving the Scenario 1 problem show that CCSB-F achieves a 31% selection accuracy with the feasibility tolerance level δ_γ of 0.025, and a 100% selection accuracy when δ_γ is 0.010. This demonstrates that CCSB-F can provide high selection accuracy when the appropriate level of feasibility tolerance is selected. Also, as expected, the results show this high level of accuracy comes with a cost of computation intensity: for a small problem with 28 solutions, a single CCSB-F procedure requires around 1.66 million simulation runs for an overall selection accuracy of 95% with a 1% feasibility tolerance level.

This computational intensity also depends on how close the solutions are to the feasibility boundary. As explained in Subsection 2.3.3, a solution space mainly consisting of solutions close to the feasibility boundary will require greater computational effort. In Scenario 1, all solutions are relatively close to the feasibility boundary (see Appendix C.2). Consequently, CCSB-F ends up collecting more samples in total compared to a scenario where more solutions are far from the feasibility boundary. Hence, the total number of samples required is close to the maximum sample size required, which is $n(\beta) \cdot 28 = 1,758,540$.

The results also show the extent of the tradeoff between computational intensity and selection accuracy. In Scenario 1, an improvement in selection accuracy from 31% to 100% is achieved through an increase in computational effort from 252,522 to 1,664,037. This computational intensity is addressed with two solution space filtering approaches discussed in Chapters 5 and 6.

Section 4.6 further examines CCSB-F by comparing its computational effort to that of a standard hypothesis testing procedure for solving probabilistically constrained problems. The comparison results show that the computational savings achieved by CCSB-F in Scenario 1 are relatively small (5% when $\delta_\gamma = 0.010$, and 8% for $\delta_\gamma = 0.025$). This small reduction in computational effort is explained by Scenario 1's solutions' closeness to the feasibility boundary. Conversely, CCSB-F can be expected to save more compared to the benchmark in problems where the solution space mainly consists of infeasible solutions far from the feasibility boundary.

Chapter 5

A Model-Based Approach for Solution Space Filtering

5.1 Introduction

Chapter 4 shows that the computational effort required can be a challenge when solving the chance-constrained formulation of the wolf RSS problem using CCSB-F. In response, this chapter proposes an approach to reduce this computational effort without affecting the statistical guarantee of the CCSB-F procedure.

RnS algorithms that provide a PCS guarantee typically examine all solutions in the solution space (see Subsection 2.3.2). However, some solutions, if they have equivalent alternatives which can be identified based on the characteristics of the simulation model, maybe 'temporarily' removed.

An approach of this kind is developed in this chapter to reduce the computational effort CCSB-F requires in solving the RSS problem. Since it is based on the characteristics of the simulation model, it is referred to here as the model based approach (MBA).

Section 5.2 explains the MBA in detail. To demonstrate the scale and effectiveness of MBA in reducing the computational effort required to solve the RSS problem, it is applied to a hypothetical grey wolf RSS scenario with 28 solutions, referred to as Scenario 2. Section 5.3 describes Scenario 2, details the application of MBA to it, and presents the problem formulation. Section 5.4 compares the computational effort and selection accuracy of using MBA with CCSB-F (referred to as M-CCSB-F) to the use of CCSB-F alone to assess the effectiveness of MBA. Section 5.5 assesses the generalisability of the MBA by applying it to Scenario 1. Section 5.6 provides a summary of the findings.

5.2 A Model Based Approach

To explain the MBA, Subsection 5.2.1 introduces the general concept and explains why and how it can be applied to the grey wolf RSS problem. To demonstrate the approach's potential for reducing computational effort, Subsection 5.2.2 provides formulae for calculating the size of the filtered solution space when using MBA, and two examples. Then, Subsection 5.2.3 demonstrates the potential computational effort saving of M-CCSB-F compared to using CCSB-F alone.

5.2.1 Logic Behind MBA

In the grey wolf simulation model developed in this research, the total number of viable packs of any habitat at the end of a simulation run depends on two factors: the starting condition of the habitat, and the chance of having a long disperser moving into the habitat (see Section 3.5).

The model assumes that the probability distribution of long dispersers moving to other habitats is uniform (i.e. the likelihood of long dispersers moving to any other habitat is equal), which means all habitats have the same probability of receiving long dispersers.

Hence, if the starting conditions of the habitats are identical, these habitats will end up with the same probability distribution of the number of viable packs.

For example, denote Habitat 1's starting condition as $[H1_1, ..., H1_n]$, where each $H1_i$ is a set that contains all relevant information for the ith territory in Habitat 1. Habitat 2's starting condition is $[H2_1, ...H2_n]$. Assuming that Habitats 1 and 2 have identical starting conditions, i.e. $[H1_1, ..., H1_n] = [H2_1, ...H2_n]$. Assuming there is a set of additional territories selected in Habitats 1 and 2, denoted as t, and since all additional territories are assumed to be identical, $[H1_1, ..., H1_n, t] = [H2_1, ...H2_n, t]$. Then, probability distribution of the output of $[H1_1, ..., H1_n, t]$ will be the same as $[H2_1, ...H2_n, t]$. Hence, Habitats 1 and 2 are indistinguishable. Such habitats are referred to as 'indistinguishable habitats' here.

Hence, solutions that only vary by the locations of the additional territories among indistinguishable habitats will have equal outcomes (these solutions are referred to as equivalent solutions here). Therefore, only one of any set of equivalent solutions needs to be simulated, and other equivalent solutions do not require simulation (referred to here as temporarily removed). At the end of the selection procedure, if any optimal (or not optimal) solution has equivalent solutions, these equivalent solutions are also optimal (or not optimal).

Since all solutions are theoretically checked, and those temporarily removed solutions are also considered at the end of the procedure, the statistical guarantee of CCSB-F (see Section 3.4) remains.

5.2.2 Number of Solutions with Indistinguishable Habitats

For an RSS problem with both distinguishable and indistinguishable habitats, to find all possible combinations of additional territories in habitats (i.e. the number of solutions), two types of combination problems need to be considered. The first one is allocating territories into distinguishable habitats, and the second one is allocating territories into indistinguishable habitats.

The calculation for the former combination problem is explained in Section 3.3. The latter problem is equivalent to finding the number of ways of partitioning $b \in \mathbf{N}$ integers (e.g. territories) into at most $h \in \mathbf{N}$ integers (e.g. indistinguishable habitats) (Rosen, 2007, p.378).

The formula for the number of combinations ($P_h(b)$) of allocating exactly b territories into h indistinguishable habitats is (Rosen, 2007, p.310):

$$P_h(b) = \begin{cases} 1 & \text{if} \quad b \in \{0, 1\} \\ 1 & \text{if} \quad h \in \{0, 1\} \\ P_b(b) & \text{if} \quad b < h \\ 1 + P_{b-1}(b) & \text{if} \quad b = h > 1 \\ P_{h-1}(b) + P_h(b-h) & \text{if} \quad b > h > 1 \end{cases} \tag{5.1}$$

For the case where there is only a single group of indistinguishable habitats, by the product rule, the total number of combinations for allocating exactly b additional territories into H habitats, where h habitats are indistinguishable and $H - h$ habitats are distinguishable, is given by (see Eq.(3.3) for $C(\cdot\,,\cdot)$):

$$\sum_{i=0}^{b} C(H - h + i - 1, H - h - 1) \cdot P_h(b - i) \tag{5.2}$$

Considering all possibilities from allocating 0 up to a budget of B additional territories, the total number of solutions is:

$$\sum_{b=0}^{B}\sum_{i=0}^{b} C(H-h+i-1, H-h-1) \cdot P_h(b-i) \tag{5.3}$$

When there is more than one group of indistinguishable habitats, the calculation becomes more complicated. Assume there are G groups of indistinguishable habitats, with each group k containing h_k indistinguishable habitats ($k \in \{1, 2, ..., G\}$). The number of distinguishable habitats is denoted as H_d, and $H_d = H - \sum_{k=1}^{G} h_k$.

Define b_k as the number of territories allocated to group k, and b_d as the number of territories allocated to distinguishable habitats (i.e. $b_d = b - \sum_{k=1}^{G} b_k$). Then, the total number of distinct solutions for allocating exactly b territories is:

$$\sum_{b_1=0}^{b}\sum_{b_2=0}^{b-b_1}\sum_{b_3=0}^{b-b_1-b_2}\cdots\sum_{b_G=0}^{b-\sum_{k=1}^{G-1} b_k} C(H_d+b_d-1, H_d-1) \cdot \prod_{k=1}^{G} P_{h_k}(b_k) \tag{5.4}$$

Considering all budget levels from 0 to B, the total number of solutions is:

$$\sum_{b=0}^{B}\sum_{b_1=0}^{b}\sum_{b_2=0}^{b-b_1}\cdots\sum_{b_G=0}^{b-\sum_{k=1}^{G-1} b_k} C(H_d+b_d-1, H_d-1) \cdot \prod_{k=1}^{G} P_{h_k}(b_k) \tag{5.5}$$

Table 5.1 provides two examples illustrating the number of solutions for the case where there is only a single group of indistinguishable habitats: one with a study area containing a total of four habitats, and another with six habitats. The number of solutions is calculated for the number of indistinguishable habitats up to the total habitats number and for budget levels (B) from 1 to 10. Note that when the number of indistinguishable habitats equals one, this is the same as all habitats being distinguishable.

Table 5.1: Number of solutions $|I|$ corresponding to the number of indistinguishable habitats up to H, for B ranging from 1 to 10, when H is (a) 4; (b) 6.

Total number of habitats $H = 4$				
	No. indistinguishable habitats			
	1	2	3	4
B				
1	5	4	3	2
2	15	11	7	4
3	35	24	14	7
4	70	46	25	12
5	126	80	41	18
6	210	130	64	27
7	330	200	95	38
8	495	295	136	53
9	715	420	189	71
10	1001	581	256	94

(a)

Total number of habitats $H = 6$						
	No. indistinguishable habitats					
	1	2	3	4	5	6
B						
1	7	6	5	4	3	2
2	28	22	16	11	7	4
3	84	62	41	25	14	7
4	210	148	91	51	26	12
5	462	314	182	95	45	19
6	924	610	337	166	74	30
7	1716	1106	587	275	116	44
8	3003	1897	973	437	176	64
9	5005	3108	1548	670	259	90
10	8008	4900	2379	997	372	125

(b)

From Table 5.1, it is clear that the total number of solutions reduces with an increasing number of indistinguishable habitats. The scale of this reduction depends on the problem size. The larger the problem, the higher the reduction percentage.

5.2.3 Potential Computational Effort Reduction

In CCSB-F, the sample size required depends on the number of solutions because the value of β is set to $\frac{\alpha}{|I|}$ to accommodate a Bonferroni correction. Hence, the greater the number of solutions, the smaller the β will need to be. From CCSB-F's sample size calculations (Eq.(2.19) and Eq.(2.20)), assuming other parameter values remain constant, reducing β increases the sample size required to declare a solution feasible ($n(\beta)$) and to declare a solution infeasible ($m_\beta(n)$).

Since the actual sample size required by CCSB-F depends on the problem, the actual computational effort savings of M-CCSB-F also depend on the problem. To show the potential computational effort savings of M-CCSB-F, Tables 5.2 and 5.3 provide two examples of the extremes in computational effort required when taking habitat indistinguishability into account, for budget levels (B) from 1 to 10.

Note that when the number of indistinguishable habitats equals one, this is equivalent to all habitats being distinguishable. Hence, the computational effort required in this case is the same as that required for CCSB-F without solution space filtering. The violation probability γ for Tables 5.2 and 5.3 is set at 0.25, the significance level α is set at 0.05, and the feasibility tolerance level δ_γ at 0.025.

Table 5.2a specifies the upper limits ($n(\beta) \cdot |I|$) on the total number of simulation runs (in 10,000) as the number of indistinguishable habitats increases from one to four for four habitats, Table 5.2b specifies the lower limits ($m_\beta(n) \cdot |I|$). Table 5.3a details the upper limits for the six habitats example, and Table 5.3b provides the lower limits.

Table 5.2: Upper and lower limits on the number of simulation runs required (in 10,000) for $|I|$ solutions for $H = 4$ habitats.

Total number of habitats $H = 4$				
	No. indistinguishable habitats			
B	1	2	3	4
1	3.13	2.33	1.57	0.89
2	12.78	8.67	4.86	2.33
3	36.05	22.81	11.73	4.86
4	82.40	50.04	23.98	9.67
5	164.19	96.46	43.60	16.02
6	296.75	170.27	74.12	26.33
7	498.60	280.52	118.04	39.80
8	791.51	438.55	179.45	59.25
9	1200.56	656.59	262.79	83.79
10	1754.25	949.24	372.71	116.58

(a) Upper limit

Total number of habitats $H = 4$				
	No. indistinguishable habitats			
B	1	2	3	4
1	0.74	0.55	0.37	0.21
2	3.03	2.06	1.15	0.55
3	8.55	5.41	2.78	1.15
4	19.55	11.87	5.69	2.29
5	38.95	22.88	10.34	3.80
6	70.39	40.39	17.58	6.25
7	118.27	66.54	28.01	9.44
8	187.80	104.05	42.57	14.06
9	284.86	155.78	62.35	19.88
10	416.22	225.20	88.42	27.65

(b) Lower limit

Table 5.3: Upper and lower limits on the number of simulation runs required (in 10,000) for $|I|$ solutions for $H = 6$ habitats.

Total number of habitats $H = 6$						
	No. indistinguishable habitats					
	1	2	3	4	5	6
B						
1	4.86	3.98	3.13	2.33	1.57	0.89
2	27.52	20.51	13.85	8.67	4.86	2.33
3	102.15	71.38	43.60	23.98	11.73	4.86
4	296.75	197.98	112.23	56.59	25.15	9.67
5	731.81	471.06	251.56	118.04	48.74	17.13
6	1603.14	1003.08	510.72	226.16	87.99	29.92
7	3209.78	1962.38	960.39	404.64	149.09	47.45
8	5986.18	3590.07	1699.15	686.92	242.00	74.12
9	10540.03	6219.11	2860.55	1115.48	377.75	110.78
10	17694.48	10295.88	4620.49	1746.35	571.73	162.68

(a) Upper limit

Total number of habitats $H = 6$						
	No. indistinguishable habitats					
	1	2	3	4	5	6
B						
1	1.15	0.94	0.74	0.55	0.37	0.21
2	6.53	4.86	3.28	2.06	1.15	0.55
3	24.23	16.93	10.34	5.69	2.78	1.15
4	70.39	46.98	26.63	13.43	5.96	2.29
5	173.62	111.75	59.68	28.01	11.57	4.06
6	380.32	237.96	121.15	53.65	20.88	7.10
7	761.56	465.52	227.87	96.00	35.37	11.26
8	1420.12	851.75	403.11	162.96	57.41	17.58
9	2500.50	1475.37	678.64	264.65	89.61	26.28
10	4197.79	2442.65	1096.24	414.35	135.63	38.59

(b) lower limit

5.3 Scenario 2

While the potential savings can be substantial (as shown in Section 5.2), the actual impact of indistinguishable habitats on the computational effort required depends on the problem. The Scenario 2 problem is designed to show the actual computational effort savings in an RSS problem. Subsection 5.3.1 describes Scenario 2. Subsection 5.3.2 presents the chance-constrained formulation of the Scenario 2 problem. Subsection 5.3.3 details the possible solutions that can be ignored from the Scenario 2 solution space due to the presence of indistinguishable habitats. Subsection 5.3.4 presents the results of the Scenario 2 problem for both the M-CCSB-F and CCSB-F methods.

5.3.1 Scenario 2: Problem Description

Scenario 2 assumes the government has a budget for two additional territories, a conservation goal of having at least four viable packs at the end of a five-year planning horizon, and a target probability of at least a 75% chance of achieving such a goal. The government aims to find the best additional territory allocation plans that not only achieve this probability but also minimise the number of additional territories used.

The hypothetical study area of the Scenario 2 problem has six habitats. Habitats 1 to 4 each have four territories in which a unique pack occupies the first territory, and the other three territories are empty. Habitats 5 and 6 are empty. There are a total of four packs at the beginning. Table 5.4 details the wolves in each pack. For simplicity, all wolves in the same age group are assumed to have the same starting age, i.e. all pups are 6 months old, all yearlings are 18 months old, all adults are 30 months old, and all alpha wolves are 54 months old. Wolves' gender is assumed to be 50% male and 50% female.

All six habitats are assumed to have a maximum number of 2 core territories (explained in Subsection 3.7.2). Mortality rates are the base mortality rates explained in Subsection 3.7.2. The starting condition of Scenario 2 is summarised in Table 5.5.

Table 5.4: Scenario 2: packs information.

	Number of wolves in each group				
Packs	Alpha	Adults	Yearlings	Pups	Total
p_1	2	2	4	4	12
p_2	2	2	2	4	10
p_3	2	0	2	2	6
p_4	2	0	0	2	4

Table 5.5: Scenario 2: starting condition.

	H_1	H_2	H_3	H_4	H_5	H_6
Starting number of territories	4	4	4	4	0	0
Starting packs	p_1	p_2	p_3	p_4	0	0
Maximum number of core territories	2	2	2	2	2	2
Core territories winter mortality rate	0.2	0.2	0.2	0.2	0.2	0.2
Peripheral territories winter mortality rate	0.4	0.4	0.4	0.4	0.4	0.4
Pup mortality rate	0.4	0.4	0.4	0.4	0.4	0.4

5.3.2 Scenario 2: Formulation

The chance-constrained SO formulation of the Scenario 2 problem is:

$$\begin{aligned} &\min_{i \in I} \sum_{h=1}^{6} X_{hi} \\ &s.t. \; Pr\{Y_i \geq 4\} \geq 75\% \end{aligned} \tag{5.6}$$

where I is the solution space. $|I| = 28$ for CCSB-F (Eq.(3.4)), $|I| = 22$ for M-CCSB-F (Eq.(5.3)).

5.3.3 Scenario 2: Solution Space Reduction

There are 28 possible solutions for selecting 0, 1 or 2 additional territories in six habitats (Eq.(3.4)). All possible solutions are listed in Table 5.6. Since Habitats 5 and 6 have the same starting conditions – i.e. no existing wolves, identical potential core territories, and the same mortality rates – they are considered indistinguishable habitats, as defined in Subsection 5.2.1. Hence, from Eq.(5.3), when there are six habitats in total, with two of them being indistinguishable, the total number of solutions is 22. Table 5.7 details these six pairs of equivalent solutions. Note that out of the paired equivalent solutions, for simplicity, this chapter always simulates the solution that selects additional territories in Habitat 5.

Table 5.6: Scenario 2: filtered solution space.

Additional Territories	**0**	**1**	**2**
Solutions	[4, 4, 4, 4, 0, 0]	[4, 4, 4, 4, 0, 1]	[4, 4, 4, 4, 0, 2]
		[4, 4, 4, 4, 1, 0]	[4, 4, 4, 4, 1, 1]
		[4, 4, 4, 5, 0, 0]	[4, 4, 4, 5, 0, 1]
		[4, 4, 5, 4, 0, 0]	[4, 4, 5, 4, 0, 1]
		[4, 5, 4, 4, 0, 0]	[4, 5, 4, 4, 0, 1]
		[5, 4, 4, 4, 0, 0]	[5, 4, 4, 4, 0, 1]
			[4, 4, 4, 4, 2, 0]
			[4, 4, 4, 5, 1, 0]
			[4, 4, 5, 4, 1, 0]
			[4, 5, 4, 4, 1, 0]
			[5, 4, 4, 4, 1, 0]
			[4, 4, 4, 6, 0, 0]
			[4, 4, 5, 5, 0, 0]
			[4, 5, 4, 5, 0, 0]
			[5, 4, 4, 5, 0, 0]
			[4, 4, 6, 4, 0, 0]
			[4, 5, 5, 4, 0, 0]
			[5, 4, 5, 4, 0, 0]
			[4, 6, 4, 4, 0, 0]
			[5, 5, 4, 4, 0, 0]
			[6, 4, 4, 4, 0, 0]

Table 5.7: Scenario 2: pairs of equivalent solutions.

Additional Territories	**Equivalent Solutions**	
1	[4, 4, 4, 4, 1, 0]	[4, 4, 4, 4, 0, 1]
2	[4, 4, 4, 4, 2, 0]	[4, 4, 4, 4, 0, 2]
2	[4, 4, 4, 5, 1, 0]	[4, 4, 4, 5, 0, 1]
2	[4, 4, 5, 4, 1, 0]	[4, 4, 5, 4, 0, 1]
2	[4, 5, 4, 4, 1, 0]	[4, 5, 4, 4, 0, 1]
2	[5, 4, 4, 4, 1, 0]	[5, 4, 4, 4, 0, 1]

5.3.4 Scenario 2: Results

Table 5.8 presents the different solutions obtained and their selection counts, based on two sets of 100 macro-replications of CCSB-F and M-CCSB-F with a total error allowance α of 0.05 and a tolerance level δ_γ of 0.025. Other solutions not listed in the table were not selected in the macro-replications. The sample size CCSB-F required to declare a solution feasible is $n(\beta) = 9,828$, and the failure counts threshold for declaring a solution infeasible is $m_\beta(n) = 2,331$ (Eqs.(2.19) and (2.20), respectively). For M-CCSB-F, since the solutions count is reduced to 22, the sample size is $n(\beta) = 9,322$ and the failure counts threshold is $m_\beta(n) = 2,211$.

The 'N/A' in Table 5.8 means that the corresponding solutions have equivalent solutions and were temporarily removed from the solution space, making the selection count not applicable.

Table 5.8: Scenario 2: results table for CCSB-F and M-CCSB-F.

	Feasibility Tolerance Level	$\delta_\gamma = 0.025$	
Additional Territories	**Solutions**	**CCSB-F**	**M-CCSB-F**
2	[4, 4, 4, 4, 1, 1]	99	94
2	[4, 4, 4, 4, 2, 0]	32	33
2	[4, 4, 4, 4, 0, 2]	34	N/A
2	[5, 4, 4, 4, 1, 0]	3	0
2	[5, 4, 4, 4, 0, 1]	1	N/A
	Total selection count	100	98
	Ave. runs per macro-rep	247,865	182,084
	Standard deviation	890	697

Based on these results, the minimal cost for achieving a 75% probability of reaching the conservation goal in Scenario 2, given a 5% significance level and a feasibility tolerance level of 2.5%, is two additional territories.

The optimal allocation plan both CCSB-F and M-CCSB-F selected are [4, 4, 4, 4, 1, 1] (select one additional territory in Habitat 5 and one in Habitat 6), [4, 4, 4, 4, 2, 0] (two additional territories in Habitat 5), and [4, 4, 4, 4, 0, 2] (two additional territories in Habitat 6). For the solution [4, 4, 4, 4, 0, 2], which is equivalent to [4, 4, 4, 4, 2, 0], M-CCSB-F ignores it during the selection process. However, in the macro-replications where [4, 4, 4, 4, 2, 0] is selected, based on the definition of MBA, [4, 4, 4, 4, 0, 2] is also 'selected'.

