Visual Search as a Queueing Process

Yiqi Li

Inaugural Dissertation
Submitted in partial fulfillment of the requirements for the degree
Doctor of Social Sciences in the
Graduate School of Economic and Social Sciences (GESS) at the
University of Mannheim

June 19, 2019

Thesis Defence:
August 26, 2019

Supervisors:
Prof. Dr. Edgar Erdfelder
Prof. Dr. Martin Schlather

Dean of the Faculty of Social Sciences:
Prof. Dr. Michael Diehl

Academic Director of the CDSS:
Prof. Dr. Thomas Gschwend

Thesis Reviewers:
Prof. Dr. Edgar Erdfelder
Prof. Dr. Martin Schlather
Prof. Dr. Thorsten Meiser

Examination Committee:
Prof. Dr. Arndt Bröder
Prof. Dr. Edgar Erdfelder
Prof. Dr. Martin Schlather
Prof. Dr. Thorsten Meiser

Abstract

A mathematical model accounting for both accuracy and response time (RT) data in standard visual search experiments on a distributional level has been developed. The model implements the conceptualization of Moore and Wolfe (2001) of a hybrid model, in which visual stimuli are selected by attention one after another and then identified in parallel. For the modeling of RT data on a distributional level, a finite-time queueing model with exponentially distributed interarrival times and service times, c parallel servers and a finite customer source is specified. It characterizes both the serial and parallel aspects of the visual search process by modeling it as a queueing system where stimuli enter serially and are processed in parallel at multiple servers. For the modeling of accuracy data, the queueing model is extended by integrating a mechanism that produces incorrect system responses. It differentiates between genuine processing errors and decision errors due to incomplete search. The observer is assumed to adapt a discrimination criterion for stimulus identification and a termination rule of search that both depend on set size to optimize search efficiency. The termination rule derived based on optimal foraging theory regulates the termination of the queueing process. Embedded in the queueing model, this mechanism accounts for both error rates and error RTs. Parameter estimation methods are developed using the maximum likelihood estimation (MLE) approach for accuracy-related parameters and the minimum distance estimation (MDE) approach based on Monte Carlo simulation for RT-related parameters. The complete model is fitted to two empirical data sets of two different search tasks. For both tasks, it fits the error rates precisely and the distributional RT data nicely, indicating a high explanatory power of the model.

C'est par la logique qu'on démontre, c'est par l'intuition qu'on invente.
Savoir critiquer est bon, savoir créer est mieux.
— *Jules Henri Poincaré,* Science et Méthode

Acknowledgment

Throughout the writing of this dissertation I have received a great deal of support and assistance. I would like to thank all those whose assistance brought me forward in the accomplishment of my goal.

I would first like to sincerely thank my supervisor, Prof. Dr. Erdfelder, without whom this dissertation project would not come into being at all. He led me to and kept me in scientific research and encouraged me to go for this ambitious, novel topic despite the uncertainties coming up in each stage of the dissertation. He always provided advice and support when I was confronted with difficulties. His insightful expertise inspired me and gave me directions.

I would like to show my warm thanks to my co-supervisor, Prof. Dr. Schlather, without whom the technical implementation would not be possible. He gave me a great deal of help when the project was in the largest difficulties. His valuable expertise and guidance helped me work out the technically most challenging part of the project. I have learned a great deal from him, not only skills and techniques, but also his rigorous working style and his passion for and dedication to science.

I wish to thank Prof. Dr. Döring, who advised me through the first two years of my dissertation project as the initial co-supervisor. He encouraged me to go for mathematical rigor and helped me explore the mathematically more desirable approach. I benefit a lot from his profound expertise in stochastics. I would like to acknowledge Prof. Dr. Meiser and Prof. Dr. Bröder, who kindly joint the examination committee on short notice and offered helpful comments on my work. I am grateful to Prof. Dr. Potthoff, who taught me probability theory, stochastic process and stochastic simulation; to Prof. Dr. Schmidt and Dr. Klein, who taught me analysis, awoke the passion for mathematics in me and

shaped my mathematical thinking. Without training in these courses I would not be able to develop the model on my own. I would like to pay my regards to Prof. Dr. Townsend, Dr. Wolfe, and Dr. Hawkins for their encouraging, enlightening and inspiring discussions; and to Prof. Dr. Schweickert for making me aware of an important reference.

I am also grateful to the GESS and the Research Training Group "Statistical Modeling of Complex Systems" for financial and infrastructural support.

A very special gratitude goes out to all friends and colleagues who supported me in various aspects along the way. Thanks to Anna-Lena, Christopher, Jonas, Dimitri, Nicholas and Daniel, who did the proofreading for parts of the dissertation, helped me solve technical problems, accompanied and encouraged me going though the most difficult stage of the writing (what a cracking place to work!); to Simone, Steven, Ludan and Peiwen for the vital emotional support!