The solution that M-CCSB-F does not select is [5, 4, 4, 4, 1, 0] (select one additional territory in Habitat 1 and one in Habitat 5) in any of the 100 macro-replications, while CCSB-F selected it three times. However, a Chi-squared test at a 5% significance level suggests there is insufficient statistical evidence to claim a difference in these selection counts (see Appendix D.1).

One observation regarding the similarities among optimal solutions is that all involve protecting territories in empty habitats. This behaviour suggests that protecting territories in empty habitats may increase species persistence in the general RSS problem (as explained in Section 4.4).

5.4 M-CCSB-F and CCSB-F Comparison

In addition to the two sets of 100 macro-replications for solving Scenario 2 with a δ_γ of 0.025, another two sets of 100 macro-replications with a δ_γ of 0.050 are performed. This enables a comparison of M-CCSB-F and CCSB-F performance under conditions that require less computational effort due to the higher feasibility tolerance level.

Subsection 5.4.1 presents the comparison results. Subsection 5.4.2 compares the selection accuracy (i.e. the number of correct selections made) of the two methods; and Subsection 5.4.3 compares their computational effort (i.e. the total number of runs used).

5.4.1 Results

For $\delta_\gamma = 0.050$, $n(\beta) = 2357$ and $m_\beta(n) = 527$ for CCSB-F. For M-CCSB-F, the $n(\beta)$ is 2235, and $m_\beta(n)$ is 500.

Table 5.9 highlights the results. The complete results table is in Appendix D.2 for CCSB-F and in Appendix D.3 for M-CCSB-F.

Table 5.9: CCSB-F and M-CCSB-F solutions selection counts comparison with two δ_γs.

Feasibility Tolerance	$\delta_\gamma = 0.05$		$\delta_\gamma = 0.025$	
Solutions	**CCSB-F**	**M-CCSB-F**	**CCSB-F**	**M-CCSB-F**
[4, 4, 4, 4, 1, 1]	21	28	99	94
[4, 4, 4, 4, 2, 0]	2	2	32	33
[4, 4, 4, 4, 0, 2]	10	N/A	34	N/A
[5, 4, 4, 4, 1, 0]	0	1	3	0
[5, 4, 4, 4, 0, 1]	2	N/A	1	N/A
[4, 5, 4, 4, 1, 0]	1	0	0	0
Total selection count	33	31	100	98
Ave. no. runs per macro-rep	56,218	41,284	247,865	182,084
Standard deviation	381	333	890	697

5.4.2 Selection Accuracy

i Feasibility Estimation

Before comparing the selection accuracy of the two methods, the feasibility of the solutions is estimated using 100,000 replications for each solution. Table 5.10 lists the solutions identified as feasible at a 5% significance level. These solutions are also the optimal solutions as explained in Section 3.4. The full feasibility estimation results are in Appendix D.4.

Table 5.10: Scenario 2: feasibility estimation for solutions identified as feasible.

Additional Territories	Solutions	$\Pr\{Y_i \geq 4\}$	Lower 90% CI	Upper 90% CI	$\Pr\{Y_i < 4\}$
2	[4, 4, 4, 4, 0, 2]	0.7616	0.7577	0.7655	0.2384
2	[4, 4, 4, 4, 1, 1]	0.7717	0.7679	0.7756	0.2283
2	[5, 4, 4, 4, 0, 1]	0.7540	0.7501	0.7580	0.2460
2	[5, 4, 4, 4, 1, 0]	0.7552	0.7513	0.7592	0.2448
2	[4, 4, 4, 4, 2, 0]	0.7586	0.7547	0.7626	0.2414

ii Selection Accuracy

Given a violation probability γ of 0.25, a clearly feasible solution should have a violation probability of less than 0.2 when $\delta_\gamma = 0.050$, and less than 0.225 when $\delta_\gamma = 0.025$ (see Subsection 2.3.3). From the feasibility estimation result (Table 5.10), there are five feasible solutions, none of which are clearly feasible. The solution closest to being clearly feasible is [4, 4, 4, 4, 1, 1] with a violation probability of 0.2283.

From Table 5.9, CCSB-F selects at least one of the five optimal solutions in 100 out of 100 macro-replications when $\delta_\gamma = 0.025$, and it does not select any infeasible solutions. When $\delta_\gamma = 0.050$, CCSB-F selects a solution in 33 out of 100 macro-replications. However, it also selects a solution not estimated to be feasible once ([4, 5, 4, 4, 1, 0]), which, given the overall error control rate α of 5%, is statistically expected.

Also from Table 5.9, with $\delta_\gamma = 0.025$, M-CCSB-F does not select [5, 4, 4, 4, 1, 0] as a feasible solution. Consequently, solution [5, 4, 4, 4, 0, 1] is also not identified as feasible because they are equivalent solutions. However, it correctly selects at least one optimal solution in 98 out of 100 macro-replications. With $\delta_\gamma = 0.050$, M-CCSB-F selects at least one optimal solution in 31 out of 100 macro-replications and does not select any infeasible solutions.

Based on a Chi-squared test (Appendix D.5), there is no statistical evidence at a 5% significance level to suggest a difference in the number of correct selections made by M-CCSB-F and CCSB-F for both δ_γ values of 0.025 and 0.050.

5.4.3 Computational Effort

As shown in Table 5.9, for $\delta_\gamma = 0.050$, the average number of runs per macro-replication is 56,218 for CCSB-F (standard deviation of 381), and 41,284 (standard deviation of

333) for M-CCSB-F. M-CCSB-F saves 26.56% compared to CCSB-F. For $\delta_\gamma = 0.025$, M-CCSB-F saves 26.54% compared to CCSB-F (247,865 to 182,084).

5.5 Applying MBA to Scenario 1

To assess the generalisability of MBA, it is applied to Scenario 1 (defined in Section 4.2). In Scenario 1, Habitat 1 and 3 have identical starting conditions, as do Habitats 5 and 6 (see Table 4.1). Hence, based on the definitions explained in Subsection 5.2.1, Habitat 1 is indistinguishable from Habitat 3, and Habitat 5 is indistinguishable from Habitat 6. With two groups of indistinguishable habitats, the size of the filtered solution space, determined using Eq.(5.5), is reduced from 28 (Eq.(4.1)) to 17 solutions. Table 5.11 presents the 11 pairs of equivalent solutions in Scenario 1.

Table 5.11: Scenario 1: pairs of equivalent solutions.

Additional Territories	**Equivalent Solutions**	
1	[6, 5, 5, 5, 0, 0]	[5, 5, 6, 5, 0, 0]
1	[5, 5, 5, 5, 1, 0]	[5, 5, 5, 5, 0, 1]
2	[7, 5, 5, 5, 0, 0]	[5, 5, 7, 5, 0, 0]
2	[5, 5, 5, 5, 2, 0]	[5, 5, 5, 5, 0, 2]
2	[6, 6, 5, 5, 0, 0]	[5, 6, 6, 5, 0, 0]
2	[6, 5, 5, 6, 0, 0]	[5, 5, 6, 6, 0, 0]
2	[6, 5, 5, 5, 1, 0]	[6, 5, 5, 5, 0, 1]
2	[6, 5, 5, 5, 1, 0]	[5, 5, 6, 5, 1, 0]
2	[6, 5, 5, 5, 0, 1]	[5, 5, 6, 5, 0, 1]
2	[5, 6, 5, 5, 1, 0]	[5, 6, 5, 5, 0, 1]
2	[5, 5, 5, 6, 1, 0]	[5, 5, 5, 6, 0, 1]

To estimate the computational effort reduction, the average number of runs (over 100 macro-replications) for solving Scenario 1 with CCSB-F (Appendix C.1) is used as a reference. Table 5.12 presents the average numbers of runs for the equivalent solutions.

Using this data and assuming the temporary removal of these equivalent solutions, the estimated savings percentage is calculated as follows (the total average number of runs for all 28 solutions is obtained from Appendix C.1):

$$\frac{\sum \text{Ave. runs of removed solutions}}{\sum \text{Ave. runs of all 28 solutions}} = \left(\frac{99{,}813.09}{252{,}522.15}\right) \approx 0.3953.$$

This result shows the potential computational effort savings that can be achieved by leveraging the indistinguishability of habitats in Scenario 1. Specifically, applying MBA reduces the solution space from 28 to 17, leading to an estimated computational savings of approximately 39.53%. This percentage represents a lower bound, as the temporary removal of equivalent solutions also decreases the number of runs required per remaining solution. For example, when MBA is applied to Scenario 1, the number of solutions decreases from 28 to 17. At $\delta_\gamma = 0.025$, $n(\beta)$ (Eq.(2.19)) is reduced from 9,828 to 8,782, and $m_\beta(n)$ (Eq.(2.20)) from 2,331 to 2,083. This reduction in the number of required runs would further lowers the total computational effort.

Table 5.12: Average number of runs for the equivalent solutions in Scenario 1, extracted from Appendix C.1.

Equivalent Solution	**Ave. No. Runs**
[5, 5, 6, 5, 0, 0]	8688.56
[5, 5, 5, 5, 0, 1]	9170.06
[5, 5, 7, 5, 0, 0]	8655.80
[5, 5, 5, 5, 0, 2]	9291.94
[5, 6, 6, 5, 0, 0]	9107.50
[5, 5, 6, 6, 0, 0]	8736.99
[6, 5, 5, 5, 0, 1]	9161.91
[5, 5, 6, 5, 1, 0]	9134.30
[5, 5, 6, 5, 0, 1]	9142.52
[5, 6, 5, 5, 0, 1]	9556.72
[5, 5, 5, 6, 0, 1]	9166.79
$\sum$	99,813.09

5.6 Summary

In answering research question RQ2, which asks how and to what extent computational effort in SO can be reduced by leveraging the simulation model, this chapter introduces the MBA to address the computational demands of solving the chance-constrained wolf RSS problem from a modelling perspective. Applied before CCSB-F, MBA saves computational effort by temporarily removing solutions with equivalent alternatives, which are identified based on the simulation model's characteristics. The performance of the MBA, in terms of selection accuracy and computational effort, is assessed by combining it with CCSB-F (M-CCSB-F) and comparing it to the use of CCSB-F alone.

Given the number of indistinguishable habitats, the size of the filtered solution space can be calculated analytically, as provided in Section 5.2. The analytical functions show that as the problem size increases, the scale of the reduction in the number of solutions also increases, which implies a corresponding saving in computational effort.

While the theoretical reduction in the number of solutions can be substantial, the actual computational effort savings of M-CCSB-F depend on the problem. To investigate this, a small scenario of the RSS problem with two indistinguishable habitats (Scenario 2) is created. Scenario 2 is a relatively small problem with a budget of 2 additional territories and 28 solutions.

The comparison of results from M-CCSB-F and CCSB-F in solving Scenario 2 shows that, as expected, MBA does not affect the statistical guarantee of CCSB-F. In terms of computational effort, M-CCSB-F reduces computational effort by 26.56% and 26.54% with $\delta_\gamma = 0.050$ and 0.025, respectively, compared to using CCSB-F alone. To assess the generalisability of MBA, it is applied to Scenario 1. The results in Section 5.5 show that, at the same accuracy level ($\delta_\gamma = 0.025$) and problem size, MBA achieves an estimated savings of 39.53% in Scenario 1, which has more indistinguishable habitats than Scenario 2, compared to 26.54% in Scenario 2.

Chapter 6

A Rule-Based Heuristic Approach for Solution Space Filtering

6.1 Introduction

To reduce the computational intensity of CCSB-F in solving the chance-constrained wolf RSS problem, Chapter 5 develops an approach that temporarily removes equivalent solutions without affecting the statistical guarantee of the CCSB-F procedure by utilising habitat indistinguishability informed by the simulation model.

This chapter aims to develop a different approach. The difference compared to the Chapter 5 approach lies in two aspects. First, this approach does not guarantee finding the optimal solution, as it might remove the optimal solution before applying CCSB-F. Second, this approach uses expert knowledge on ordering information (i.e. information on the superiority of habitats) to inform habitat distinguishability. Note that this chapter does not assume the expert knowledge is complete or entirely correct.

In essence, this approach uses ordering information to create solution dominance rules, then removes a set of solutions that are being dominated from the solution space to reduce the number of solutions CCSB-F needs to consider. Since the solution space filtering is based on the solution dominance rules, this approach is referred to as the rule based heuristic approach (RBHA) here.

Section 6.2 explains the RBHA in detail. To demonstrate its potential effectiveness, RBHA is used with CCSB-F to solve a hypothetical wolf RSS scenario (Scenario 3). Section 6.3 describes Scenario 3. Section 6.4 describes the solution space reduction and potential computational effort savings in Scenario 3 across various budget levels with the use of RBHA. Following this, a specific case of Scenario 3 is used to investigate the performance of the entire procedure – filtering the solution space using RBHA and subsequently selecting the optimal solution with CCSB-F (referred to as H-CCSB-F) – in comparison to the use of CCSB-F alone. The application of RBHA in this specific case, alongside its chance-constrained formulation, is described in Section 6.5. Section 6.6 presents the empirical results and an analysis of them. Section 6.7 assesses the generalisability of the MBA by applying it to Scenario 1. The summary of this chapter and its key findings are in Section 6.8.

6.2 A Rule-Based Heuristic Approach

The RBHA uses expert ordering information to develop solution dominance rules. Subsection 6.2.1 describes the assumptions underlying RBHA. Subsection 6.2.2 explains the three specific pieces of ordering information used for demonstrating how RBHA works in the context of wolf RSS. Based on this ordering information, three corresponding solution dominance rules are created. These rules are then explained in Subsection 6.2.3.

6.2.1 Definitions

Suppose a grey wolf conservation expert has ordering information (i.e. information on the superiority/inferiority) on different habitats, where the habitat superiority is defined for this chapter as follows: a habitat is considered superior to another habitat if, after having a certain number of additional territories, its probability of having the target number of viable packs by the end of planning horizon, is higher. By contrast, the habitat with a lower probability is referred to as the inferior habitat. This information is the basis of the rules for solution space filtering. The rules are referred to as the solution dominance rules in this chapter.

Based on this definition of habitat superiority, when comparing two solutions that only differ in the selection of a certain number of additional territories between superior and inferior habitats, the solution that selects the additional territories in inferior habitats is less likely to reach a target number of viable packs than the one that selects the additional territories in superior habitats. Hence, the former solution is removed from the solution space before CCSB-F. The former solution is referred to as being dominated in this chapter.

6.2.2 Three Pieces of Ordering Information

In the grey wolf RSS problem, the input parameters affecting the superiority of habitats are the number of territories currently in the habitat, the number of packs and wolves in the habitat, the number of core territories in the habitat, and the three mortality rates in the habitat (see Section 3.7). Focusing on these input parameters, three pieces of ordering information are used for demonstrating the RBHA. The first piece of ordering information focuses on the mortality rates in the habitat and is stated as:

Ordering information 1: Assuming all other things being equal, a habitat is considered superior to another if it has lower mortality rates (e.g., Nickel and Walther (2019)).

The second piece of information focuses on the number of core territories in the habitat and is stated as:

Ordering information 2: Assuming all other things being equal, habitats that have more core territories are superior (as core territories have lower mortality rates compared to peripheral territories, e.g., Treves et al. (2017)).

The third piece of ordering information focus on the pack and population densities. Pack density in this chapter is defined as the ratio of the number of packs to the number of territories, and population density is the ratio of the number of wolves to the number of territories. Accordingly, the third piece of information is stated as:

Ordering information 3: Assuming all other factors remain equal, a habitat is considered superior to another if: (a) it has both a higher population density (ratio of wolves to territories) and a higher pack density (ratio of packs to territories); or (b) if pack densities are equal, the habitat with the higher population density is considered superior (e.g., Chapron et al. (2016); Hayes and Harestad (2000)).

Note that when two packs have an identical number of wolves but differ in age and gender structure, RBHA assumes they are equivalent. This is based on the understanding that expert knowledge may not be complete. In this case, expert knowledge on the influence of both age and gender structures on the superiority of habitats with the same number of packs is assumed to be lacking, as no existing research specifically addresses how age and gender structure affect habitat superiority when expanding wolf reserves. Therefore, this influence is assumed to be unimportant in the context of this study.

6.2.3 Three Solution Dominance Rules

To aid the explanation of how solution dominance rules are defined based on the three pieces of ordering information, the study area's starting conditions, solutions, and relevant solutions' characteristics are defined in Table 6.1. The i_n in Table 6.1 is the number of additional territories in Habitat n in solution i. P_n and W_n are the number of starting packs and the total number of wolves in the habitat n, respectively.

Table 6.1: Study area starting condition definition, and relevant characteristics of solutions.

		H_1	H_2	...	H_n	
Starting number of territories	[	t_1	t_2	...	t_n	]
Maximum number of core territories	[	c_1	c_2	...	c_n	]
Core territories winter mortality rate	[	cm_1	cm_2	...	cm_n	]
Peripheral territories winter mortality rate	[	prm_1	prm_2	...	prm_n	]
Pup mortality rate	[	ppm_1	ppm_2	...	ppm_n	]
Solution i	[	$t_1 + i_1$	$t_2 + i_2$	...	$t_n + i_n$	]
Pack density	[	$P_1/(t_1 + i_1)$	$P_2/(t_2 + i_2)$	...	$P_n/(t_n + i_n)$	]
Population density	[	$W_1/(t_1 + i_1)$	$W_2/(t_2 + i_2)$	...	$W_n/(t_n + i_n)$	]

With this notation, the three solution dominance rules are defined as follows:

Solution dominance rule 1: From ***Ordering information 1***, if two habitats k, l have $cm_k < cm_l$ and $prm_k < prm_l$ and $ppm_k < ppm_l$, then Habitat k is superior to Habitat l. Consider two solutions A and B, which only differ in terms of the numbers of additional territories selected in Habitats k and l. Say solution A has A_k and A_l in Habitats k and l respectively, while solution B has B_k and B_l in Habitats k and l respectively. If $A_k > B_k$ and $A_l < B_l$ then solution A dominates solution B.

Solution dominance rule 2: From ***Ordering information 2***, if two habitats k, l have $c_k > c_l$, then Habitat k is superior to Habitat l. Consider two solutions A and B, which only differ in terms of the numbers of additional territories selected in Habitats k and l. Say solution A has A_k and A_l in Habitats k and l respectively, while solution B has B_k and B_l in Habitats k and l respectively. If $A_k > B_k$ and $A_l < B_l$ then solution A dominates solution B.

Solution dominance rule 3: From ***Ordering information 3***, if two habitats k, l have $\frac{W_k}{t_k+i_k} > \frac{W_l}{t_l+i_l}$ and $\frac{P_k}{t_k+i_k} \geq \frac{P_l}{t_l+i_l}$, then Habitat k is superior to Habitat l. Consider two solutions A and B, which only differ in terms of the numbers of additional territories selected in Habitats k and l. Say solution A has A_k and A_l in Habitats k and l respectively, while solution B has B_k and B_l in Habitats k and l respectively. If $A_k > B_k$ and $A_l < B_l$ then solution A dominates solution B.

6.3 Scenario 3: Problem Description

To demonstrate the potential effectiveness of RBHA, a hypothetical grey wolf RSS problem, namely Scenario 3, is created. Scenario 3 assumes the government has a budget for protecting some additional territories, a conservation goal of having at least 14 viable packs at the end of a five-year planning horizon, and at least a 75% chance of achieving such a goal. The government aims to find additional territories allocation plans that not only achieve this probability but also minimise the number of additional territories selected.

The Scenario 3 problem is set to have six habitats. The detailed starting conditions are presented in Table 6.2. Each starting pack in Table 6.2 is denoted by a unique label, indicating the habitat location of the pack and its size. Recall that packs are categorised as big (p_b, 11 or more wolves), medium (p_m, 5 to 10 wolves), and small (p_s, 0 to 4 wolves). For a detailed explanation, see Subsection 3.7.2. The details of individual wolves within each pack are in Table 6.3.

The values highlighted in red in Table 6.2 are the differences between two consecutive habitats. Habitat 1 is considered as the 'baseline' habitat. Compared to Habitat 1, each mortality rate is 0.1 higher in Habitat 2. Habitat 3 has two fewer core territories than Habitat 2. Habitat 4 has two medium-sized packs and one small-sized pack compared to Habitat 3, which has a big pack, a medium-sized pack, and a small-sized pack. Habitat 5, compared to Habitat 4, has only 2 starting territories, with a medium-sized pack and a small-sized pack in it. Habitat 6 has no territories and no packs in it compared to Habitat 5.

Table 6.2: Scenario 3: starting condition.

		H_1	H_2	H_3	H_4	H_5	H_6	
Starting number of territories	[	6	6	6	6	2	0	]
Starting packs	[	p_{b_1} p_{m_1} p_{s_1}	p_{b_2} p_{m_2} p_{s_2}	p_{b_3} p_{m_3} p_{s_3}	$p_{m_4^1}$ $p_{m_4^2}$ p_{s_4}	p_{m_5} p_{s_5}	0	]
Maximum number of core territories	[	4	4	2	2	2	2	]
Core territories winter mortality rate	[	0.1	0.2	0.2	0.2	0.2	0.2	]
Peripheral territories winter mortality rate	[	0.3	0.4	0.4	0.4	0.4	0.4	]
Pup mortality rate	[	0.3	0.4	0.4	0.4	0.4	0.4	]

Table 6.3: Scenario 3: packs information.

Age	**Habitat 1**			**Habitat 2**			**Habitat 3**		
	p_{b_1}	p_{m_1}	p_{s_1}	p_{b_2}	p_{m_2}	p_{s_2}	p_{b_3}	p_{m_3}	p_{s_3}
Alpha (54)	2	2	2	2	2	2	2	2	2
Adults (30)	4	2	0	2	0	0	4	2	0
Yearlings (18)	2	0	0	4	2	0	4	2	0
Pups (6)	4	2	0	4	2	0	2	0	0
Location:	Core	Core	Core	Core	Core	Core	Core	Core	Peri.
Total:	12	6	2	12	6	2	12	6	2
	Habitat 4			**Habitat 5**			**Habitat 6**		
	$p_{m_4^1}$	$p_{m_4^2}$	p_{s_4}	p_{m_5}	p_{s_5}		0		
Alpha (54)	2	2	2	2	2		0		
Adults (30)	2	1	0	1	0		0		
Yearlings (18)	2	1	0	2	0		0		
Pups (6)	2	2	0	1	0		0		
Location:	Core	Core	Peri.	Core	Core				
Total:	8	6	2	6	2		0		

6.4 Scenario 3: Solution Space Reduction

To demonstrate the scale of the reduction in the solution space when using RBHA, Subsection 6.4.1 explains three solution dominance rules in Scenario 3, and calculates the size of the solution space after applying each of those rules. To show the potential computational savings that can be made by combining the RBHA with CCSB-F (referred to as H-CCSB-F), Subsection 6.4.2 provides a comparison between the anticipated computational efforts required by H-CCSB-F versus CCSB-F in Scenario 3 for 10 different budget levels.

6.4.1 Solution Space Reduction

In Scenario 3, all mortality rates of Habitat 1 are lower than those in Habitat 2 (Table 6.2). Hence, based on ***Solution dominance rule 1***, Habitat 1 is superior to Habitat 2. This implies that: if there exist two solutions that differ only in the numbers of additional territories each selects in Habitats 1 and 2, the solution that selects more territories in Habitat 1 and fewer territories in Habitat 2 dominates the other solution.

Habitat 2 has more core territories than Habitat 3 (Table 6.2). Hence, based on ***Solution dominance rule 2***, Habitat 2 is superior to Habitat 3. This implies that: if there exist two solutions that differ only in the numbers of additional territories each selects in Habitats 2 and 3, the solution that selects more territories in Habitat 2 and fewer territories in Habitat 3 dominates the other solution.

Habitats 3 and 4 have the same pack density, but Habitat 3 has a higher population density than Habitat 4. Both Habitats 3 and 4 have 3 packs, but Habitat 3 has packs of sizes 12, 6, and 2 wolves, while Habitat 4 has packs of sizes 8, 6, and 2 wolves (Table 6.3). Hence, based on ***Solution dominance rule 3***, Habitat 3 is superior to Habitat 4. This implies that: if there exist two solutions that differ only in the numbers of additional territories selected in Habitats 3 and 4, the solution that selects more territories in Habitat 3 and fewer territories in Habitat 4 dominate the other solution.

Habitat 3 has 3 packs in 6 territories with a total of 20 wolves, Habitat 4 has 3 packs in 6 territories with a total of 16 wolves, and Habitat 5 has 2 packs with a total of 8 wolves in 2 territories (Table 6.3). This means that Habitat 5 has a higher pack density and a higher population density than each of Habitats 3 and 4. Hence, based on Solution dominance rule 3, Habitat 5 is superior to both Habitat 3 and 4. This implies that: if there exist two solutions that differ only in the numbers of additional territories selected in Habitat 5 and Habitat 3 (or 4), the solution that selects more territories in Habitat 5 and fewer territories in Habitat 3 (or 4) dominates the other solution.

Table 6.4 presents the difference in the number of solutions for Scenario 3 before and after applying the above three solution dominance rules for each of the 10 budget levels. The original solution space sizes are calculated by Eq.(3.4). The filtered solution space sizes are calculated by removing dominated solutions and counting the remaining solutions. The filtered solution space sizes are presented in the 'Filtered Solution Space' column.

Table 6.4: Scenario 3: solution space size comparison between the original solution space versus solution space after applying RBHA.

Budget Levels	Original Solution Space	Filtered Solution Space	Reduction Percentage
1	7	4	42.86%
2	28	11	60.71%
3	84	24	71.43%
4	210	47	77.62%
5	462	85	81.60%
6	924	145	84.31%
7	1716	235	86.31%
8	3003	366	87.81%
9	5005	551	88.99%
10	8008	807	89.92%

It is clear from Table 6.4 that the bigger the problem in terms of the size of the budget (and solution space), the greater the percentage reduction in the solution space RBHA makes in Scenario 3.

6.4.2 Potential Computational Effort Reduction

To demonstrate the potential computational effort savings that H-CCSB-F can make compared to CCSB-F, Table 6.5 shows the upper limit on the number of simulation runs required (in 10,000) for one macro-replication of each of H-CCSB-F and CCSB-F, and their corresponding reduction percentages for budget levels range from 1 to 10. For this comparison, the violation probability γ is set to 0.25, overall error allowance $\alpha = 0.05$, and feasibility tolerance $\delta_\gamma = 0.025$.

The values for CCSB-F's $n(\beta)$ (sample size required for declaring a solution feasible) and $m_\beta(n)$ (the failure count threshold for declaring a solution infeasible) are calculated by Eq.(2.19) and Eq.(2.20), with the original solution space sizes from Table 6.4. The $n(\beta)$ and $m_\beta(n)$ values for H-CCSB-F are calculated using the filtered solution space sizes.

The reduction percentages indicate the potential computational effort savings. These are calculated under the assumption that all solutions in the Scenario 3 problem are either clearly feasible (i.e. with $Y_i \geq 14$ for each simulation output) for the $n(\beta)$, or are clearly infeasible (i.e. with $Y_i < 14$ for each simulation output) for the $m_\beta(n)$. The actual savings H-CCSB-F achieved will depend on the problem, specifically the percentage of solutions near the feasibility boundary in the solution space, and on the characteristics of the solutions that RBHA removes.

Table 6.5 shows that the larger the budget level, the higher the potential computational savings achievable by H-CCSB-F when compared to CCSB-F in the Scenario 3 problem. This indicates H-CCSB-F's potential to reduce computational effort when solving large-scale RSS problems.

Table 6.5: Total number of runs required for one macro-replication (in 10,000), and the potential computational effort savings of H-CCSB-F.

Budget Levels	CCSB-F $n(\beta)$	H-CCSB-F $n(\beta)$	Reduction Percentage	CCSB-F $m_\beta(n)$	H-CCSB-F $m_\beta(n)$	Reduction Precentage
1	4.86	2.33	52.17%	1.15	0.55	52.18%
2	27.52	8.67	68.51%	6.53	2.06	68.50%
3	102.15	22.81	77.67%	24.23	5.41	77.67%
4	296.75	51.34	82.70%	70.39	12.18	82.70%
5	731.81	103.59	85.84%	173.62	24.57	85.85%
6	1603.14	193.33	87.94%	380.32	45.86	87.94%
7	3209.78	337.79	89.48%	761.56	80.14	89.48%
8	5986.18	561.22	90.62%	1420.12	133.15	90.62%
9	10540.03	893.89	91.52%	2500.50	212.08	91.52%
10	17694.48	1376.34	92.22%	4197.79	326.51	92.22%

6.5 Specific Case: Solution Space Reduction

Subsection 6.4.2 shows the potential computational effort savings of H-CCSB-F compared to CCSB-F in Scenario 3 under the assumption that all solutions are either clearly feasible or clearly infeasible. To demonstrate a more precise example of computational effort savings, both procedures are applied to a specific case of Scenario 3 with a budget of four additional territories. This section provides the details of the original solution space and the filtered solution space for this specific case.

To select four additional territories in six habitats, the number of possible combinations (i.e. original solution space size) is 210 (Eq.(3.4)). Table 6.6 lists all 163 possible solutions that have been removed from the original solution space for being dominated by others. These removed solutions are categorised based on the specific solution dominance rule they have violated. Table 6.7 presents the filtered solution space with 47 remaining solutions (which is the solution space for H-CCSB-F).

As detailed in Section 6.3, Scenario 3 has 14 starting wolf packs and six habitats. With a budget of four additional territories, the chance-constrained formulation is written as follows:

$$\min_{i \in I} \sum_{h=1}^{6} X_{hi}$$
$$s.t.\ Pr\{Y_i \geq 14\} \geq 75\% \tag{6.1}$$

The solution space for the CCSB-F procedure has 210 solutions. The solution space for H-CCSB-F has 47 solutions. The violation probability γ is 0.25.

6.6 Specific Case: H-CCSB-F and CCSB-F Comparison

To compare the performance of H-CCSB-F and CCSB-F in terms of their selection accuracy and computational effort for the specific case of Scenario 3, an experiment with 100 macro-replications of each procedure is performed. The overall error allowance α is set to 5%, and the feasibility tolerance level δ_γ is set to 0.025 for the experiment. Subsection 6.6.1 presents the results, Subsection 6.6.2 provides a selection accuracy comparison, and Subsection 6.6.3 provides an analysis on the computational efforts comparison.

6.6.1 Results

For 210 solutions with α of 5% and δ_γ of 0.025, CCSB-F's $n(\beta)$ is 14,131 and $m_\beta(n)$ is 3,352 (calculated from Eq.(2.19) and Eq.(2.20) respectively). For H-CCSB-F, with 47 solutions, $n(\beta) = 10,924$ and $m_\beta(n) = 2,591$. The random seeds for the experiment are detailed in the Appendix E.1. Table 6.8 presents all the solutions that are declared as feasible by CCSB-F, the number of times those solutions have been selected out of 100 macro-replications, and their corresponding selection count with H-CCSB-F. The

'N/A' entries mean that the corresponding solution has been filtered out by RBHA, and hence is not considered in H-CCSB-F.

Table 6.6: Specific case of Scenario 3: solutions that have been filtered out.

Rule 1	Rule 1 cont.	**Rule 2**	Rule 2 cont.	**Rule 3**	**Rule 3 cont.**
[6,7,6,6,2,0]	[6,7,7,7,2,1]	[6,6,7,6,2,0]	[7,6,7,7,2,1]	[6,6,6,7,2,0]	[6,6,6,7,3,0]
[6,7,6,6,2,1]	[6,8,6,7,2,1]	[6,6,7,6,2,1]	[6,6,9,6,2,1]	[6,6,6,7,2,1]	[6,6,6,7,3,1]
[6,7,6,6,3,0]	[6,7,8,6,2,1]	[6,6,7,6,3,0]	[7,6,8,6,2,1]	[6,6,6,8,2,0]	[6,6,6,7,4,0]
[6,7,6,7,2,0]	[6,8,7,6,2,1]	[6,6,7,7,2,0]	[8,6,7,6,2,1]	[7,6,6,7,2,0]	[7,6,6,7,3,0]
[6,7,7,6,2,0]	[6,9,6,6,2,1]	[6,6,8,6,2,0]	[6,6,7,6,5,0]	[6,6,6,7,2,2]	[6,6,6,7,3,2]
[6,8,6,6,2,0]	[7,8,6,6,2,1]	[7,6,7,6,2,0]	[6,6,7,7,4,0]	[6,6,6,8,2,1]	[6,6,6,7,4,1]
[6,7,6,6,2,2]	[6,7,6,6,5,0]	[6,6,7,6,2,2]	[6,6,8,6,4,0]	[7,6,6,7,2,1]	[7,6,6,7,3,1]
[6,7,6,6,3,1]	[6,7,6,7,4,0]	[6,6,7,6,3,1]	[7,6,7,6,4,0]	[6,6,6,8,3,0]	[6,6,6,7,5,0]
[6,7,6,7,2,1]	[6,7,7,6,4,0]	[6,6,7,7,2,1]	[6,6,7,8,3,0]	[6,6,6,9,2,0]	[6,6,6,8,4,0]
[6,7,7,6,2,1]	[6,8,6,6,4,0]	[6,6,8,6,2,1]	[6,6,8,7,3,0]	[7,6,6,8,2,0]	[7,6,6,7,4,0]
[6,8,6,6,2,1]	[6,7,6,8,3,0]	[7,6,7,6,2,1]	[7,6,7,7,3,0]	[7,7,6,7,2,0]	[7,7,6,7,3,0]
[6,7,6,6,4,0]	[6,7,7,7,3,0]	[6,6,7,6,4,0]	[6,6,9,6,3,0]	[8,6,6,7,2,0]	[8,6,6,7,3,0]
[6,7,6,7,3,0]	[6,8,6,7,3,0]	[6,6,7,7,3,0]	[7,6,8,6,3,0]	[7,7,7,6,2,0]	
[6,7,7,6,3,0]	[6,7,8,6,3,0]	[6,6,8,6,3,0]	[8,6,7,6,3,0]	[6,6,6,7,2,3]	
[6,8,6,6,3,0]	[6,8,7,6,3,0]	[7,6,7,6,3,0]	[6,6,7,9,2,0]	[6,6,6,8,2,2]	
[6,7,6,8,2,0]	[6,9,6,6,3,0]	[6,6,7,8,2,0]	[6,6,8,8,2,0]	[7,6,6,7,2,2]	
[6,7,7,7,2,0]	[7,8,6,6,3,0]	[6,6,8,7,2,0]	[7,6,7,8,2,0]	[6,6,6,8,3,1]	
[6,8,6,7,2,0]	[6,7,6,9,2,0]	[7,6,7,7,2,0]	[6,6,9,7,2,0]	[6,6,6,9,2,1]	
[6,7,8,6,2,0]	[6,7,7,8,2,0]	[6,6,9,6,2,0]	[7,6,8,7,2,0]	[7,6,6,8,2,1]	
[6,8,7,6,2,0]	[6,8,6,8,2,0]	[7,6,8,6,2,0]	[8,6,7,7,2,0]	[7,7,6,7,2,1]	
[6,9,6,6,2,0]	[6,7,8,7,2,0]	[8,6,7,6,2,0]	[6,6,10,6,2,0]	[8,6,6,7,2,1]	
[7,8,6,6,2,0]	[6,8,7,7,2,0]	[6,6,7,6,2,3]	[7,6,9,6,2,0]	[7,7,7,6,2,1]	
[6,7,6,6,2,3]	[6,9,6,7,2,0]	[6,6,7,6,3,2]	[7,7,8,6,2,0]	[6,6,6,9,3,0]	
[6,7,6,6,3,2]	[7,8,6,7,2,0]	[6,6,7,7,2,2]	[8,6,8,6,2,0]	[7,6,6,8,3,0]	
[6,7,6,7,2,2]	[6,7,9,6,2,0]	[6,6,8,6,2,2]	[9,6,7,6,2,0]	[6,6,6,10,2,0]	
[6,7,7,6,2,2]	[6,8,8,6,2,0]	[7,6,7,6,2,2]		[7,6,6,9,2,0]	
[6,8,6,6,2,2]	[6,9,7,6,2,0]	[6,6,7,6,4,1]		[7,7,6,8,2,0]	
[6,7,6,6,4,1]	[7,8,7,6,2,0]	[6,6,7,7,3,1]		[8,6,6,8,2,0]	
[6,7,6,7,3,1]	[6,10,6,6,2,0]	[6,6,8,6,3,1]		[7,7,7,7,2,0]	
[6,7,7,6,3,1]	[7,9,6,6,2,0]	[7,6,7,6,3,1]		[8,7,6,7,2,0]	
[6,8,6,6,3,1]		[6,6,7,8,2,1]		[9,6,6,7,2,0]	
[6,7,6,8,2,1]		[6,6,8,7,2,1]		[8,7,7,6,2,0]	

Table 6.7: Specific case of Scenario 3: filtered solution space.

Additional Territories=0	Additional Territories=1	Additional Territories=2	Additional Territories=3	Additional Territories=4
[6, 6, 6, 6, 2, 0]	[6, 6, 6, 6, 2, 1]	[6, 6, 6, 6, 2, 2]	[6, 6, 6, 6, 2, 3]	[6, 6, 6, 6, 2, 4]
	[6, 6, 6, 6, 3, 0]	[6, 6, 6, 6, 3, 1]	[6, 6, 6, 6, 3, 2]	[6, 6, 6, 6, 3, 3]
	[7, 6, 6, 6, 2, 0]	[7, 6, 6, 6, 2, 1]	[7, 6, 6, 6, 2, 2]	[7, 6, 6, 6, 2, 3]
		[6, 6, 6, 6, 4, 0]	[6, 6, 6, 6, 4, 1]	[6, 6, 6, 6, 4, 2]
		[7, 6, 6, 6, 3, 0]	[7, 6, 6, 6, 3, 1]	[7, 6, 6, 6, 3, 2]
		[7, 7, 6, 6, 2, 0]	[7, 7, 6, 6, 2, 1]	[7, 7, 6, 6, 2, 2]
		[8, 6, 6, 6, 2, 0]	[8, 6, 6, 6, 2, 1]	[8, 6, 6, 6, 2, 2]
			[6, 6, 6, 6, 5, 0]	[6, 6, 6, 6, 5, 1]
			[7, 6, 6, 6, 4, 0]	[7, 6, 6, 6, 4, 1]
			[7, 7, 6, 6, 3, 0]	[7, 7, 6, 6, 3, 1]
			[8, 6, 6, 6, 3, 0]	[8, 6, 6, 6, 3, 1]
			[8, 7, 6, 6, 2, 0]	[8, 7, 6, 6, 2, 1]
			[9, 6, 6, 6, 2, 0]	[9, 6, 6, 6, 2, 1]
				[6, 6, 6, 6, 6, 0]
				[7, 6, 6, 6, 5, 0]
				[7, 7, 6, 6, 4, 0]
				[8, 6, 6, 6, 4, 0]
				[7, 7, 7, 6, 3, 0]
				[8, 7, 6, 6, 3, 0]
				[9, 6, 6, 6, 3, 0]
				[8, 8, 6, 6, 2, 0]
				[9, 7, 6, 6, 2, 0]
				[10, 6, 6, 6, 2, 0]

Table 6.8: Specific case of Scenario 3: results table for CCSB-F and H-CCSB-F.

Cost	Solutions	CCSB-F Selection Count	CCSB-F Selection Percentage	H-CCSB-F Selection Count	H-CCSB-F Selection Percentage
3	[8, 6, 6, 6, 2, 1]	16	16%	33	33%
3	[9, 6, 6, 6, 2, 0]	60	60%	68	68%
4	[6, 6, 6, 6, 4, 2]	1	3%	0	0%
4	[7, 6, 6, 6, 3, 2]	35	100%	24	100%
4	[7, 6, 6, 7, 2, 2]	3	9%	N/A	N/A
4	[7, 6, 7, 6, 2, 2]	29	83%	N/A	N/A
4	[7, 7, 6, 6, 2, 2]	35	100%	23	96%
4	[8, 6, 6, 6, 2, 2]	35	100%	24	100%
4	[7, 6, 6, 6, 4, 1]	35	100%	24	100%
4	[7, 6, 7, 6, 3, 1]	26	74%	N/A	N/A
4	[7, 7, 6, 6, 3, 1]	34	97%	22	92%
4	[8, 6, 6, 6, 3, 1]	35	100%	24	100%
4	[8, 6, 6, 7, 2, 1]	35	100%	N/A	N/A
4	[8, 6, 7, 6, 2, 1]	35	100%	N/A	N/A
4	[8, 7, 6, 6, 2, 1]	35	100%	24	100%
4	[9, 6, 6, 6, 2, 1]	35	100%	24	100%
4	[8, 6, 6, 6, 4, 0]	35	100%	24	100%
4	[8, 6, 6, 7, 3, 0]	19	54%	N/A	N/A
4	[8, 6, 7, 6, 3, 0]	35	100%	N/A	N/A
4	[8, 7, 6, 6, 3, 0]	35	100%	24	100%
4	[9, 6, 6, 6, 3, 0]	35	100%	24	100%
4	[9, 6, 6, 7, 2, 0]	35	100%	N/A	N/A
4	[8, 7, 7, 6, 2, 0]	2	6%	N/A	N/A
4	[9, 6, 7, 6, 2, 0]	35	100%	N/A	N/A
4	[9, 7, 6, 6, 2, 0]	35	100%	24	100%
4	[10, 6, 6, 6, 2, 0]	35	100%	24	100%
Selection count for a cost of 3 territories		65	65%	76	76%
Selection count for a cost $\leq$ 4 territories		100	100%	100	100%

As noted in Section 3.4, CCSB-F stops its process once a feasible solution is identified and after it has checked all designs that use the same amount of territories. For the specific case of Scenario 3, in 65 out of 100 macro-replications, CCSB-F terminates after checking all solutions with a cost of three additional territories (i.e. from solution 1 to

84). It checks the entire solution space (i.e. from solution 1 to 210) for the remaining 35 macro-replications. Hence, the 'Selection Percentage' column in Table 6.8 represents the proportion of times a solution is selected corresponding to its cost.

Similarly, in 76 out of 100 macro-replications, H-CCSB-F terminates before checking solutions that use four additional territories (i.e. from solution 1 to 24). It checks the entire solution space (i.e. from solution 1 to 47) for the remaining 24 macro-replications.

Both CCSB-F and H-CCSB-F results show that the minimal cost for the Scenario 3 problem is three additional territories. Two different plans can achieve the conservation goal of having at least 14 viable packs at the end of the 5-year planning horizon with at least 75% probability. One is to select two additional territories in Habitat 1 and one additional territory in Habitat 6 (solution [8, 6, 6, 6, 2, 1]). The other is to select 3 additional territories in Habitat 1 (solution [9, 6, 6, 6, 2, 0]).

6.6.2 Selection Accuracy

i Feasibility Estimation

To estimate the true feasibility of solutions, 10,000 simulations are run on each solution and a t-test with a 5% significance level is performed. From these runs, 17 of them are identified as feasible (Table 6.9). Among the 193 solutions identified as infeasible in the initial experiment, 34 of them had an upper 90% confidence interval (Bonferroni corrected) of over 75%, i.e. solutions which have an estimated violation probability $\hat{p}_i$ such that:

$$\hat{p}_i - t_{1-0.05/210,10000-1} \cdot \sqrt{\frac{\hat{p}_i(1-\hat{p}_i)}{10000-1}} \leq 0.25$$

To clarify the feasibility of these solutions, a further examination with 100,000 simulations on each of them was performed to get a more accurate estimation. From the result of 34 x 100k runs, a further 11 solutions are identified as feasible, and 23 solutions are infeasible. Of the 23 solutions considered infeasible, 8 of them had an upper confidence level greater than 75%. These solutions underwent an additional 500,000 runs. Only one solution is identified as feasible. The remaining 7 solutions are identified as infeasible.

Table 6.9 shows the 29 feasible solutions and their 90% confidence levels. The remaining 181 solutions are infeasible based on the feasibility estimation result, and their details are in Appendix E.2.

ii Feasible but Dominated Solutions

Of the 29 feasible solutions identified, 11 (marked in red in Table 6.9) are not in the H-CCSB-F solution space, as they have been removed by the RBHA prior to the CCSB-F procedure. Table 6.10 details the estimated probability of $Y_i \geq 14$ for these dominated solutions and the specific solution dominance rule each violates. Additionally, this table presents the corresponding dominating solutions, alongside the dominating solutions' estimated probabilities and their lower 90% confidence limits.

iii Selection Accuracy Comparison

From the feasibility estimation result, the optimal solutions (i.e. feasible solutions that use the minimal cost) are [8, 6, 6, 6, 2, 1] and [9, 6, 6, 6, 2, 0], both cost three additional territories. As expected, CCSB-F and H-CCSB-F do not select any infeasible solutions in 100 macro-replications.

Based on 100 macro-replications, the probabilities of selecting any of the optimal solutions for CCSB-F and H-CCSB-F are 65% and 76% respectively (Table 6.8). There is not enough statistical evidence at a 5% significance level to say that there is a difference between the probability of correct selection of CCSB-F and H-CCSB-F based on a Chi-squared test (Appendix E.3).

Table 6.9: Specific case of Scenario 3: feasibility estimation for solutions identified as feasible.

	Solutions	**Pr**$\{Y_i \geq 14\}$	**Lower 90% CI**	**Upper 90% CI**
Feasible after 10k runs	[10, 6, 6, 6, 2, 0]	0.8162	0.8027	0.8297
	[9, 6, 6, 6, 3, 0]	0.81	0.7963	0.8237
	[9, 6, 6, 6, 2, 1]	0.8081	0.7943	0.8219
	[8, 6, 6, 6, 2, 2]	0.8017	0.7878	0.8156
	[8, 6, 6, 6, 3, 1]	0.7977	0.7837	0.8117
	[8, 7, 6, 6, 2, 1]	0.7868	0.7725	0.8011
	[9, 7, 6, 6, 2, 0]	0.7867	0.7724	0.8010
	[8, 6, 6, 6, 4, 0]	0.7856	0.7713	0.7999
	[7, 6, 6, 6, 3, 2]	0.7815	0.7671	0.7959
	[8, 7, 6, 6, 3, 0]	0.7801	0.7656	0.7946
	[9, 6, 7, 6, 2, 0]	0.7775	0.7630	0.7920
	[8, 6, 6, 7, 2, 1]	0.7758	0.7612	0.7904
	[8, 6, 7, 6, 2, 1]	0.7754	0.7608	0.7900
	[7, 6, 6, 6, 4, 1]	0.7731	0.7585	0.7877
	[9, 6, 6, 7, 2, 0]	0.7729	0.7583	0.7875
	[8, 6, 7, 6, 3, 0]	0.769	0.7543	0.7837
	[7, 7, 6, 6, 3, 1]	0.7674	0.7526	0.7822
Feasible after 100k runs	[7, 6, 6, 7, 2, 2]	0.7597	0.7557	0.7637
	[9, 6, 6, 6, 2, 0]	0.7629	0.7589	0.7669
	[8, 6, 6, 7, 3, 0]	0.76441	0.7604	0.7684
	[7, 6, 7, 6, 2, 2]	0.76667	0.7627	0.7706
	[8, 6, 6, 6, 2, 1]	0.76011	0.7561	0.7641
	[6, 6, 6, 6, 4, 2]	0.75575	0.7517	0.7598
	[7, 6, 7, 6, 3, 1]	0.76324	0.7592	0.7672
	[7, 7, 6, 6, 2, 2]	0.77241	0.7685	0.7764
	[8, 7, 7, 6, 2, 0]	0.75945	0.7554	0.7635
	[7, 6, 6, 7, 3, 1]	0.75693	0.7529	0.7610
	[7, 6, 6, 6, 2, 3]	0.75684	0.7528	0.7609
Feasible after 500k runs	[8, 6, 6, 6, 3, 0]	0.751588	0.7501	0.7531

Table 6.10: Estimated $\Pr\{Y_i \geq 14\}$ of dominated solutions and their corresponding dominating solutions.

	Dominated Solutions	**Pr** $\{Y_i \geq 14\}$	**Dominating Solutions**	**Pr** $\{Y_i \geq 14\}$	**Lower 90% CI**
Violate *Solution dominance rule 2*	[9, 6, 7, 6, 2, 0]	0.7775	[9, 7, 6, 6, 2, 0]	0.7867	0.7724
	[8, 6, 7, 6, 2, 1]	0.7754	[8, 7, 6, 6, 2, 1]	0.7868	0.7725
	[8, 6, 7, 6, 3, 0]	0.769	[8, 7, 6, 6, 3, 0]	0.7801	0.7656
	[7, 6, 7, 6, 2, 2]	0.76667	[7, 7, 6, 6, 2, 2]	0.77241	0.7456
	[7, 6, 7, 6, 3, 1]	0.76324	[7, 7, 6, 6, 3, 1]	0.7674	0.7526
Violate *Solution dominance rule 3*	[8, 7, 7, 6, 2, 0]	0.75945	[8, 7, 6, 6, 3, 0]	0.7801	0.7543
	[8, 6, 6, 7, 2, 1]	0.7758	[8, 6, 7, 6, 2, 1]	0.7754	0.7608
	[9, 6, 6, 7, 2, 0]	0.7729	[9, 6, 7, 6, 2, 0]	0.7775	0.7630
	[7, 6, 6, 7, 2, 2]	0.7597	[7, 6, 7, 6, 2, 2]	0.76667	0.7478
Violate *rule 4*	[8, 6, 6, 7, 3, 0]	0.76441	[8, 6, 7, 6, 3, 0]	0.769	0.7543
	[7, 6, 6, 7, 3, 1]	0.75693	[7, 6, 7, 6, 3, 1]	0.76324	0.7458

6.6.3 Computational Effort

Table 6.11 shows the average number of runs for a single macro-replication of the CCSB-F and H-CCSB-F, the standard deviation of those averages, and the scale of the computational effort saving made by H-CCSB-F.

Table 6.11: Specific case of Scenario 3: computational effort (simulation runs in million) comparison.

	CCSB-F	**H-CCSB-F**	**Reduction Percentage**
Average (macro-replications stops at cost=3)	0.8550	0.2071	75.77%
Standard Deviation	0.0014	0.0007	
Average (macro-replications stops at cost=4)	2.3363	0.4469	80.87%
Standard Deviation	0.0020	0.0009	

Table 6.11 shows there is a difference in the average savings of the number of runs

between macro-replications that terminate at a cost of three additional territories versus those that terminate at a cost of four additional territories. As discussed in Subsection 6.4.2, the more solutions RBHA removes, the more computational effort H-CCSB-F saves. In the specific case of Scenario 3, out of 210 solutions, solutions 1 to 84 use a cost of at most three additional territories, and RBHA removes 60 of them (71.43%). Solutions 85 to 210 use all four territories, and RBHA removes 103 of these (81.75%). Therefore, when comparing macro-replications that stop at a cost of four additional territories, it is expected to have a higher reduction percentage in the total number of runs compared to those that terminate at a cost of three additional territories.

6.7 Applying RBHA to Scenario 1

To assess the generalisability of RBHA, it is applied to Scenario 1 (defined in Section 4.2). In Scenario 1, Habitat 2 has a higher pack and a higher population density than Habitats 1, 3, and 4. Habitat 4 has a higher pack and a higher population density than Habitats 1 and 3. Based on *Solution dominance rule 3* (Subsection 6.2.3), if two solutions differ only in the numbers of additional territories each selects in Habitats 2 and 1 (or 3, or 4), then the solution that selects more territories in Habitat 2 and fewer in Habitats 1 (or 3, or 4) dominates the other solution. Similarly, if two solutions differ only in the number of additional territories each selects in Habitats 4 and 1 (or 3), then the solution that selects more territories in Habitat 4 and fewer in Habitats 1 (or 3) dominates the other solution.

Table 6.12 lists all 17 solutions removed due to being dominated by other solutions when applying the heuristic approach to Scenario 1, along with their corresponding average number of runs, extracted from Appendix C.1.

Table 6.12: Solutions removed when applying RBHA to Scenario 1.

Removed as less in Habitat 2 (15)	**Ave. No. Runs**	**Removed as less in Habitat 4 (2)**	**Ave. No. Runs**
[5, 5, 5, 6, 0, 0]	8727.33	[5, 6, 6, 5, 0, 0]	9107.50
[5, 5, 6, 5, 0, 0]	8688.56	[6, 6, 5, 5, 0, 0]	9116.25
[6, 5, 5, 5, 0, 0]	8687.92		
[5, 5, 5, 6, 0, 1]	9166.79		
[5, 5, 6, 5, 0, 1]	9142.52		
[6, 5, 5, 5, 0, 1]	9161.91		
[5, 5, 5, 6, 1, 0]	9165.15		
[5, 5, 6, 5, 1, 0]	9134.30		
[6, 5, 5, 5, 1, 0]	9143.57		
[5, 5, 5, 7, 0, 0]	8329.18		
[5, 5, 6, 6, 0, 0]	8736.99		
[6, 5, 5, 6, 0, 0]	8726.64		
[5, 5, 7, 5, 0, 0]	8655.80		
[6, 5, 6, 5, 0, 0]	8699.26		
[7, 5, 5, 5, 0, 0]	8633.37		
	$\sum$ Ave. runs	151,023.04	

The optimal solutions (Section 4.4) are not removed by the heuristic approach when applied to Scenario 1. The lower bound of the estimated computational effort savings, calculated using the average run data from Appendix C.1, is given by:

$$\left(\frac{151{,}023.04}{252{,}522.15}\right) \times 100\% \approx 59.81\%.$$

This result shows the potential computational effort savings achievable when applying RBHA to Scenario 1. Specifically, RBHA removes 17 dominated solutions, resulting in an estimated computational savings of 59.81%. This percentage represents a lower bound, as removing dominated solutions also reduces the number of runs required for the remaining solutions. For example, when RBHA is applied to Scenario 1, the number of solutions decreases from 28 to 11. At $\delta_\gamma = 0.025$, $n(\beta)$ (Eq.(2.19)) is reduced from

9,828 to 7,878, and $m_\beta(n)$ (Eq.(2.20)) from 2,331 to 1,869. This reduction in the number of required runs would further decrease the total computational effort.

6.8 Summary

In addressing **RQ3: How and to what extent can expert opinion be used to reduce the computational effort in SO?**, Chapter 6 introduces an RBHA that aims to reduce the computational intensity issue faced by CCSB-F in solving the chance-constrained RSS problem. The RBHA uses expert knowledge on the ordering of habitats to create solution dominance rules, then removes solutions that are being dominated from the solution space to reduce the total simulation runs CCSB-F requires. The combination of RBHA and CCSB-F is referred to as H-CCSB-F.

An example RSS problem (Scenario 3) is used to demonstrate the potential computational effort savings of H-CCSB-F in comparison to using CCSB-F alone. The results in Section 6.4 show that the potential savings made by H-CCSB-F are substantial. They also suggest that the savings increase not only in absolute terms but also in percentage terms as the size of the solution space increases.

To investigate the actual impact of H-CCSB-F, it is compared with CCSB-F in solving a specific case of Scenario 3 described in Section 6.5. The results in Section 6.6 demonstrate that H-CCSB-F, while maintaining statistically equivalent selection accuracy, reduces computational effort by an average of 80.73% compared to CCSB-F across 100 macro-replications.

When RBHA is applied to Scenario 1, Section 6.7 shows that the estimated savings (59.81%) is lower than when applied to Scenario 3 (80.73%). One possible reason for this lower savings is that only one rule applies to Scenario 1, whereas all three rules apply to Scenario 3.

Although RBHA does not remove any optimal solutions in both examples, there is no guarantee that optimal solutions will never be removed. However, selected solutions are guaranteed to be feasible and the best among those in the filtered solution space (under the given feasibility tolerance level and significance level).

Based on the feasibility estimation results, RBHA does remove feasible but not optimal solutions in the specific case of Scenario 3. Hence, in contexts where a comprehensive identification of all feasible solutions is required, the use of RBHA should be carefully considered. However, such a scenario is not the focus of this research, which examines the RSS problem with the aim of identifying optimal solutions.

Chapter 7

Conclusions and Further Work

This thesis investigates the application of SO to the RSS problem, with a particular emphasis on solution methods with a statistical guarantee. SO enables the identification of the statistically guaranteed optimal set of reserve sites while accounting for inherent stochasticity in species survival. To the best of the author's knowledge, this is the first successful attempt to solve an RSS problem with SO that provides a statistical guarantee. Additionally, contributions are made in developing two approaches to reduce computational intensity in solving the RSS problem.

This chapter concludes the thesis by reflecting on the contributions made to the wildlife RSS problem and SO. Section 7.1 summarises these contributions. Section 7.2 identifies areas for further work.

7.1 Contributions

7.1.1 Solving the RSS Problem

The first contribution of this thesis, presented in Chapters 3 and 4, is solving the RSS problem with an SO method that has a statistical guarantee. This thesis focuses particularly on the chance-constrained RSS problem. A scenario of the problem, informed by wolf situations in California (Chapter 4), is solved using an established procedure designed for solving chance-constrained SO problems (CCSB-F). As expected, the result shows that solving the chance-constrained RSS problem with a statistical guarantee can be computationally intense (defined in terms of the number of simulation runs).

In theory, the computational intensity depends not only on the characteristics of the solution space (i.e. how close the solutions are to the feasibility boundary), but also on the selection accuracy one wants to achieve. As detailed in Chapter 2 and in Hong, Luo, and Nelson (2015), with the indifference zone approach that CCSB-F uses, for a solution space with solutions close to the feasibility boundary, i.e. with $\Pr\{Y < N\} \in (\gamma - \delta_\gamma, \gamma]$, the computational effort required is likely to be higher than for a solution space mainly consisting of infeasible solutions. Additionally, the computational intensity depends on the required selection accuracy. A higher selection accuracy requires a smaller δ_γ which in turn increases the computational effort. Chapter 4 provides an example illustrating the possible scale of this effect through Scenario 1, where most solutions are close to the feasibility boundary.

Moreover, CCSB-F is expected to be less computationally intense than a standard hypothesis testing procedure. Chapter 4 uses Scenario 1 to illustrate the extent of this computational intensity reduction in solving the RSS problem. In theory, the scale of the computational effort savings depends on the characteristics of the solution space, and CCSB-F will tend to save more computational effort in problems where the solution space mainly consists of infeasible solutions far from the feasibility boundary.

7.1.2 Computational Intensity Reduction Approaches

The second contribution is the development of two approaches, presented in Chapters 5 and 6, that are designed to address the computational intensity of the SO method highlighted in Chapter 4. Both approaches reduce the computational intensity by filtering the solution space prior to the use of CCSB-F.

The model-based approach detailed in Chapter 5 is built on the characteristics of the simulation model. Specifically, it works based on the understanding that when two habitats have equal parameter values, they become indistinguishable within the model. Thus, solutions differing only in the locations of additional territories among these indistinguishable habitats are considered identical. Only one of these solutions requires simulation, while the rest can be temporarily removed from the solution space. CCSB-F is then applied to the filtered solution space. Since all solutions are theoretically checked, and those temporarily removed solutions are also considered at the end of the procedure, the statistical guarantee still holds for the entire solution space. The advantage of this combined method compared to the use of CCSB-F alone is that it keeps the statistical guarantee while reducing the computational intensity of CCSB-F.

Given the number of indistinguishable habitats, the size of the filtered solution space can be calculated with established formulae. As the problem size and the number of indistinguishable habitats increases, the scale of the reduction in the number of solutions also increases, implying a corresponding reduction in computational intensity. However, since CCSB-F is a fixed-precision procedure and the simulation output is a random variable, the actual computational effort required is expected to depend on the characteristics of the solution space. In the Scenario 2 problem with 28 solutions, where most solutions are close to the feasibility boundary, the observed computational saving is approximately 26%.

The second solution space filtering approach discussed in Chapter 6 is a rule-based heuristic approach. This approach assumes that experts provide some information on habitat superiority. Such information is then used to create solution dominance rules. Solutions that are dominated by others are then removed from the solution space. Then, CCSB-F is applied to the filtered solution space. Since this approach is based on expert opinion and this thesis does not assume expert opinion is entirely accurate, optimal solutions may be removed during the filtering process. Therefore, this approach does not guarantee finding the optimal solution for the entire solution space. However, since CCSB-F is used in the filtered solution space, the statistical guarantee of finding the optimal solution in the filtered solution space, and the solution found being feasible, still holds.

The strength of the heuristic approach is that it reduces the computational effort drastically. Although an analytical function is not available to calculate the size of the filtered solution space in this case, the number of solutions can be enumerated. As the Scenario 3 example demonstrates, the reduction in the number of solutions increases with the problem size. This suggests that the reduction in computational effort required is likely to also increase with the problem size. However, in theory, this reduction is dependent on the characteristics of the solution space. In a specific case of Scenario 3 with 210 solutions, a substantial computational effort saving of around 80% is observed.

The key difference between the model-based approach and the heuristic approach is that the former does not remove solutions from the solution space, while the latter does. This results in a difference in the statistical guarantees held by the two approaches. In terms of computational effort, for the same solution space, the heuristic approach is likely to filter out more solutions and thus save more computational effort than the model-based approach. However, this comes at the cost of losing the statistical guarantee for the entire solution space.

A key limitation of the model-based approach is that it assumes the existence of indistinguishable habitats. If all habitats are distinguishable, it cannot be applied. For the heuristic approach, it depends on expert opinion, which may not always be available. Additionally, some solution dominance rules may be problem-specific and not generalisable to a wider range of problems—for example, *Solution Dominance Rule 1* and *Solution Dominance Rule 2* do not apply under the starting conditions of Scenario 1.

7.2 Further Work

7.2.1 Improvement on CCSB-F

As detailed in Chapter 2, the CCSB-F works by sequentially collecting samples for each solution. The feasibility of a solution is determined by collecting up to $n(\beta)$ samples. A solution is declared infeasible whenever the failure count threshold is reached. A solution is declared feasible if, upon collecting $n(\beta)$ samples, the failure count threshold has not been reached. However, for some 'highly feasible' solutions, it may not be necessary to collect the full $n(\beta)$ samples to declare them feasible. The computational efficiency of the CCSB-F could be improved by reducing the number of samples required for these solutions. One possible approach is to incorporate a feasibility measurement into the CCSB-F. Eckman, Henderson, and Shashaani (2023) introduced three potential methods for measuring feasibility: the feasibility score, the likelihood ratio score, and the posterior probability of feasibility. Exploring the computational effort of CCSB-F given some feasibility measurements could contribute to the field of SO.

7.2.2 Solution space filtering approaches

Despite that this thesis focuses particularly on the species persistence (estimated by the number of viable wolf packs at the end of the planning horizon) as a secondary performance measure (constraint), other measures, such as the location of the packs, number of pups, or the compactness of the sites can be easily incorporated into the formulation. The same applies to the primary performance measure (objective), this thesis focuses on the number of sites in the objective, but other measures such as the individual costs of each territory or habitat, can be incorporated easily by amending the deterministic objective function. As such changes do not change the structure of the formulation, the main conclusions of this thesis are still applicable. However, if the objective function becomes stochastic and requires simulation to evaluate, the effectiveness of the two approaches in Chapters 5 and 6 requires further investigation.

7.2.3 Feasibility tolerance level

Throughout this thesis, the importance of the feasibility tolerance level δ_γ is highlighted multiple times. In theory, this value should be chosen by the decision maker based on the smallest difference in solution performance that would have a practical impact. However, choosing this value is not straightforward in practice. One existing approach in providing a robust method for selecting δ_γ is from Lee, Park, and Park (2018). They present a fully sequential RnS framework for self-adjusting δ_γ in stochastic constrained SO problems. Their high-level idea involves running two RnS procedures simultaneously and iteratively, adjusting δ_γ at the end of each iteration until both procedures yield the same set of feasible solutions. Since this method involves multiple iterations of two RnS procedures, the computational intensity problem faced by CCSB-F is likely to be a more severe problem in their framework. Therefore, exploring a more robust and less computationally intense method for selecting δ_γ could contribute to the application of SO to RSS.

7.2.4 Other ways of reducing computational effort

One of the established ways of reducing the computational intensity is to use meta-models (an analytical approximation between the simulation output and input). An existing attempt of using a meta-model based approach to solve stochastically constrained SO problems is Tsai, Park, and Chang (2023). The high-level idea is to repetitively fit meta-models, find the optimal solution by solving the meta-model based optimisation problem in each iteration, and then add this solution to a set of promising solutions if it is feasible. The final optimal solution is obtained from the set of promising solutions using a RnS procedure. Evaluating the performance of this method in solving the RSS problem will allow for an assessment of the effectiveness of the two approaches developed in this thesis. Moreover, given the special structure of the solution space in the RSS problem, a more tailored meta-model could be developed.

7.2.5 Future Work on the Wolf Model

The simulation model described in Chapter 3 provides a baseline for modelling the grey wolf population. However, several additional aspects could be incorporated into the model to further test the performance of SO in a more realistic conservation context. One such aspect is environmental uncertainty. This could be incorporated into the model by introducing random environmental events, such as wildfires during summer or extreme cold weather during winter, in different habitats. Another consideration is spatial aspects. As discussed in Subsection 3.5.2, the current model does not explicitly account for spatial factors. One way to address this is by incorporating non-uniform dispersal probabilities. Currently, the model assumes that dispersers have an equal chance of moving into any other habitat, regardless of distance or habitat quality. This may not accurately reflect actual dispersal behaviour. Incorporating a non-uniform dispersal probabilities matrix that reflects habitat proximity, connectivity, or quality would be a straightforward way to include spatial aspects in the model.

7.3 Final Comment

This research aims to understand the extent to which SO can be used in RSS. In addressing this question, it proposes two principles for reducing computational effort in using SO to solve RSS problems. The first principle is to explore the details of the simulation model and the structure of the solution space, which may help develop problem-specific methods for filtering the solution space. The second principle is to use expert knowledge if available, as this can guide the creation of solution dominance rules to filter the solution space.

Appendix A

Appendix: Chapter 2

A.1 Systematic Literature Search

A.1.1 Objectives

Because there is no recent (up to 2019) review of how simulation and optimisation have been applied to the RSS problem, this literature search aims to provide an overview of the scope and structure of existing literature and to identify the most relevant studies in the broader field of reserve design.

A.1.2 Search Strategy

Search terms were divided into two dimensions: the application area and the method used. A scoping search was initially conducted in Scopus to identify key studies and the thesaurus of the literature on reserve design problems. Method-wise, since this thesis aims to identify the use of SO in RSS, the focus is on simulation and optimisation (heuristics and stochastic programming are included in the search terms in case papers do not mention optimisation).

The following search strings were used: “Reserve Design*” OR “Reserve Site Select*” OR “Reserve selection” AND Simulation OR Optimi?ation OR “Stochastic Program*” OR Heuristic*. These terms were searched in the following bibliographic databases:

- EBSCOhost (EBSCOhost, 2019), including:
  - Academic Search Ultimate: multi-disciplinary bibliographic database.
  - Business Source Complete: includes scholarly business journals.
  - EconLit: source of references to economic literature.
  - GreenFILE: offers well-researched information covering all aspects of human impact on the environment.
  - MathSciNet via EBSCOhost: mathematical bibliographic database.
  - OpenDissertations: access to past dissertations.
- Scopus (Scopus, 2019)

The search terms were searched in any field using EBSCOhost, and in the title, abstract, and keywords using Scopus on the 8th of October, 2019.

EBSCOhost initially identified 274 results, including 254 academic journals, 7 reviews, 2 conference papers, 2 dissertations, and 2 working papers. Scopus returned 301 results, including 257 articles, 27 conference papers, 13 reviews, and 4 other materials. These results were exported to Endnote X9, and after both automatic and manual deduplication, 324 references remained.

A.1.3 Apply Exclusion Criteria

Abstracts of 324 references were examined manually, and exclusion criteria were applied. Since the aim of the search was to provide an overview of existing literature on SO methods applied to RSS problems, only studies that focus on using either simulation, optimisation, or other relevant OR methods to solve RSS problems, or that introduce criteria or decision rules for reserve design to use with simulation or optimisation methods, were included. Hence, articles were excluded if they were either not available in English, not related to wildlife conservation, or did not use either simulation or optimisation. Fifty-four studies were removed after applying the exclusion criteria.

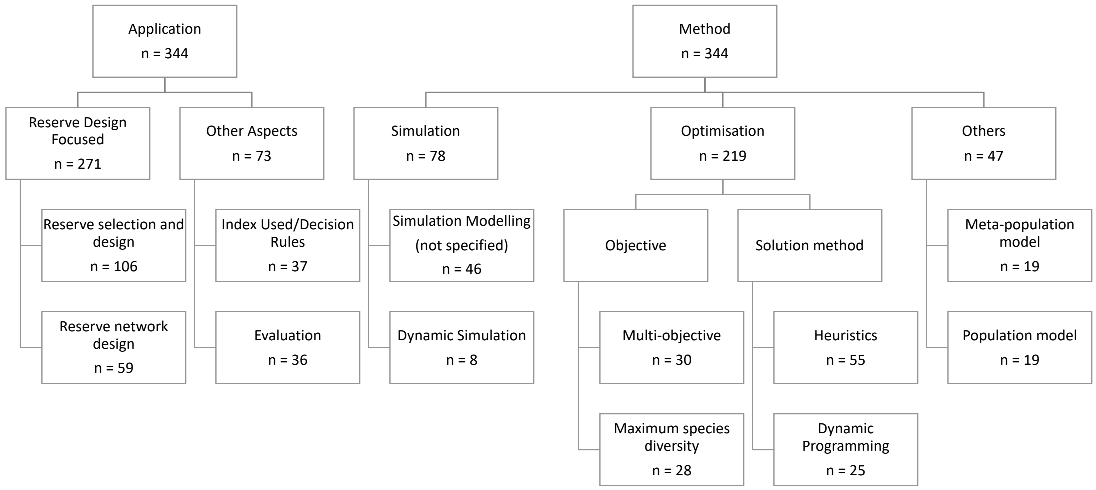
A.1.4 Reference Lists Checking

While reviewing the search results, 15 review papers and one SO paper were identified. The references of the review papers and the citation list of the SO paper were checked, leading to the addition of 74 relevant papers, resulting in a total of 344 papers.

A.1.5 Classification

These 344 studies were then classified into different categories. Most studies were classified twice: once based on the method applied and once based on the application area. Each of these categories contains sub-categorises. The following graph presents the categorisation, but due to the large number of sub-categories in each categorise, only the most popular sub-categorises (with more than 5 papers on the subject) are presented.

Figure A.1: Classification result



A.1.6 Results

Based on the search results, mathematical optimisation is the most popular technique compared to simulation in the field of reserve design. Within optimisation papers, most are formulated as static models rather than stochastic models, and many have multiple objectives. Heuristics is the most popular solution method applied. Other formulations include, but are not limited to, mixed integer linear programming, and nonlinear programming. Other solution methods include metaheuristics and dynamic programming.

Papers that used simulation techniques usually did not specify which simulation technique they used in their abstracts. Most used simulation as a tool to test possible management plans, understand ecosystem interactions, or evaluate current reserves. Some simulation models are also used with optimisation to either determine parameter values or evaluate results from optimisation.

Application-wise, most of the papers focus on RSS. Others discuss the evaluation of reserve design or the search for new and better decision rules or decision indexes. These are more focused on the ecological side of the problem and hence have been separated from the main reserve design category.

A.2 Literature Search - SO in RSS

Search date: 2024 May the 14th.

Database searched:

- Scopus: search within the article title, abstract and keywords.
- EBSCOhost: search within the article title, abstract and keywords. The databases searched are:
  - Academic Search Ultimate
  - Applied Science & Technology Full Text (H.W. Wilson)
  - Business Source Alumni Edition
  - Business Source Complete
  - eBook Collection (EBSCOhost)
  - EconLit
  - Environment Complete
  - GeoRef
  - GreenFILE

- MathSciNet via EBSCOhost
- OpenDissertations

Search terms: "site* select*" OR "reserve* design*" AND "simulation optimi?ation" OR "optimi?ation via simulation".

Out of 12 resulting papers from Scopus, apart from Haight and Travis (2008), none addresses SO in RSS. Out of 5 resulting papers from EBSCOhost, apart from Haight and Travis (2008), none addresses SO in RSS.

All 20 papers from Scopus that cited Haight and Travis (2008) were checked, and none of them addresses SO in RSS.

Appendix B

Appendix: Chapter 3

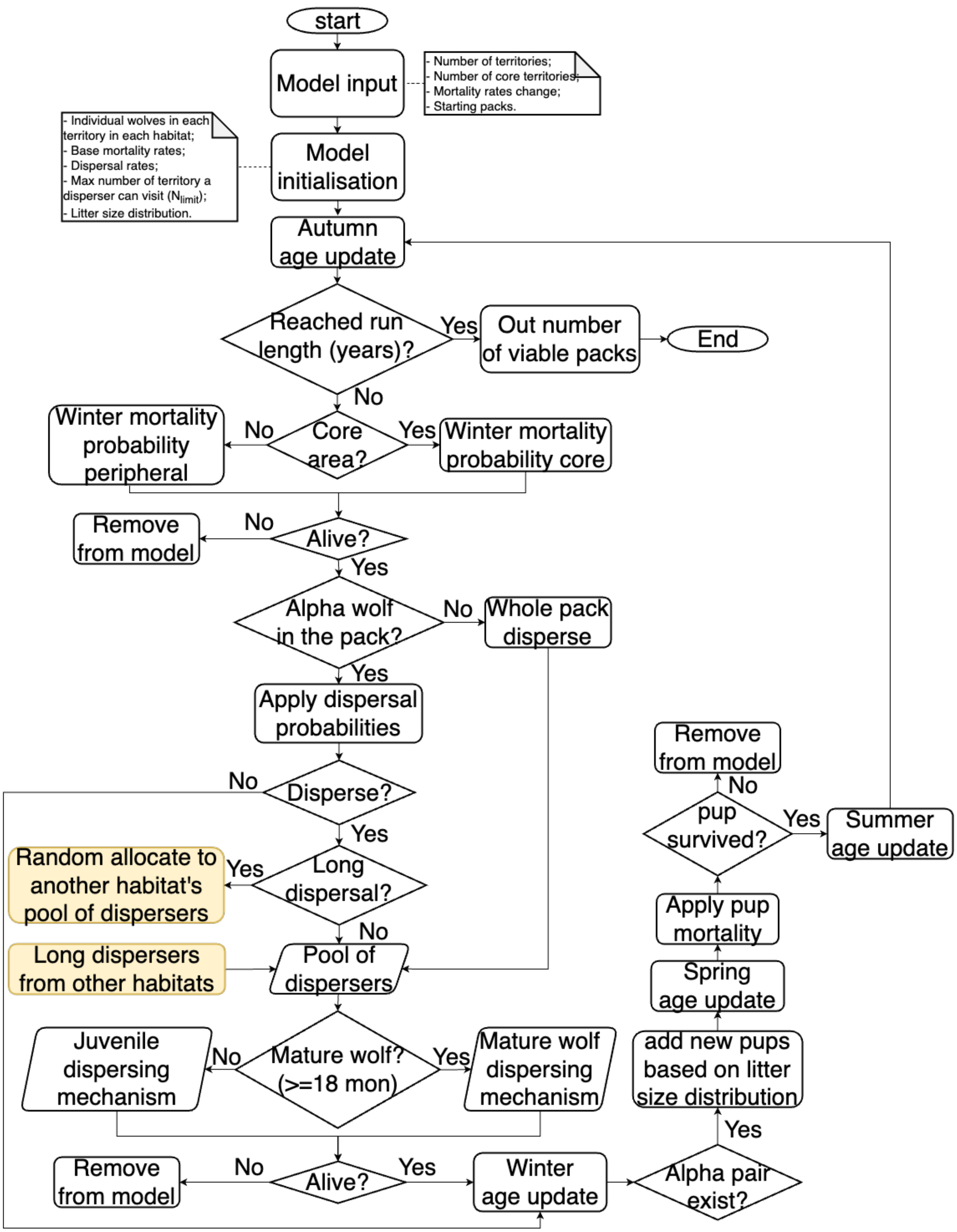
B.1 Computer Model Run Logic

B.1.1 Flowchart for Single Habitat in Single Replication

Figure B.1 describes the logic of the computer model based on a single habitat and a single replication. The processes within each habitat are identical, and the habitats are linked together by the long dispersers.

This method of coding the conceptual model simplifies the simulation compared to modelling every single individual activity at every time step. Such simplification is achieved by grouping wolves that are in the same stage and modelling their group activities at corresponding time steps.

Figure B.1: Flowchart for single habitat in single replication.



B.1.2 Model Execution Flowchart

The above logic is coded in Python using functional programming. The code structure for multiple replications is summarised in Figure B.2. The execution of the functions for single replication is summarised in Figure B.3.

Figure B.2: Flowchart for multiple replications.

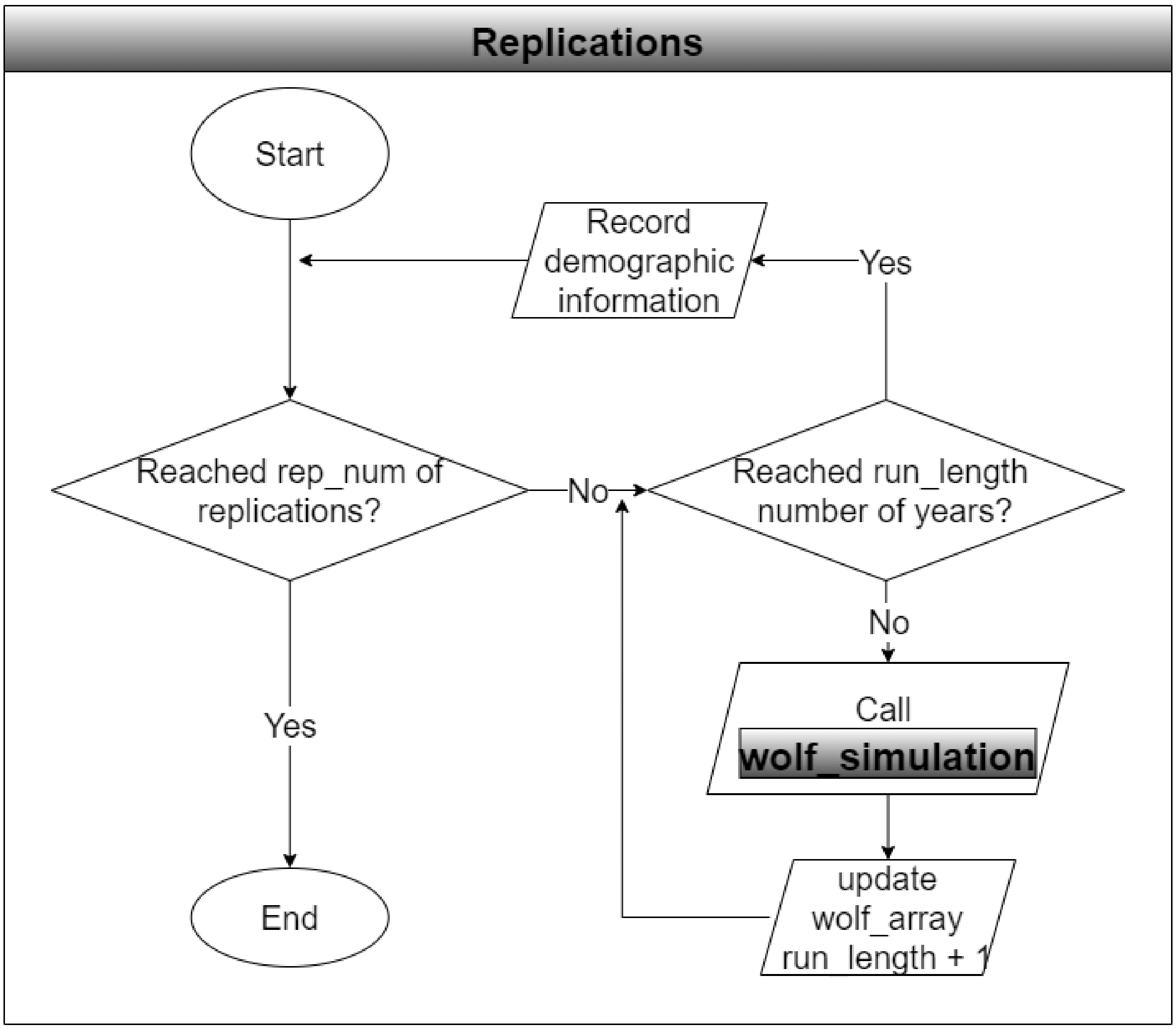
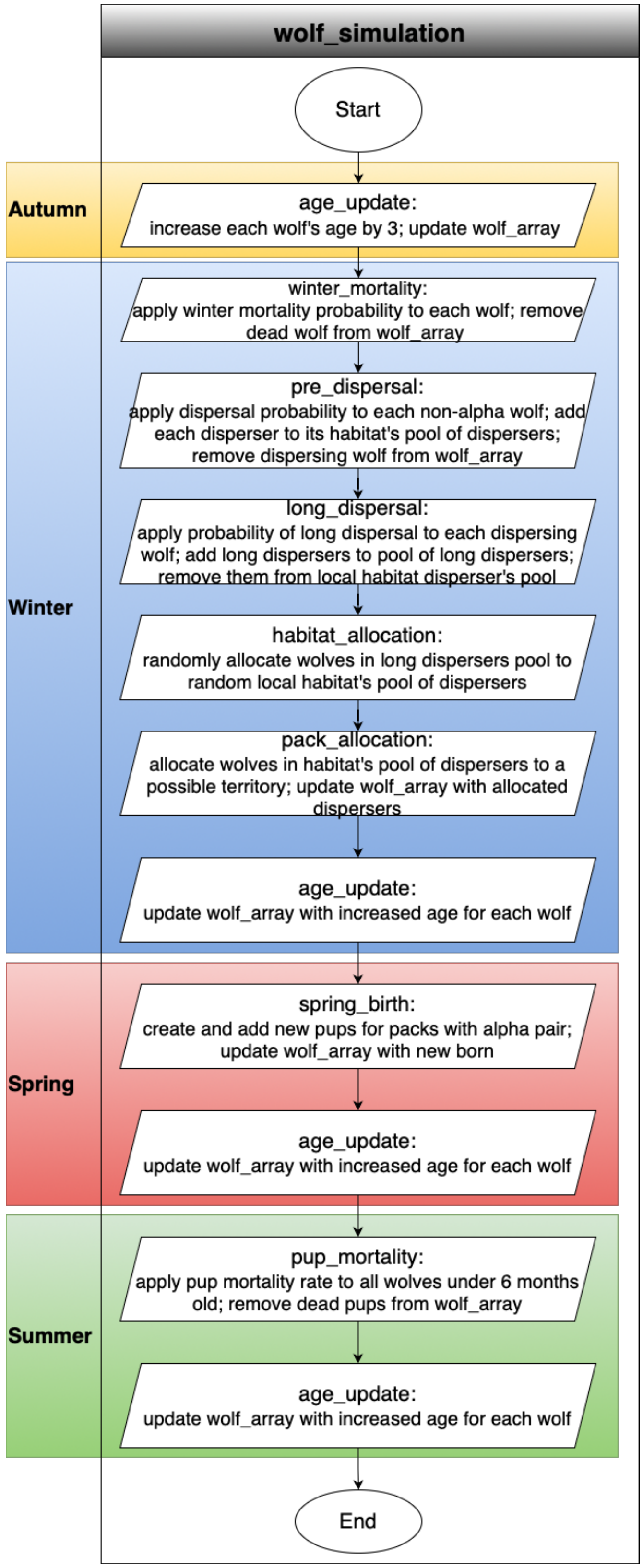


Figure B.3: Flowchart for a single simulation run.



B.2 Assumptions Summary Table

Table B1 summarises the assumptions the model made.

Table B1: Summary table for assumptions.

Real World	**Assumptions**
Pack size varies from 3 to 11 in general	No limit
In rare cases, a pack can have more than one pair of wolves producing offspring	Only 1 alpha pair per pack
Rare cases of more than one female giving birth within a pack	Only 1 litter per pack per year
In general, the earliest reproduction age is two years, with the youngest recorded at 10 months and the oldest at five years	The minimal mating age is 21 months for both genders
Breeding longevity is four to five years in the wild	No upper limit on breeding longevity
Some packs may allow young, typically male, dispersers to remain in the pack for a short time	Dispersers only stay in another pack if they find a mate or if the territory is empty
Younger wolves travel further than older wolves	There is no age or gender difference in dispersal distance and success rate
The size of territories changes, but the territory core remains the same	The sizes of territories are uniform and do not change

B.3 Technical Simplifications

Table B2 summarises the simplifications made to enhance computational efficiency.

Table B2: Technical design simplification table.

Real World	**Technical Design Simplification**
Litter size average is six pups, ranging from 1 to 13 pups	Litter size average is 6, ranging from 0-12 pups, with equal probability
Most pups disperse from their natal pack between 9 and 36 months	The youngest disperser is 9 months, increasing the probability of dispersal for each age group
Wolves disperse throughout the year	Wolves only disperse during winter
If a pup is orphaned, it can be reared by other pack members	Not applicable. By the time of the dispersal process, wolves will be at least 9 months old
Pairs form during dispersal and find suitable land to establish a pack	If a disperser settles in an empty territory, another suitable disperser mate can find them later and pair up
Wolves die all year round	Accumulate adult mortality to winter
Wolves between 6 and 12 months die primarily because of malnutrition in winter	Accumulate all pup mortality to summer

B.4 Model Verification

Based on Balci (1994) framework for validation, verification, and testing, consistency checking and desk checking are performed with the formulated problem, the system and objective definition, and the conceptual model. For the programmed model verification, bottom-up testing, debugging, execution monitoring, execution tracing, and black box testing are performed.

Consistency checking is done by ensuring that the language, parameters, notations, and graph representations do not contain contradictions and are used consistently. Desk checking is undertaken by thoroughly examining the work to ensure correctness, completeness, consistency, and clarity.

Bottom-up testing is completed by testing each function from the minimal functional level to the whole model level, ensuring each function acts as expected. Debugging is executed by identifying and fixing possible bugs in the code. Execution monitoring and execution tracing are carried out by monitoring and tracing the running of the code step by step. Black box testing is performed by inputting extreme values for which the output can be predicted and checking if the output aligns with the expectations.

Appendix C

Appendix: Chapter 4

C.1 Results Table for Scenario 1: 100 Macro-replications of CCSB-F

To speed up the experiments, some macro-replications were run in parallel. To ensure the independence of each replication, different random number seeds were used for parallel macro-replications. The random number seed is generated randomly with $random.randrange(0, 2^{32} - 1)$ function in Python ($2^{32} - 1$ is the maximum seed number the *random* package can take). The first 10,000 random numbers produced by this function were stored. And those random numbers were used sequentially as the random number seed for each experiment. Table C1 documents the starting random number seed for each parallel macro-replication in Scenario 1.

Table C1: Scenario 1: random number seed settings for 100 macro-replications of CCSB-F.

Random Number Index	Random Number Seed	Problem	macro-replications Index
19	36772	Scenario 1 CCSB-F $\delta_\gamma = 0.025$	1-20
20	2847821818	Scenario 1 CCSB-F $\delta_\gamma = 0.025$	21-40
21	2830182769	Scenario 1 CCSB-F $\delta_\gamma = 0.025$	41-60
22	3089602841	Scenario 1 CCSB-F $\delta_\gamma = 0.025$	61-80
23	224252415	Scenario 1 CCSB-F $\delta_\gamma = 0.025$	81-100
29	507094555	Scenario 1 CCSB-F $\delta_\gamma = 0.01$	1-20
30	1034920211	Scenario 1 CCSB-F $\delta_\gamma = 0.01$	21-40
31	1366554720	Scenario 1 CCSB-F $\delta_\gamma = 0.01$	41-60
32	4009340797	Scenario 1 CCSB-F $\delta_\gamma = 0.01$	61-80
33	3988537505	Scenario 1 CCSB-F $\delta_\gamma = 0.01$	81-100

The experiments were run on Lenovo with Inter(R) Core(TM) i5-6500T CPU@2.50GHz and Mac mini with Apple M1, macOS 13.4 (22F66) computers. The average run time per macro-replications of CCSB-F for $\delta_\gamma = 0.025$ is 29.8288 seconds with a standard deviation of 1.6563 seconds. For $\delta_\gamma = 0.01$, the average time is 6.9660 hours, with a standard deviation of 0.3825 hours. Table C2 presents the results for 100 macro-replications of CCSB-F in solving Scenario 1. The average total number of runs for all 28 solutions is 252,522.15.

Table C2: Scenario 1: CCSB-F results table (100 macro-replications).

	Tolerance Level	$\delta_\gamma = 0.025$		$\delta_\gamma = 0.01$	
Additional Territories	Solutions	Selection Count	Average Number of Runs	Selection Count	Average Number of Runs
0	[5, 5, 5, 5, 0, 0]	0	8705.71	0	57477.41
1	[5, 5, 5, 5, 0, 1]	0	9170.06	0	60350.71
1	[5, 5, 5, 5, 1, 0]	0	9134.73	0	60332.75
1	[5, 5, 5, 6, 0, 0]	0	8727.33	0	57481.44
1	[5, 5, 6, 5, 0, 0]	0	8688.56	0	57385.1
1	[5, 6, 5, 5, 0, 0]	0	9097.84	0	59978.15
1	[6, 5, 5, 5, 0, 0]	0	8687.92	0	57363.21
2	[5, 5, 5, 5, 0, 2]	0	9291.94	0	61515.59
2	[5, 5, 5, 5, 1, 1]	20	9633.27	100	62805
2	[5, 5, 5, 6, 0, 1]	0	9166.79	0	60508.56
2	[5, 5, 6, 5, 0, 1]	0	9142.52	0	60347.34
2	[5, 6, 5, 5, 0, 1]	5	9556.72	74	62741.94
2	[6, 5, 5, 5, 0, 1]	0	9161.91	0	60284.48
2	[5, 5, 5, 5, 2, 0]	0	9331.23	0	61475.55
2	[5, 5, 5, 6, 1, 0]	0	9165.15	0	60506.54
2	[5, 5, 6, 5, 1, 0]	0	9134.3	0	60309.91
2	[5, 6, 5, 5, 1, 0]	8	9558.23	71	62697.91
2	[6, 5, 5, 5, 1, 0]	0	9143.57	0	60279.17
2	[5, 5, 5, 7, 0, 0]	0	8329.18	0	55032.73
2	[5, 5, 6, 6, 0, 0]	0	8736.99	0	57459.76
2	[5, 6, 5, 6, 0, 0]	0	9087.16	0	60130.6
2	[6, 5, 5, 6, 0, 0]	0	8726.64	0	57510.41
2	[5, 5, 7, 5, 0, 0]	0	8655.8	0	57003.53
2	[5, 6, 6, 5, 0, 0]	0	9107.5	0	59899.88
2	[6, 5, 6, 5, 0, 0]	0	8699.26	0	57412.63
2	[5, 7, 5, 5, 0, 0]	0	8932.22	0	58965.63
2	[6, 6, 5, 5, 0, 0]	0	9116.25	0	59889.57
2	[7, 5, 5, 5, 0, 0]	0	8633.37	0	56891.31

C.2 Feasibility Estimation Results Table for Scenario 1

Each solution is run 100,000 times. The significance level is set to 5%, and the critical value from student t distribution is $t_{1-0.05/28,99999} = 2.913795$. A solution is declared feasible when the test statistics (calculated based on the sample values) are greater than the critical value. The random number starting seed is 36772.

Table C3: Scenario 1: feasibility estimation results.

Additional Territories	Solutions	Pr $\{Y_i \geq 3\}$	Test Statistics	Lower 90% CI	Upper 90% CI	Feasibility	Pr $\{Y_i < 3\}$
0	[5,5,5,5,0,0]	0.73293	-12.2008	0.7289	0.7370	Infeasible	0.2671
1	[5,5,5,5,0,1]	0.74526	-3.4401	0.7412	0.7493	Infeasible	0.2547
1	[5,5,5,5,1,0]	0.74453	-3.9662	0.7405	0.7485	Infeasible	0.2555
1	[5,5,5,6,0,0]	0.73504	-10.7198	0.7310	0.7391	Infeasible	0.2650
1	[5,5,6,5,0,0]	0.73224	-12.6836	0.7282	0.7363	Infeasible	0.2678
1	[5,6,5,5,0,0]	0.74452	-3.9734	0.7405	0.7485	Infeasible	0.2555
1	[6,5,5,5,0,0]	0.73084	-13.6609	0.7268	0.7349	Infeasible	0.2692
2	[5,5,5,5,0,2]	0.75218	1.5967	0.7482	0.7562	Infeasible	0.2478
2	[5,5,5,5,1,1]	0.75999	7.3969	0.7561	0.7639	Feasible	0.2400
2	[5,5,5,6,0,1]	0.74499	-3.6348	0.7410	0.7490	Infeasible	0.2550
2	[5,5,6,5,0,1]	0.74585	-3.0142	0.7418	0.7499	Infeasible	0.2542
2	[5,6,5,5,0,1]	0.75818	6.0412	0.7542	0.7621	Feasible	0.2418
2	[6,5,5,5,0,1]	0.74256	-5.3811	0.7385	0.7466	Infeasible	0.2574
2	[5,5,5,5,2,0]	0.751	0.7313	0.7470	0.7550	Infeasible	0.2490
2	[5,5,5,6,1,0]	0.74262	-5.3381	0.7386	0.7466	Infeasible	0.2574
2	[5,5,6,5,1,0]	0.74523	-3.4618	0.7412	0.7492	Infeasible	0.2548
2	[5,6,5,5,1,0]	0.75483	3.5505	0.7509	0.7588	Feasible	0.2452
2	[6,5,5,5,1,0]	0.74652	-2.5298	0.7425	0.7505	Infeasible	0.2535
2	[5,5,5,7,0,0]	0.72026	-20.9517	0.7161	0.7244	Infeasible	0.2798
2	[5,5,6,6,0,0]	0.73546	-10.4241	0.7314	0.7395	Infeasible	0.2645
2	[5,6,5,6,0,0]	0.74066	-6.7391	0.7366	0.7447	Infeasible	0.2593
2	[6,5,5,6,0,0]	0.7307	-13.7585	0.7266	0.7348	Infeasible	0.2693
2	[5,5,7,5,0,0]	0.72824	-15.4678	0.7241	0.7323	Infeasible	0.2718
2	[5,6,6,5,0,0]	0.74225	-5.6031	0.7382	0.7463	Infeasible	0.2578
2	[6,5,6,5,0,0]	0.73102	-13.5354	0.7269	0.7351	Infeasible	0.2690
2	[5,7,5,5,0,0]	0.73568	-10.2691	0.7316	0.7397	Infeasible	0.2643
2	[6,6,5,5,0,0]	0.74307	-5.0155	0.7390	0.7471	Infeasible	0.2569
2	[7,5,5,5,0,0]	0.73281	-12.2849	0.7287	0.7369	Infeasible	0.2672

There are 3 feasible solutions, [5, 5, 5, 5, 1, 1], [5, 6, 5, 5, 0, 1] and [5, 6, 5, 5, 1, 0] (highlighted in grey in Table C3).

C.3 Results Table for Scenario 1: 100 Macro-replications of the Benchmark

Table C4: Scenario 1: benchmark results table (100 macro-replications).

	Tolerance Level	$\delta_\gamma = 0.025$	$\delta_\gamma = 0.01$
Additional Territories	Solutions	Selection Count	Selection Count
0	[5, 5, 5, 5, 0, 0]	0	0
1	[5, 5, 5, 5, 0, 1]	0	0
1	[5, 5, 5, 5, 1, 0]	0	0
1	[5, 5, 5, 6, 0, 0]	0	0
1	[5, 5, 6, 5, 0, 0]	0	0
1	[5, 6, 5, 5, 0, 0]	0	0
1	[6, 5, 5, 5, 0, 0]	0	0
2	[5, 5, 5, 5, 0, 2]	0	0
2	[5, 5, 5, 5, 1, 1]	19	98
2	[5, 5, 5, 6, 0, 1]	0	0
2	[5, 5, 6, 5, 0, 1]	0	0
2	[5, 6, 5, 5, 0, 1]	7	78
2	[6, 5, 5, 5, 0, 1]	0	0
2	[5, 5, 5, 5, 2, 0]	0	0
2	[5, 5, 5, 6, 1, 0]	0	0
2	[5, 5, 6, 5, 1, 0]	0	0
2	[5, 6, 5, 5, 1, 0]	12	74
2	[6, 5, 5, 5, 1, 0]	0	0
2	[5, 5, 5, 7, 0, 0]	0	0
2	[5, 5, 6, 6, 0, 0]	0	0
2	[5, 6, 5, 6, 0, 0]	0	0
2	[6, 5, 5, 6, 0, 0]	0	0
2	[5, 5, 7, 5, 0, 0]	0	0
2	[5, 6, 6, 5, 0, 0]	0	0
2	[6, 5, 6, 5, 0, 0]	0	0
2	[5, 7, 5, 5, 0, 0]	0	0
2	[6, 6, 5, 5, 0, 0]	0	0
2	[7, 5, 5, 5, 0, 0]	0	0

All experiments were run on a Lenovo with Intel(R) Core(TM) i5-6500T CPU @ 2.50GHz and a Mac mini with Apple M1, macOS 13.4 (22F66). Some of the experiments were run in parallel. The parallel runs and starting random number seeds are detailed in Table C5.

Table C5: Scenario 1: random number seed settings for 100 macro-replications of the benchmark.

Random Number Index	Random Number Seed	Problem	macro-replications Index
19	36772	Scenario 1 Benchmark $\delta_\gamma = 0.025$	1-100
24	3809321646	Scenario 1 Benchmark $\delta_\gamma = 0.01$	1-20
25	2683149395	Scenario 1 Benchmark $\delta_\gamma = 0.01$	21-40
26	2807176977	Scenario 1 Benchmark $\delta_\gamma = 0.01$	41-60
27	2145564691	Scenario 1 Benchmark $\delta_\gamma = 0.01$	61-80
28	329900673	Scenario 1 Benchmark $\delta_\gamma = 0.01$	81-100

C.4 Chi-squared Test: Selection Accuracy Comparison

For $\delta_\gamma = 0.025$:

H_0: There are equal numbers of correct selections for the benchmark and CCSB

H_1: There are unequal numbers of correct selections for the benchmark and CCSB

Solutions	**Hypothesised Proportion**	**Benchmark**	**CCSB**	**Expected**	χ^2	**Critical Value**
[5, 6, 5, 5, 1, 0]	0.5	12	8	10	0.8	3.8415

Since the χ^2 is less than the critical value with an α of 5%, there is no statistical evidence at a 5% significance level to support that there is a difference in the number of correct selections made by the benchmark and CCSB for solution $[5, 6, 5, 5, 1, 0]$.

For $\delta_\gamma = 0.010$:

H$_0$: There are equal numbers of correct selections for the benchmark and CCSB

H$_1$: There are unequal numbers of correct selections for the benchmark and CCSB

Solutions	**Hypothesised Proportion**	**Benchmark**	**CCSB**	**Expected**	χ^2	**Critical Value**
[5, 6, 5, 5, 0, 1]	0.5	78	74	76	0.8	3.8415

Since the χ^2 is less than the critical value with an α of 5%, there is no statistical evidence at a 5% significance level to support that there is a difference in the number of correct selections made by the benchmark and CCSB for solution $[5, 6, 5, 5, 0, 1]$.

Appendix D

Appendix: Chapter 5

D.1 Chi-squared Test: Results Comparison

H_0: There are equal numbers of selections for M-CCSB-F and CCSB-F

H_1: There are unequal numbers of selections for M-CCSB-F and CCSB-F

δ_γ	**Hypothesised Proportion**	**CCSB-F**	**M-CCSB-F**	**Expected**	χ^2	**Critical Value**
0.025	0.5	3	0	1.5	3	3.8415

With an α of 5%, there is no statistical evidence to support that there is a difference in the number of selections made by M-CCSB-F and CCSB-F.

D.2 Results Tables for Scenario 2: 100 Macro-replications of CCSB-F

Table D1: Scenario 2: CCSB-F results table (100 macro-replications).

	Tolerance Level	$\delta_\gamma = 0.05$		$\delta_\gamma = 0.025$	
Additional Territories	Solutions	Selection Count	Average Number of Runs	Selection Count	Average Number of Runs
0	[4, 4, 4, 4, 0, 0]	0	1906.71	0	8381.16
1	[4, 4, 4, 4, 0, 1]	0	2072.24	0	9185.65
1	[4, 4, 4, 4, 1, 0]	0	2080.36	0	9183.83
1	[4, 4, 4, 5, 0, 0]	0	1884.29	0	8319.05
1	[4, 4, 5, 4, 0, 0]	0	1887.58	0	8337.65
1	[4, 5, 4, 4, 0, 0]	0	1920.58	0	8470.42
1	[5, 4, 4, 4, 0, 0]	0	1939.99	0	8630.63
2	[4, 4, 4, 4, 0, 2]	10	2215.7	34	9699.25
2	[4, 4, 4, 4, 1, 1]	21	2283.19	99	9827.26
2	[4, 4, 4, 5, 0, 1]	0	2060.42	0	9105.09
2	[4, 4, 5, 4, 0, 1]	0	2062.22	0	9113.16
2	[4, 5, 4, 4, 0, 1]	0	2109.83	0	9251.48
2	[5, 4, 4, 4, 0, 1]	2	2141.74	1	9450.66
2	[4, 4, 4, 4, 2, 0]	2	2200.14	32	9716.8
2	[4, 4, 4, 5, 1, 0]	0	2057.99	0	9092.15
2	[4, 4, 5, 4, 1, 0]	0	2046.08	0	9095.38
2	[4, 5, 4, 4, 1, 0]	1	2099.59	0	9297.87
2	[5, 4, 4, 4, 1, 0]	0	2150.26	3	9427.53
2	[4, 4, 4, 6, 0, 0]	0	1864.22	0	8232.09
2	[4, 4, 5, 5, 0, 0]	0	1860.07	0	8261.43
2	[4, 5, 4, 5, 0, 0]	0	1905.67	0	8380.68
2	[5, 4, 4, 5, 0, 0]	0	1932.85	0	8562.84
2	[4, 4, 6, 4, 0, 0]	0	1859.48	0	8215.59
2	[4, 5, 5, 4, 0, 0]	0	1905.06	0	8409.25
2	[5, 4, 5, 4, 0, 0]	0	1945.36	0	8566.18
2	[4, 6, 4, 4, 0, 0]	0	1906.06	0	8362.78
2	[5, 5, 4, 4, 0, 0]	0	1984.87	0	8689.08
2	[6, 4, 4, 4, 0, 0]	0	1935.56	0	8600.17

Table D2: Scenario 2: random number seed settings for 100 macro-replications of CCSB-F.

Random Number Index	Random Number Seed	Problem	macro-replications Index
0	1623527968	Scenario 2 CCSB-F $\delta_\gamma = 0.05$	1-100
2	3631723266	Scenario 2 CCSB-F $\delta_\gamma = 0.025$	1-25
3	248154644	Scenario 2 CCSB-F $\delta_\gamma = 0.025$	26-50
4	2000033036	Scenario 2 CCSB-F $\delta_\gamma = 0.025$	51-75
5	3839190294	Scenario 2 CCSB-F $\delta_\gamma = 0.025$	76-100

D.3 Results Tables for Scenario 2: 100 Macro-replications of M-CCSB-F

Table D3: Scenario 2: random number seed settings for 100 macro-replications of M-CCSB-F.

Random Number Index	Random Number Seed	Problem	macro-replications Index
1	3836883120	Scenario 2 M-CCSB-F $\delta_\gamma = 0.05$	1-100
6	615377162	Scenario 2 M-CCSB-F $\delta_\gamma = 0.025$	1-25
7	2659974823	Scenario 2 M-CCSB-F $\delta_\gamma = 0.025$	26-50
8	1283674829	Scenario 2 M-CCSB-F $\delta_\gamma = 0.025$	51-75
9	3574523394	Scenario 2 M-CCSB-F $\delta_\gamma = 0.025$	76-100

Table D4: Scenario 2: CCSB-F results table (100 macro-replications).

	Tolerance Level	$\delta_\gamma = 0.05$		$\delta_\gamma = 0.025$	
Additional Territories	Solutions	Selection Count	Average Number of Runs	Selection Count	Average Number of Runs
0	[4, 4, 4, 4, 0, 0]	0	1803.08	0	7951.01
1	[4, 4, 4, 4, 1, 0]	0	1966.04	0	8719.94
1	[4, 4, 4, 5, 0, 0]	0	1787.44	0	7876.79
1	[4, 4, 5, 4, 0, 0]	0	1782.74	0	7872.78
1	[4, 5, 4, 4, 0, 0]	0	1809.1	0	8024.36
1	[5, 4, 4, 4, 0, 0]	0	1849.93	0	8167.79
2	[4, 4, 4, 4, 1, 1]	28	2176.12	94	9315.26
2	[4, 4, 4, 4, 2, 0]	2	2097.96	33	9200.04
2	[4, 4, 4, 5, 1, 0]	0	1944.27	0	8615.12
2	[4, 4, 5, 4, 1, 0]	0	1946.83	0	8623.95
2	[4, 5, 4, 4, 1, 0]	0	1985.15	0	8800.1
2	[5, 4, 4, 4, 1, 0]	1	2020.91	0	8991.3
2	[4, 4, 4, 6, 0, 0]	0	1775.14	0	7816.52
2	[4, 4, 5, 5, 0, 0]	0	1774.09	0	7827.15
2	[4, 5, 4, 5, 0, 0]	0	1798.54	0	7963.48
2	[5, 4, 4, 5, 0, 0]	0	1825.98	0	8110.05
2	[4, 4, 6, 4, 0, 0]	0	1770.32	0	7799.17
2	[4, 5, 5, 4, 0, 0]	0	1806.12	0	7952.78
2	[5, 4, 5, 4, 0, 0]	0	1835.81	0	8108.13
2	[4, 6, 4, 4, 0, 0]	0	1800.15	0	7934.85
2	[5, 5, 4, 4, 0, 0]	0	1874.61	0	8279.83
2	[6, 4, 4, 4, 0, 0]	0	1854.15	0	8133.7

D.4 Feasibility Estimation Results Table for Scenario 2

Each solution is run 100,000 times. The significance level is set to 5%, and the critical value from student t distribution is $t_{1-0.05/28,99999} = 2.913795$. A solution is declared feasible when the test statistics (calculated based on the sample values) are greater than the critical value. The random number seed used is 1258189053. For the extra 100,000 replications for solution [4, 5, 4, 4, 1, 0], the random number seed used is 2948196755.

Table D5: Scenario 2: feasibility estimation results.

Additional Territories	Solutions	$\Pr\{Y_i \geq 4\}$	Test Statistics	Upper 90% CI	Lower 90% CI	Feasibility
0	[4, 4, 4, 4, 0, 0]	0.7195	-21.4625	0.7236	0.7154	Infeasible
1	[4, 4, 4, 4, 0, 1]	0.7454	-3.3391	0.7494	0.7414	Infeasible
1	[4, 4, 4, 4, 1, 0]	0.7450	-3.6348	0.7490	0.7410	Infeasible
1	[4, 4, 4, 5, 0, 0]	0.7170	-23.1394	0.7212	0.7129	Infeasible
1	[4, 4, 5, 4, 0, 0]	0.7195	-21.4557	0.7237	0.7154	Infeasible
1	[4, 5, 4, 4, 0, 0]	0.7266	-16.6023	0.7307	0.7225	Infeasible
1	[5, 4, 4, 4, 0, 0]	0.7302	-14.1136	0.7343	0.7261	Infeasible
2	[4, 4, 4, 4, 0, 2]	0.7616	8.6088	0.7655	0.7577	Feasible
2	[4, 4, 4, 4, 1, 1]	0.7717	16.3720	0.7756	0.7679	Feasible
2	[4, 4, 4, 5, 0, 1]	0.7420	-5.7820	0.7460	0.7380	Infeasible
2	[4, 4, 5, 4, 0, 1]	0.7432	-4.9078	0.7472	0.7392	Infeasible
2	[4, 5, 4, 4, 0, 1]	0.7490	-0.7657	0.7529	0.7450	Infeasible
2	[5, 4, 4, 4, 0, 1]	0.7540	2.9666	0.7580	0.7501	Feasible
2	[4, 4, 4, 4, 2, 0]	0.7586	6.3701	0.7626	0.7547	Feasible
2	[4, 4, 4, 5, 1, 0]	0.7417	-5.9751	0.7458	0.7377	Infeasible
2	[4, 4, 5, 4, 1, 0]	0.7438	-4.4841	0.7478	0.7398	Infeasible
2	[4, 5, 4, 4, 1, 0]	0.7481	-1.3695	0.7521	0.7441	Infeasible
2	[5, 4, 4, 4, 1, 0]	0.7552	3.8392	0.7592	0.7513	Feasible
2	[4, 4, 4, 6, 0, 0]	0.7170	-23.1936	0.7211	0.7128	Infeasible
2	[4, 4, 5, 5, 0, 0]	0.7152	-24.4105	0.7193	0.7110	Infeasible
2	[4, 5, 4, 5, 0, 0]	0.7232	-18.9624	0.7273	0.7190	Infeasible
2	[5, 4, 4, 5, 0, 0]	0.7285	-15.3222	0.7325	0.7244	Infeasible
2	[4, 4, 6, 4, 0, 0]	0.7184	-22.2443	0.7225	0.7142	Infeasible
2	[4, 5, 5, 4, 0, 0]	0.7244	-18.1455	0.7285	0.7202	Infeasible
2	[5, 4, 5, 4, 0, 0]	0.7250	-17.7122	0.7291	0.7209	Infeasible
2	[4, 6, 4, 4, 0, 0]	0.7186	-22.0541	0.7228	0.7145	Infeasible
2	[5, 5, 4, 4, 0, 0]	0.7320	-12.8723	0.7361	0.7279	Infeasible
2	[6, 4, 4, 4, 0, 0]	0.7291	-14.8782	0.7332	0.7250	Infeasible

The solutions highlighted in grey are feasible based on the 100k results. There are 5 feasible solutions: [4, 4, 4, 4, 0, 2], [4, 4, 4, 4, 1, 1], [5, 4, 4, 4, 0, 1], [4, 4, 4, 4, 2, 0] and [5, 4, 4, 4, 1, 0].

D.5 Chi-squared Test: Selection Accuracy Comparison

H_0: There are equal numbers of correct selections for M-CCSB-F and CCSB-F

H_1: There are unequal numbers of correct selections for M-CCSB-F and CCSB-F

δ_γ	**Hypothesised Proportion**	**CCSB-F**	**M-CCSB-F**	**Expected**	χ^2	**Critical Value**
0.05	0.5	33	31	32	0.0625	5.0239
0.025	0.5	100	98	99	0.0202	5.0239

With an α of 5% (with Bonferroni correction), there is no statistical evidence to support that there is a difference in the number of correct selections made by M-CCSB-F and CCSB-F.

Appendix E

Appendix: Chapter 6

E.1 Experiment Settings for Scenario 3

All experiments are conducted on a Lenovo with an Intel(R) Core(TM) i5-6500T CPU @ 2.50GHz and a Mac mini with an Apple M1, running macOS 13.4 (22F66).

For feasibility estimations, the random number seed is 2000033036, and the run time for all 210 solutions is 7.3305 hours for 1 macro-replication.

For 100 macro-replications of CCSB-F and H-CCSB-F, the experiments are run in parallel. Table E1 provides details of the random number seeds used for each experiment.

Table E1: Scenario 3: random number seed settings for macro-replications.

Random Number Index	Random Number Seed	Problem	macro-replications Index
5	3839190294	Scenario 3: CCSB-F	1-11
6	615377162	Scenario 3: CCSB-F	12-22
7	2659974823	Scenario 3: CCSB-F	23-33
8	1283674829	Scenario 3: CCSB-F	34-44
9	3574523394	Scenario 3: CCSB-F	45-55
10	1258189053	Scenario 3: CCSB-F	56-66
11	2948196755	Scenario 3: CCSB-F	67-77
12	1821795604	Scenario 3: CCSB-F	78-89
13	1298501725	Scenario 3: CCSB-F	90-100
14	892462304	Scenario 3: H-CCSB-F	1-20
15	1298749205	Scenario 3: H-CCSB-F	21-40
16	3596989659	Scenario 3: H-CCSB-F	41-60
17	822626078	Scenario 3: H-CCSB-F	61-80
18	830021159	Scenario 3: H-CCSB-F	81-100

E.2 Feasibility Estimation Results for Infeasible Solutions in Scenario 3

Table E2: Scenario 3: feasibility estimation results for infeasible solutions.

Solutions	$\Pr\{Y_i \geq 14\}$	Lower 90% CI	Upper 90% CI	Critical Value	Test Statistics
[6, 7, 6, 6, 3, 2]	0.748478	0.7469	0.7500	2.3264	-2.4804
[7, 7, 6, 6, 4, 0]	0.750176	0.7486	0.7517	2.3264	0.2875
[7, 7, 7, 6, 2, 1]	0.748698	0.7472	0.7502	2.3264	-2.1225
[8, 8, 6, 6, 2, 0]	0.750628	0.7491	0.7522	2.3264	1.0264
[6, 6, 6, 7, 3, 2]	0.73318	0.7290	0.7373	2.9739	-12.0257
[8, 6, 7, 7, 2, 0]	0.7423	0.7382	0.7464	2.9739	-5.5673
[6, 6, 7, 6, 3, 2]	0.74477	0.7407	0.7489	2.9739	-3.7934
[7, 6, 7, 6, 4, 0]	0.746392	0.7449	0.7479	2.4977	-5.8639
[7, 7, 7, 6, 3, 0]	0.73975	0.7356	0.7439	2.9739	-7.3873
[6, 6, 6, 6, 3, 3]	0.73049	0.7263	0.7347	2.9739	-13.9047
[7, 6, 6, 6, 2, 2]	0.74532	0.7412	0.7494	2.9739	-3.3969
[7, 6, 6, 6, 3, 1]	0.74165	0.7375	0.7458	2.9739	-6.0323
[7, 6, 6, 7, 4, 0]	0.7379	0.7338	0.7420	2.9739	-8.7007
[7, 8, 6, 6, 2, 1]	0.74161	0.7375	0.7457	2.9739	-6.0609
[8, 7, 6, 7, 2, 0]	0.749024	0.7475	0.7506	2.4977	-1.5917
[7, 6, 7, 7, 2, 1]	0.7359	0.7318	0.7400	2.9739	-10.1141
[7, 6, 6, 6, 5, 0]	0.74666	0.7451	0.7482	2.4977	-5.4302
[7, 7, 6, 7, 2, 1]	0.74201	0.7379	0.7461	2.9739	-5.7748
[8, 6, 8, 6, 2, 0]	0.74154	0.7374	0.7457	2.9739	-6.1109
[8, 7, 6, 6, 2, 0]	0.73418	0.7300	0.7383	2.9739	-11.3243
[7, 6, 8, 6, 2, 1]	0.73192	0.7278	0.7361	2.9739	-12.9073
[6, 7, 6, 6, 4, 1]	0.73754	0.7334	0.7417	2.9739	-8.9556
[6, 6, 7, 6, 4, 1]	0.7342	0.7188	0.7496	3.4950	-3.5764
[7, 7, 6, 6, 2, 1]	0.7322	0.7167	0.7477	3.4950	-4.0196
[8, 6, 7, 6, 2, 0]	0.7321	0.7166	0.7476	3.4950	-4.0417
[7, 6, 7, 7, 3, 0]	0.732	0.7165	0.7475	3.4950	-4.0638
[7, 7, 6, 7, 3, 0]	0.732	0.7165	0.7475	3.4950	-4.0638
[6, 6, 6, 6, 5, 1]	0.7315	0.7160	0.7470	3.4950	-4.1742
[7, 8, 6, 6, 3, 0]	0.7285	0.7130	0.7440	3.4950	-4.8341
[7, 6, 6, 6, 4, 0]	0.7283	0.7128	0.7438	3.4950	-4.8780
[7, 6, 8, 6, 3, 0]	0.7276	0.7120	0.7432	3.4950	-5.0313
[6, 6, 6, 7, 4, 1]	0.7271	0.7115	0.7427	3.4950	-5.1406
[6, 7, 7, 6, 3, 1]	0.7257	0.7101	0.7413	3.4950	-5.4462
[8, 6, 6, 8, 2, 0]	0.7251	0.7095	0.7407	3.4950	-5.5769
[6, 7, 7, 6, 2, 2]	0.7239	0.7083	0.7395	3.4950	-5.8378
[7, 7, 6, 6, 3, 0]	0.7212	0.7055	0.7369	3.4950	-6.4224

Solutions	$\Pr\{Y_i \geq 14\}$	Lower 90% CI	Upper 90% CI	Critical Value	Test Statistics
[7, 6, 7, 6, 2, 1]	0.721	0.7053	0.7367	3.4950	-6.4656
[8, 6, 6, 7, 2, 0]	0.7187	0.7030	0.7344	3.4950	-6.9609
[6, 7, 6, 7, 2, 2]	0.7181	0.7024	0.7338	3.4950	-7.0897
[7, 6, 7, 6, 3, 0]	0.7169	0.7012	0.7326	3.4950	-7.3469
[6, 8, 6, 6, 3, 1]	0.715	0.6992	0.7308	3.4950	-7.7530
[7, 6, 6, 8, 2, 1]	0.7146	0.6988	0.7304	3.4950	-7.8383
[6, 8, 6, 6, 2, 2]	0.7145	0.6987	0.7303	3.4950	-7.8596
[7, 6, 6, 7, 2, 1]	0.7136	0.6978	0.7294	3.4950	-8.0513
[6, 6, 7, 7, 3, 1]	0.7124	0.6966	0.7282	3.4950	-8.3063
[6, 7, 6, 6, 2, 3]	0.7119	0.6961	0.7277	3.4950	-8.4124
[7, 7, 8, 6, 2, 0]	0.7111	0.6953	0.7269	3.4950	-8.5820
[6, 6, 6, 6, 3, 2]	0.7109	0.6951	0.7267	3.4950	-8.6244
[7, 8, 7, 6, 2, 0]	0.7107	0.6949	0.7265	3.4950	-8.6667
[6, 7, 6, 7, 3, 1]	0.7106	0.6948	0.7264	3.4950	-8.6879
[6, 7, 7, 6, 4, 0]	0.7104	0.6945	0.7263	3.4950	-8.7302
[6, 6, 6, 6, 4, 1]	0.7097	0.6938	0.7256	3.4950	-8.8781
[8, 6, 6, 6, 2, 0]	0.7091	0.6932	0.7250	3.4950	-9.0048
[6, 7, 6, 6, 2, 2]	0.709	0.6931	0.7249	3.4950	-9.0259
[6, 7, 6, 6, 5, 0]	0.709	0.6931	0.7249	3.4950	-9.0259
[6, 6, 7, 7, 2, 2]	0.7081	0.6922	0.7240	3.4950	-9.2157
[7, 6, 6, 7, 3, 0]	0.7075	0.6916	0.7234	3.4950	-9.3420
[7, 7, 7, 7, 2, 0]	0.7072	0.6913	0.7231	3.4950	-9.4051
[7, 6, 6, 8, 3, 0]	0.707	0.6911	0.7229	3.4950	-9.4472
[6, 6, 8, 6, 2, 2]	0.7055	0.6896	0.7214	3.4950	-9.7622
[6, 6, 7, 6, 2, 3]	0.7052	0.6893	0.7211	3.4950	-9.8251
[6, 6, 7, 6, 5, 0]	0.705	0.6891	0.7209	3.4950	-9.8670
[6, 6, 8, 6, 3, 1]	0.7048	0.6889	0.7207	3.4950	-9.9089
[6, 6, 7, 6, 2, 2]	0.7046	0.6887	0.7205	3.4950	-9.9508
[6, 8, 6, 6, 4, 0]	0.7024	0.6864	0.7184	3.4950	-10.4106
[6, 7, 6, 6, 3, 1]	0.7014	0.6854	0.7174	3.4950	-10.6191
[6, 6, 6, 7, 5, 0]	0.7013	0.6853	0.7173	3.4950	-10.6399
[7, 8, 6, 7, 2, 0]	0.7007	0.6847	0.7167	3.4950	-10.7648
[7, 6, 8, 7, 2, 0]	0.6999	0.6839	0.7159	3.4950	-10.9311
[6, 6, 7, 6, 3, 1]	0.6978	0.6817	0.7139	3.4950	-11.3667
[6, 7, 6, 7, 4, 0]	0.6975	0.6814	0.7136	3.4950	-11.4289
[7, 7, 6, 7, 2, 0]	0.6973	0.6812	0.7134	3.4950	-11.4702
[6, 8, 7, 6, 2, 1]	0.6947	0.6786	0.7108	3.4950	-12.0072
[6, 6, 6, 7, 3, 1]	0.6941	0.6780	0.7102	3.4950	-12.1308

Solutions	$\Pr\{Y_i \geq 14\}$	Lower 90% CI	Upper 90% CI	Critical Value	Test Statistics
[6, 6, 8, 6, 4, 0]	0.694	0.6779	0.7101	3.4950	-12.1514
[6, 6, 6, 7, 2, 3]	0.6937	0.6776	0.7098	3.4950	-12.2131
[6, 7, 7, 7, 2, 1]	0.6936	0.6775	0.7097	3.4950	-12.2337
[6, 6, 6, 8, 2, 2]	0.6934	0.6773	0.7095	3.4950	-12.2749
[6, 8, 7, 6, 3, 0]	0.6931	0.6770	0.7092	3.4950	-12.3366
[6, 6, 7, 7, 4, 0]	0.693	0.6769	0.7091	3.4950	-12.3571
[7, 6, 6, 6, 2, 1]	0.6929	0.6768	0.7090	3.4950	-12.3777
[7, 7, 7, 6, 2, 0]	0.6924	0.6763	0.7085	3.4950	-12.4804
[7, 9, 6, 6, 2, 0]	0.6907	0.6745	0.7069	3.4950	-12.8292
[7, 8, 6, 6, 2, 0]	0.6898	0.6736	0.7060	3.4950	-13.0134
[7, 7, 6, 8, 2, 0]	0.6879	0.6717	0.7041	3.4950	-13.4017
[7, 6, 6, 6, 3, 0]	0.6878	0.6716	0.7040	3.4950	-13.4221
[6, 6, 6, 6, 6, 0]	0.6877	0.6715	0.7039	3.4950	-13.4425
[6, 7, 8, 6, 2, 1]	0.6874	0.6712	0.7036	3.4950	-13.5037
[6, 7, 7, 7, 3, 0]	0.6874	0.6712	0.7036	3.4950	-13.5037
[6, 6, 6, 8, 3, 1]	0.6873	0.6711	0.7035	3.4950	-13.5241
[6, 7, 8, 6, 3, 0]	0.6867	0.6705	0.7029	3.4950	-13.6464
[6, 6, 6, 6, 2, 3]	0.6863	0.6701	0.7025	3.4950	-13.7279
[7, 6, 7, 8, 2, 0]	0.6858	0.6696	0.7020	3.4950	-13.8297
[6, 6, 6, 6, 2, 4]	0.6853	0.6691	0.7015	3.4950	-13.9314
[6, 6, 6, 7, 2, 2]	0.6844	0.6682	0.7006	3.4950	-14.1143
[7, 6, 8, 6, 2, 0]	0.6834	0.6671	0.6997	3.4950	-14.3172
[6, 6, 8, 7, 2, 1]	0.683	0.6667	0.6993	3.4950	-14.3984
[6, 8, 6, 6, 2, 1]	0.6819	0.6656	0.6982	3.4950	-14.6212
[6, 7, 6, 6, 4, 0]	0.6818	0.6655	0.6981	3.4950	-14.6414
[6, 9, 6, 6, 2, 1]	0.6808	0.6645	0.6971	3.4950	-14.8437
[6, 8, 6, 7, 2, 1]	0.6807	0.6644	0.6970	3.4950	-14.8639
[7, 6, 9, 6, 2, 0]	0.6798	0.6635	0.6961	3.4950	-15.0458
[6, 6, 6, 6, 3, 1]	0.6776	0.6613	0.6939	3.4950	-15.4893
[6, 7, 7, 6, 2, 1]	0.6766	0.6603	0.6929	3.4950	-15.6906
[6, 8, 6, 7, 3, 0]	0.6766	0.6603	0.6929	3.4950	-15.6906
[7, 6, 7, 7, 2, 0]	0.6761	0.6597	0.6925	3.4950	-15.7911
[6, 6, 7, 6, 4, 0]	0.6743	0.6579	0.6907	3.4950	-16.1525
[6, 6, 6, 6, 5, 0]	0.6742	0.6578	0.6906	3.4950	-16.1725
[6, 6, 8, 7, 3, 0]	0.6737	0.6573	0.6901	3.4950	-16.2728
[7, 7, 6, 6, 2, 0]	0.6715	0.6551	0.6879	3.4950	-16.7131
[6, 9, 6, 6, 3, 0]	0.6678	0.6513	0.6843	3.4950	-17.4513
[6, 7, 6, 7, 2, 1]	0.667	0.6505	0.6835	3.4950	-17.6105
[6, 7, 6, 8, 2, 1]	0.667	0.6505	0.6835	3.4950	-17.6105
[6, 8, 6, 6, 3, 0]	0.6668	0.6503	0.6833	3.4950	-17.6503
[6, 6, 6, 7, 4, 0]	0.6654	0.6489	0.6819	3.4950	-17.9285

Solutions	$\Pr\{Y_i \geq 14\}$	Lower 90% CI	Upper 90% CI	Critical Value	Test Statistics
[6, 7, 7, 6, 3, 0]	0.665	0.6485	0.6815	3.4950	-18.0079
[6, 6, 9, 6, 3, 0]	0.665	0.6485	0.6815	3.4950	-18.0079
[6, 6, 7, 8, 2, 1]	0.6648	0.6483	0.6813	3.4950	-18.0476
[6, 7, 6, 7, 3, 0]	0.6644	0.6479	0.6809	3.4950	-18.1270
[6, 7, 6, 8, 3, 0]	0.6623	0.6458	0.6788	3.4950	-18.5432
[6, 6, 9, 6, 2, 1]	0.6613	0.6448	0.6778	3.4950	-18.7411
[6, 6, 6, 8, 4, 0]	0.6613	0.6448	0.6778	3.4950	-18.7411
[6, 6, 8, 6, 2, 1]	0.6612	0.6447	0.6777	3.4950	-18.7609
[6, 6, 6, 6, 2, 2]	0.6607	0.6442	0.6772	3.4950	-18.8597
[6, 6, 6, 6, 4, 0]	0.6589	0.6423	0.6755	3.4950	-19.2153
[6, 6, 7, 8, 3, 0]	0.6589	0.6423	0.6755	3.4950	-19.2153
[6, 6, 7, 7, 3, 0]	0.6576	0.6410	0.6742	3.4950	-19.4716
[7, 6, 6, 7, 2, 0]	0.657	0.6404	0.6736	3.4950	-19.5899
[7, 6, 7, 6, 2, 0]	0.6568	0.6402	0.6734	3.4950	-19.6293
[7, 6, 6, 9, 2, 0]	0.6566	0.6400	0.6732	3.4950	-19.6687
[6, 7, 6, 6, 2, 1]	0.6563	0.6397	0.6729	3.4950	-19.7277
[6, 8, 8, 6, 2, 0]	0.6558	0.6392	0.6724	3.4950	-19.8261
[6, 6, 7, 7, 2, 1]	0.6549	0.6383	0.6715	3.4950	-20.0032
[6, 7, 8, 7, 2, 0]	0.6542	0.6376	0.6708	3.4950	-20.1408
[6, 6, 8, 6, 3, 0]	0.6532	0.6366	0.6698	3.4950	-20.3372
[6, 8, 7, 7, 2, 0]	0.6512	0.6345	0.6679	3.4950	-20.7295
[7, 6, 6, 8, 2, 0]	0.6486	0.6319	0.6653	3.4950	-21.2386
[6, 9, 7, 6, 2, 0]	0.6477	0.6310	0.6644	3.4950	-21.4146
[6, 6, 6, 9, 2, 1]	0.6474	0.6307	0.6641	3.4950	-21.4733
[6, 6, 7, 6, 2, 1]	0.6468	0.6301	0.6635	3.4950	-21.5905
[6, 6, 7, 6, 3, 0]	0.6448	0.6281	0.6615	3.4950	-21.9809
[6, 7, 9, 6, 2, 0]	0.643	0.6263	0.6597	3.4950	-22.3317
[6, 8, 7, 6, 2, 0]	0.6398	0.6230	0.6566	3.4950	-22.9544
[6, 7, 7, 7, 2, 0]	0.6387	0.6219	0.6555	3.4950	-23.1681
[6, 7, 6, 6, 3, 0]	0.6379	0.6211	0.6547	3.4950	-23.3235
[6, 7, 8, 6, 2, 0]	0.6379	0.6211	0.6547	3.4950	-23.3235
[6, 7, 7, 8, 2, 0]	0.6374	0.6206	0.6542	3.4950	-23.4205
[6, 6, 6, 8, 2, 1]	0.6369	0.6201	0.6537	3.4950	-23.5176
[6, 9, 6, 7, 2, 0]	0.6352	0.6184	0.6520	3.4950	-23.8472
[6, 6, 6, 8, 3, 0]	0.6338	0.6170	0.6506	3.4950	-24.1184
[6, 8, 6, 8, 2, 0]	0.6317	0.6148	0.6486	3.4950	-24.5249
[6, 6, 6, 9, 3, 0]	0.6304	0.6135	0.6473	3.4950	-24.7762
[6, 6, 9, 7, 2, 0]	0.629	0.6121	0.6459	3.4950	-25.0467
[7, 6, 6, 6, 2, 0]	0.6279	0.6110	0.6448	3.4950	-25.2592
[6, 6, 6, 7, 2, 1]	0.6278	0.6109	0.6447	3.4950	-25.2785
[6, 6, 8, 8, 2, 0]	0.6244	0.6075	0.6413	3.4950	-25.9342

Solutions	$\Pr\{Y_i \geq 14\}$	Lower 90% CI	Upper 90% CI	Critical Value	Test Statistics
[6, 8, 6, 7, 2, 0]	0.6242	0.6073	0.6411	3.4950	-25.9728
[6, 6, 6, 6, 2, 1]	0.6241	0.6072	0.6410	3.4950	-25.9921
[6, 9, 6, 6, 2, 0]	0.6231	0.6062	0.6400	3.4950	-26.1847
[6, 6, 6, 7, 3, 0]	0.6219	0.6050	0.6388	3.4950	-26.4158
[6, 6, 8, 7, 2, 0]	0.6191	0.6021	0.6361	3.4950	-26.9545
[6, 7, 7, 6, 2, 0]	0.618	0.6010	0.6350	3.4950	-27.1660
[6, 10, 6, 6, 2, 0]	0.616	0.5990	0.6330	3.4950	-27.5503
[6, 7, 6, 8, 2, 0]	0.6156	0.5986	0.6326	3.4950	-27.6272
[6, 7, 6, 9, 2, 0]	0.6111	0.5941	0.6281	3.4950	-28.4909
[6, 6, 10, 6, 2, 0]	0.6106	0.5936	0.6276	3.4950	-28.5867
[6, 6, 9, 6, 2, 0]	0.6066	0.5895	0.6237	3.4950	-29.3534
[6, 6, 7, 9, 2, 0]	0.6056	0.5885	0.6227	3.4950	-29.5450
[6, 8, 6, 6, 2, 0]	0.6051	0.5880	0.6222	3.4950	-29.6408
[6, 6, 7, 8, 2, 0]	0.6048	0.5877	0.6219	3.4950	-29.6982
[6, 7, 6, 7, 2, 0]	0.6047	0.5876	0.6218	3.4950	-29.7174
[6, 6, 6, 6, 3, 0]	0.6035	0.5864	0.6206	3.4950	-29.9472
[6, 6, 8, 6, 2, 0]	0.5964	0.5793	0.6135	3.4950	-31.3058
[6, 7, 6, 6, 2, 0]	0.5916	0.5744	0.6088	3.4950	-32.2238
[6, 6, 7, 7, 2, 0]	0.5891	0.5719	0.6063	3.4950	-32.7018
[6, 6, 7, 6, 2, 0]	0.5847	0.5675	0.6019	3.4950	-33.5431
[6, 6, 6, 8, 2, 0]	0.5762	0.5589	0.5935	3.4950	-35.1691
[6, 6, 6, 10, 2, 0]	0.5743	0.5570	0.5916	3.4950	-35.5327
[6, 6, 6, 9, 2, 0]	0.5732	0.5559	0.5905	3.4950	-35.7433
[6, 6, 6, 7, 2, 0]	0.5714	0.5541	0.5887	3.4950	-36.0881
[6, 6, 6, 6, 2, 0]	0.5458	0.5284	0.5632	3.4950	-41.0104

E.3 Chi-Squared Test: Selection Accuracy Comparison

H_0: There are equal rates of correct selections between CCSB-F and H-CCSB-F.

H_1: There are unequal rates of correct selections between CCSB-F and H-CCSB-F.

CCSB-F	**H-CCSB-F**	**Expected**	χ^2	**Critical Value**
65	76	70.5	0.8582	3.8415

With a significance level of 5%, there is no statistical evidence to support that M-CCSB-F and CCSB-F made different numbers of correct selections.

Reference List

Álvarez-Miranda, E., M. Goycoolea, I. Ljubić, and M. Sinnl (2021). “The Generalized Reserve Set Covering Problem with Connectivity and Buffer Requirements”. In: *European Journal of Operational Research* 289.3, pp. 1013–1029.

Amaran, S., N. V. Sahinidis, B. Sharda, and S. J. Bury (2016). “Simulation Optimization: A Review Of Algorithms And Applications”. In: *Annals of Operations Research* 240.1, pp. 351–380.

Anderson, D. R., D. J. Sweeney, T. A. Williams, J. Freeman, and E. Shoesmith (2014). *Statistics For Business And Economics*. 3rd ed. SP10 5BE, UK: Cengage Learning EMEA.

Andradóttir, S. (2006). “Chapter 20 An Overview of Simulation Optimization via Random Search”. In: *Simulation*. Ed. by S. G. Henderson and B. L. Nelson. Vol. 13. Handbooks in Operations Research and Management Science. Elsevier, pp. 617–631.

Andradóttir, S. and S.-H. Kim (2010). “Fully Sequential Procedures For Comparing Constrained Systems Via Simulation”. In: *Naval Research Logistics (NRL)* 57.5, pp. 403–421.

Antonelli, S., K. Boysen, C. Piechowski, M. Smith, and G. Willard (2016). “An Analysis of Wolf-Livestock Conflict Hotspots and Conflict Reduction Strategies in Northern California”. MA thesis. Bren School of Environmental Science and Management at the University of California, Santa Barbara.

Arthur, J. L., R. G. Haight, C. A. Montgomery, and S. Polasky (2002). "Analysis Of The Threshold And Expected Coverage Approaches To The Probabilistic Reserve Site Selection Problem". In: *Environmental Modeling and Assessment* 7.2, pp. 81–89.

Balci, O. (1994). "Validation, verification, and testing techniques throughout the life cycle of a simulation study". In: *Annals of Operations Research* 53.1, pp. 121–173.

Bandara, D. and L. Weerasena (2016). "Metaheuistic Algorithms to Solve the Reserve Design Problem". In: *Annual International Conference on Enterprise Marketing and Globalization*, pp. 107–111.

Berge, C. (1957). "Two Theorems in Graph Theory". In: *Proceedings of the National Academy of Sciences of the United States of America* 43.9, pp. 842–844.

Billionnet, A. (2013). "Mathematical optimization ideas for biodiversity conservation". In: *European Journal of Operational Research* 231.3, pp. 514–534.

Blanco, J. C. and Y. Cortés (2007). "Dispersal patterns, social structure and mortality of wolves living in agricultural habitats in Spain". In: *Journal of Zoology* 273.1, pp. 114–124.

Boesel, J., B. L. Nelson, and N. Ishii (2003). "A framework for simulation-optimization software". In: *IIE Transactions* 35.3, pp. 221–229.

Bonneau, M., R. Sabbadin, F. A. Johnson, and B. Stith (2018). "Dynamic minimum set problem for reserve design: Heuristic solutions for large problems". In: *PLoS ONE* 13.3. Cited By :1 Export Date: 8 October 2019.

Box, G. (1979). "Robustness in the Strategy of Scientific Model Building". In: *Robustness in Statistics*. Ed. by R. L. LAUNER and G. N. WILKINSON. Academic Press, pp. 201–236.

Boysen, K. (2016). *Potential Wolf Habitat, California, Logistic Regression.*

Butchart, S. H. M., J. P. W. Scharlemann, M. I. Evans, S. Quader, S. Aricò, J. Arinaitwe, M. Balman, L. A. Bennun, B. Bertzky, C. Besançon, T. M. Boucher, T. M. Brooks, I. J. Burfield, N. D. Burgess, S. Chan, R. P. Clay, M. J. Crosby, N. C. Davidson, N. De Silva, C. Devenish, G. C. L. Dutson, D. F. D. z. Fernández, L. D. C. Fishpool, C. Fitzgerald, M. Foster, M. F. Heath, M. Hockings, M. Hoffmann, D. Knox, F. W. Larsen, J. F. Lamoreux, C. Loucks, I. May, J. Millett, D. Molloy, P. Morling, M. Parr, T. H. Ricketts, N. Seddon, B. Skolnik, S. N. Stuart, A. Upgren, and S. Woodley (Mar. 2012). "Protecting Important Sites for Biodiversity Contributes to Meeting Global Conservation Targets". In: *PLOS ONE* 7.3, pp. 1–8.

Cabeza, M. and A. Moilanen (2001). "Design of reserve networks and the persistence of biodiversity". In: *Trends in Ecology & Evolution* 16.5, pp. 242–248.

California Department of Fish and Wildlife (2016). *Conservation Plan for Gray Wolves in California 2016*. Tech. rep. Sacramento, California: California Department of Fish and Wildlife.

Camm, J. D., S. K. Norman, S. Polasky, and A. R. Solow (2002). "Nature Reserve Site Selection to Maximize Expected Species Covered". In: *Operations Research* 50.6, pp. 946–955.

Camm, J. D., S. Polasky, A. Solow, and B. Csuti (1996). "A note on optimal algorithms for reserve site selection". In: *Biological Conservation* 78.3, pp. 353–355.

CDFW (May 2023). *Approximate Area Of Gray Wolf Activity*. Tech. rep. California Department of Fish and Wildlife.

Chapron, G., C. Wikenros, O. Liberg, P. Wabakken, Ø. Flagstad, C. Milleret, J. Månsson, L. Svensson, B. Zimmermann, M. Åkesson, and H. Sand (2016). "Estimating wolf (Canis lupus) population size from number of packs and an individual based model". In: *Ecological Modelling* 339, pp. 33–44.

Chen, C.-H., S. E. Chick, L. H. Lee, and N. A. Pujowidianto (2015). "Ranking and selection: Efficient simulation budget allocation". In: *Handbook of Simulation Optimization*. Ed. by M. C. Fu, pp. 45–80.

Chen, C.-H., Q.-S. Jia, and L. H. Lee, eds. (2013). *Stochastic Simulation Optimization for Discrete Event Systems*. WORLD SCIENTIFIC.

Chen, C.-H. and L. H. Lee (2011). *Stochastic simulation optimization: an optimal computing budget allocation*. Vol. 1. World scientific.

Chen, C.-H., J. Lin, E. Yücesan, and S. E. Chick (2000). "Simulation Budget Allocation for Further Enhancing the Efficiency of Ordinal Optimization". In: *Discrete Event Dynamic Systems* 10.3, pp. 251–270.

Church, R. and C. ReVelle (1974). "The maximal covering location problem". In: *Papers of the Regional Science Association* 32.1, pp. 101–118.

Clemens, M. A., C. S. Revelle, and J. C. Williams (1999). "Reserve design for species preservation". In: *European Journal of Operational Research* 112.2. Cited By :33 Export Date: 8 October 2019, pp. 273–283.

Costello, C. and S. Polasky (2004). "Dynamic reserve site selection". In: *Resource and Energy Economics* 26.2, pp. 157–174.

Currie, C. and T. Monks (2021). "A practical approach to subset selection for multi-objective optimization via simulation". In: *ACM Transactions on Modeling and Computer Simulation* 31.4, pp. 1–15.

Delmelle, E., M. R. Desjardins, and J. Deng (2017). "Designing spatially cohesive nature reserves with backup coverage". In: *International Journal of Geographical Information Science* 31.12. Cited By :3 Export Date: 8 October 2019, pp. 2505–2523.

Drechsler, M. (2020). "Model-based integration of ecology and socio-economics for the management of biodiversity and ecosystem services: State of the art, diversity and current trends". In: *Environmental Modelling and Software* 134, p. 104892.

Eckman, D. J., S. G. Henderson, and S. Shashaani (2023). “Stochastic Constraints: How Feasible Is Feasible?” In: *2023 Winter Simulation Conference (WSC)*, pp. 3589–3600.

Eckman, D. J., M. Plumlee, and B. L. Nelson (2020). “Revisiting subset selection”. In: *2020 Winter Simulation Conference (WSC)*. IEEE, pp. 2972–2983.

Ellingson, A. R. and P. M. Lukacs (2003). “Improving Methods for Regional Landbird Monitoring: A Reply to Hutto and Young”. In: *Wildlife Society Bulletin (1973-2006)* 31.3, pp. 896–902.

Fahrig, L. (2001). “How much habitat is enough?” In: *Biological Conservation* 100.1, pp. 65–74.

Frazier, P. I. (2018). *A Tutorial on Bayesian Optimization*.

Fu, M. C., ed. (2015). *Handbook of Simulation Optimization*. 1st ed. New York, NY: Springer.

Fu, M. C. and S. G. Henderson (2017). “History of seeking better solutions, AKA simulation optimization”. In: *2017 Winter Simulation Conference (WSC)*, pp. 131–157.

Fuller, T. K., L. D. Mech, and J. F. Cochrane (2003). “Wolves: Behavior, Ecology, and Conservation”. In: ed. by L. D. Mech and L. Boitani. Chicago IL: University of Chicago Press. Chap. Wolf population dynamics.

Fuller, T. K. (1989). “Population Dynamics of Wolves in North-Central Minnesota”. In: *Wildlife Monographs* 105, pp. 3–41.

Fulton, E. A., N. J. Bax, R. H. Bustamante, J. M. Dambacher, C. Dichmont, P. K. Dunstan, K. R. Hayes, A. J. Hobday, R. Pitcher, É. E. Plagányi, A. E. Punt, M. Savina-Rolland, A. D. M. Smith, and D. C. Smith (2015). “Modelling marine protected areas: Insights and hurdles”. In: *Philosophical Transactions of the Royal Society B: Biological Sciences* 370.1681.

Gaucherel, C., C. Carpentier, I. Geijzendorffer, C. Noûs, and F. Pommereau (2021). "Discrete-event models for conservation assessment of integrated ecosystems". In: *Ecological Informatics* 61, p. 101205.

Haight, R., B. Cypher, P. Kelly, S. E. Phillips, K. Ralls, and H. Possingham (2004). "Optimizing reserve expansion for disjunct populations of San Joaquin kit fox". In: *Biological Conservation* 117, pp. 61–72.

Haight, R. G. and L. E. Travis (1997). "Wildlife conservation planning using stochastic optimization and importance sampling". In: *Forest Science* 43.1, pp. 129–139.

Haight, R. G., D. J. Mladenoff, and A. P. Wydeven (1998). "Modeling Disjunct Gray Wolf Populations in Semi-Wild Landscapes". In: *Conservation Biology* 12.4, pp. 879–888.

Haight, R. G. and S. A. Snyder (2009). "Integer Programming Methods for Reserve Selection and Design". In: *Spatial Conservation Prioritization: Quantitative Methods and Computational Tools*. Ed. by A. Moilanen, K. A. Wilson, and H. P. Possingham. Oxford University Press, p. 0.

Haight, R. G. and L. E. Travis (2008). "Reserve Design to Maximize Species Persistence". In: *Environmental Modeling and Assessment* 13, pp. 243–253.

Haight, R. G., L. E. Travis, K. Nimerfro, and L. D. Mech (2002). "Computer Simulation of Wolf-Removal Strategies for Animal Damage Control". In: *Wildlife Society Bulletin (1973-2006)* 30.3, pp. 844–852.

Han, J., S.-H. Kim, and C. Park (2021). "Improved Penalty Function with Memory for Stochastically Constrained Optimization via Simulation". In: *ACM Trans. Model. Comput. Simul.* 31.4.

Harrington, F. H., P. C. Paquet, J. Ryon, and J. C. Fentress (1982). "Wolves of the world: Perspectives of behavior, ecology, and conservation". In: ed. by F. H. Harrington and P. C. Paquet. Park Ridge, New Jersey: Noyes Publications. Chap. Monogamy in wolves: a review of the evidence, pp. 209–222.

Hayes, R. D. and A. S. Harestad (2000). "Demography of a recovering wolf population in the Yukon". In: *Canadian Journal of Zoology* 78.1, pp. 36–48.

Hof, J. and M. G. Raphael (1997). "Optimization of habitat placement: A case study of the Northern Spotted Owl in the Olympic Peninsula". In: *Ecological Applications* 7.4, pp. 1160–1169.

Hong, J. L., J. Luo, and B. L. Nelson (2015). "Chance Constrained Selection of the Best". In: *INFORMS Journal on Computing* 27, pp. 317–334.

Hong, L. J., W. Fan, and J. Luo (2021). "Review on ranking and selection: A new perspective". In: *Frontiers of Engineering Management* 8.3, pp. 321–343.

Horng, S.-C. and S.-S. Lin (2023). "Improved Beluga Whale Optimization for Solving the Simulation Optimization Problems with Stochastic Constraints". In: *Mathematics* 11.8.

Hunter, S. R. and B. L. Nelson (2017). "Parallel ranking and selection". In: *Advances in Modeling and Simulation: Seminal Research from 50 Years of Winter Simulation Conferences*. Springer, pp. 249–275.

IPBES (2019). *Global Assessment Report on Biodiversity and Ecosystem Services*. Tech. rep. Paris, France: Intergovernmental Science-Policy Platform on Biodiversity and Ecosystem Services.

IUCN (2023). *The IUCN Red List of Threatened Species. Version 2022-2.* URL: `https://www.iucnredlist.org` (visited on 2023).

Jian, N. and S. G. Henderson (2015). "An introduction to simulation optimization". In: *2015 Winter Simulation Conference (WSC)*, pp. 1780–1794.

Jimenez, M. D., V. J. Asher, C. Bergman, E. Bangs, and S. Woodruff (2008). "Gray wolves, canis lupus, killed by cougars, puma concolor, and a grizzly bear, ursus arctos, in Montana, Alberta, and Wyoming". In: *Canadian Field Naturalist* 122(1), pp. 76–78.

Joppa, L. N. and A. Pfaff (2011). "Global protected area impacts". In: *Proceedings of the Royal Society B: Biological Sciences* 278.1712, pp. 1633–1638.

Kim, S.-h. and B. Nelson (2001). "A Fully Sequential Procedure for Indifference-Zone Selection in Simulation". In: *ACM Transactions on Modeling and Computer Simulation* 11.3, pp. 251–273.

Kim, S.-H. and B. L. Nelson (2006). "Chapter 17 Selecting the Best System". In: *Simulation*. Ed. by S. G. Henderson and B. L. Nelson. Vol. 13. Handbooks in Operations Research and Management Science. Elsevier, pp. 501–534.

– (2007). "Recent advances in ranking and selection". In: *2007 Winter Simulation Conference*, pp. 162–172.

Kingsland, S. (June 2002). "Creating a Science of Nature Reserve Design: Perspectives from History". In: *Environmental Modeling and Assessment* 7.

Kojola, I., J. Aspi, A. Hakala, S. Heikkinen, C. Ilmoni, and S. Ronkainen (Apr. 2006). "Dispersal in an Expanding Wolf Population in Finland". In: *Journal of Mammalogy* 87.2, pp. 281–286.

Lam, H. and H. Qian (2019). "Validating Optimization with Uncertain Constraints". In: *2019 Winter Simulation Conference (WSC)*, pp. 3621–3632.

Langhammer, P. F., J. W. Bull, J. E. Bicknell, J. L. Oakley, M. H. Brown, M. W. Bruford, S. H. M. Butchart, J. A. Carr, D. Church, R. Cooney, S. Cutajar, W. Foden, M. N. Foster, C. Gascon, J. Geldmann, P. Genovesi, M. Hoffmann, J. Howard-McCombe, T. Lewis, N. B. W. Macfarlane, Z. E. Melvin, R. S. Merizalde, M. G. Morehouse, S. Pagad, B. Polidoro, W. Sechrest, G. Segelbacher, K. G. Smith, J. Steadman, K. Strongin, J. Williams, S. Woodley, and T. M. Brooks (2024). "The positive impact of conservation action". In: *Science* 384.6694, pp. 453–458.

Lee, L. H., C.-h. Chen, E. P. Chew, J. Li, N. A. Pujowidianto, and S. Zhang (2010). "A review of optimal computing budget allocation algorithms for simulation optimization problem". In: *International Journal of Operations Research* 7.2, pp. 19–31.

Lee, M. L., C. Park, and D. U. Park (2018). "Self-adjusting the tolerance level in a fully sequential feasibility check procedure". In: *European Journal of Operational Research* 271.2, pp. 733–745.

Luck, G. W. (2007). "A review of the relationships between human population density and biodiversity". In: *Biological Reviews* 82.4, pp. 607–645.

McDonnell, M. D., H. P. Possingham, I. R. Ball, and E. A. Cousins (2002). "Mathematical methods for spatially cohesive reserve design". In: *Environmental Modeling and Assessment* 7.2. Cited By :178 Export Date: 8 October 2019, pp. 107–114.

Mech, L. D. and L. Boitani (2003). "Wolves: Behavior, Ecology, and Conservation". In: ed. by L. Mech and L. Boitani. Chicago, Illinois: University of Chicago Press. Chap. Ecosystem effects of wolves, pp. 158–160.

Mech, L. D. (1997). *The arctic wolf: Ten years with the pack*. Swan Hill.

Mech, L. (1970). *The wolf: The ecology and behavior of an endangered species*. Garden City, New Jersey: Natural History Press.

Meester, G. A., A. Mehrotra, J. S. Ault, and E. K. Baker (2004). "Designing Marine Reserves for Fishery Management". In: *Management Science* 50.8, pp. 1031–1043.

Miller, M. L., K. M. Ringelman, J. C. Schank, and J. M. Eadie (2014). "SWAMP: An agent-based model for wetland and waterfowl conservation management". In: *SIMULATION* 90.1, pp. 52–68.

Morales-González, A., A. Fernández-Gil, M. Quevedo, and E. Revilla (2022). "Patterns and determinants of dispersal in grey wolves (Canis lupus)". In: *Biological Reviews* 97.2, pp. 466–480.

Nelson, B. L. (2022). "Let's Do Ranking & Selection". In: *2022 Winter Simulation Conference (WSC)*, pp. 1–12.

Nelson, B. L. and L. Pei (2021). *Foundations and Methods of Stochastic Simulation: A First Course*. 2nd ed. Vol. 316. Springer Nature.

Nickel, T. and S. Walther (2019). "Recolonizing Gray Wolves (Canis lupus) in Northern California: Preliminary Analysis of Suitable Areas for Reoccupancy". In: *Natural Areas Journal* 39.3, pp. 384–390.

Packard, J. M. and L. D. Mech (1980). "Biosocial Mechanisms of Population Regulation". In: ed. by M. N. Cohen, R. Malpass, and H. Klein. New Haven: Yale University Press. Chap. Population regulation in wolves, pp. 135–150.

Paquet, P. and L. Carbyn (2003). "Wild Mammals of North America: Biology, Management, and Conservation". In: ed. by G. Feldhamer, B. Thompson, and J. Champan. JHU Press, Baltimore, Maryland, USA. Chap. 23, pp. 483–510.

Peterson, R. O., N. J. Thomas, J. M. Thurber, J. A. Vucetich, and T. A. Waite (1998). "Population limitation and the wolves of Isle Royale". In: *Journal of Mammalogy* 79, pp. 828–841.

Peterson, R. O., J. D. Woolington, and T. N. Bailey (1984). "Wolves of the Kenai Peninsula, Alaska". In: *Wildlife Monographs* 88.

Possingham, H. P. and I. Davies (1995). "ALEX: A model for the viability analysis of spatially structured populations". In: *Biological Conservation* 73.2, pp. 143–150.

Rinott, Y. (1978). "On two-stage selection procedures and related probability-inequalities". In: *Communications in Statistics - Theory and Methods* 7.8, pp. 799–811.

Robinson, S., R. Brooks, K. Kotiadis, and D. Van Der Zee, eds. (2010). *Conceptual Modeling for Discrete-Event Simulation*. CRC Press.

Rosen, K. H. (2007). *Discrete Mathematics and Its Applications*. 6th ed. New York, NY: McGraw-Hill.

Sætersdal, M., J. M. Line, and H. Birks (1993). "How to maximize biological diversity in nature reserve selection: Vascular plants and breeding birds in deciduous woodlands, western Norway". In: *Biological Conservation* 66.2, pp. 131–138.

Salemi, P. L., E. Song, B. L. Nelson, and J. Staum (2019). "Gaussian Markov Random Fields for Discrete Optimization via Simulation: Framework and Algorithms". In: *Operations Research* 67.1, pp. 250–266.

Snyder, S., C. ReVelle, and R. Haight (2004). "One- and two-objective approaches to an area-constrained habitat reserve site selection problem". In: *Biological Conservation* 119.4, pp. 565–574.

Snyder, S. A. and R. G. Haight (2016). "Application of the Maximal Covering Location Problem to Habitat Reserve Site Selection: A Review". In: *International Regional Science Review* 39.1, pp. 28–47.

Swisher, J. R., S. H. Jacobson, and E. Yücesan (2003). "Discrete-event simulation optimization using ranking, selection, and multiple comparison procedures: A survey". In: *ACM Transactions on Modeling and Computer Simulation (TOMACS)* 13.2, pp. 134–154.

Szechtman, R. and E. Yücesan (2016). "A Bayesian approach to feasibility determination". In: *2016 Winter Simulation Conference (WSC)*, pp. 782–790.

Taylor, M. F. J., P. S. Sattler, M. Evans, R. A. Fuller, J. E. M. Watson, and H. P. Possingham (2011). "What works for threatened species recovery? An empirical evaluation for Australia". In: *Biodiversity and Conservation* 20.4, pp. 767–777.

Treves, A., J. A. Langenberg, J. López-Bao, and M. F. Rabenhorst (2017). "Gray wolf mortality patterns in Wisconsin from 1979 to 2012." In: *Journal of Mammalogy* 98.1, pp. 17–32.

Tsai, S. C., C. Park, and M. H. Chang (2023). "Metamodel-based simulation optimization considering a single stochastic constraint". In: *Computers and Operations Research* 155, p. 106239.

Tsai, Y.-A., G. Pedrielli, L. Mathesen, Z. B. Zabinsky, H. Huang, A. Candelieri, and R. Perego (2018). "Stochastic Optimization For Feasibility Determination: An Application To Water Pump Operation In Water Distribution Network". In: *2018 Winter Simulation Conference (WSC)*, pp. 1945–1956.

Underhill, L. (1994). "Optimal and suboptimal reserve selection algorithms". In: *Biological Conservation* 70.1, pp. 85–87.

U.S. Fish and Wildlife Service (1987). *Northern Rocky Mountain Wolf Recovery Plan.* Tech. rep. Denver, Colorado: U.S. Fish and Wildlife Service.

– (2022). *2022 Gray Wolf Questions and Answers.*

Wang, Y., H. Önal, and Q. Fang (2018). "How large spatially-explicit optimal reserve design models can we solve now? An exploration of current models' computational efficiency". In: *Nature Conservation* 27, pp. 17–34.

Williams, J. C., C. S. ReVelle, and S. A. Levin (2005). "Spatial attributes and reserve design models: A review". In: *Environmental Modeling and Assessment* 10.3, pp. 163–181.

Wydeven, A. P., T. R. van Deelen, and E. Heske, eds. (2009). *Recovery of Gray Wolves in the Great Lakes Region of the United States : An Endangered Species Success Story.* New York, NY, US: Springer New York.

Yoon, M. and J. Bekker (2020). "Multi-objective simulation optimisation on discrete sets: a literature review". In: *International Journal of Operational Research* 39.3, pp. 364–405.

Zhou, S. (2023). "To What Extent Can Simulation Optimization be Used in Wildlife Reserve Design?" In: *Proceedings of the 2023 Winter Simulation Conference (Archive).*

Ed. by C. G. Corlu, S. R. Hunter, H. Lam, B. S. Onggo, J. Shortle, and B. Biller. Texas, USA: IEEE Press.

Zhou, S., D. Worthington, R. Williams, and L. Rhodes-Leader (2023). "To What Extent Can Simulation Optimisation be Used in Wildlife Conservation?" In: *Proceedings of the Operational Research Society Simulation Workshop 2023 (SW23)*. Ed. by C. Currie and L. Rhodes-Leader. The Operational Research Society. Southampton, UK, p. 285.

Zhou, Y., S. Andradóttir, S.-H. Kim, and C. Park (2022). "Finding Feasible Systems for Subjective Constraints Using Recycled Observations". In: *INFORMS Journal on Computing* 34.6, pp. 3080–3095.