Most of all, I am thankful to Bizhen, the wisest, strongest and most loving woman, who gave me life, raised me, and made me who I am. Thank you. This work is dedicated to you.

List of Symbols

This list displays the symbols that are used for deriving or expressing the equations in this thesis.

A	interarrival time of customers in the Kendall's notation for queueing systems
S	service time of servers in the Kendall's notation for queueing systems
c	number of parallel servers in the Kendall's notation for queueing systems
B	number of buffers in the Kendall's notation for queueing systems
K	population size of customers in the Kendall's notation for queueing systems
SD	queue discipline in the Kendall's notation for queueing systems
$\mathbb{N}$	set of natural numbers
T_0	beginning of the queueing process
T_i	instant at which the i-th customer arrives
A_i	interarrival time between the $(i-1)$-th and the i-th arrivals
S_i	service time of the i-th customer
k	set size (i.e., the number of the stimuli in the display)
T_p	time elapsed from the beginning of the arrival process until the queueing system responds "target-present"

T_a time elapsed from the beginning of the arrival process until the queuing system responds "target-absent"

T_{resn} residual time of a "no" response

T_{resy} residual time of a "yes" response

T_{res} common residual time without distinction between "no" and "yes" responses

t a precise time instant in the queueing process

λ mean interarrival rate

μ mean service rate

$Q(t)$ number of customers in the system at time t construed as a stochastic process

"T" | D event of mistaking an ordinary customer (i.e., distractor) for the target customer (i.e., target)

"D" | D complementary event of "T" | D, i.e., identifying an ordinary customer (distractor) correctly

"D" | T event of mistaking the target customer (target) for an ordinary customer (distractor)

"T" | T complementary event of "D" | T, i.e., identifying the target customer (target) correctly

p_1 probability of mistaking an ordinary customer (distractor) for the target customer (target)

p_2 the probability of mistaking the target customer (target) for an ordinary customer (distractor)

$\mathbb{P}$(false alarm) probability of responding "yes" on a target-absent trial

$\mathbb{P}$(miss) probability of responding "no" on a target-present trial

l number of customers that have been served and judged as "distractor" to initiate a "no" response on a trial

r ratio of l to set size k

SAD(l) event that l items have been searched and rejected

TA event that the trial is target-absent

TP event that the trial is target-present

s minimum posterior probability of a "no" response being correct given the event SAD(l), i.e., a confidence criterion

$N_k(l)$ random variable of the number of items examined until termination (regardless of the outcome response) under the giving-up policy of "stop and response 'yes' once the target is found, otherwise stop and response 'no' after examining l items" on a trial with set size k

$R_k(l)$ random variable of the number of correct responses under the giving-up policy of "stop and response 'yes' once the target is found, otherwise stop and response 'no' after examining l items" on a trial with set size k

$\mathbb{E}(X)$ expected value of a random variable X

$\gamma_k(l)$ mean reward rate under the giving-up policy of "stop and response 'yes' once the target is found, otherwise stop and response 'no' after examining l items" on a trial with set size k

z minimum acceptable accuracy

m number of set size levels realized in a visual search experiment

$R_{(k_1,\ldots,k_m)}(l_1,\ldots,l_m)$ total number of correct responses in an experiment with m set size levels $(k_1,\ldots,k_m)$ under the giving-up policy $(l_1,\ldots,l_m)$ on trials with the corresponding set size levels

$N_{(k_1,\ldots,k_m)}(l_1,\ldots,l_m)$ total number of items to search in an experiment with m set size levels $(k_1,\ldots,k_m)$ under the giving-up policy $(l_1,\ldots,l_m)$ on trials with the corresponding set size levels

β steepness of the reflected logistic curve used to approximate the theoretically optimal termination criterion

$-\frac{\alpha}{\beta}$ x-coordinate of the inflection point of the reflected logistic curve used to approximate the theoretically optimal termination criterion

$\tilde{m}$ smallest non-negative integer satisfying $mz - (m - \tilde{m} - 1) \geq 0.5$

a_1 coefficient of the monomial describing the power law relation between p_1 and k, i.e., $p_1 = a_1 k^{(} - b)$

b additive inverse of the exponent of k in the power law relation between p_1 and k, i.e., $p_1 = a_1 k^{(} - b)$; at the same time also exponent of k in the power law relation between p_2 and k, i.e., $p_2 = a_1 k^b$)

a_2 coefficient of the monomial describing the power law relation between k and p_2, i.e., $p_2 = a_1 k^b$

L stopping criterion modeled as a random variable

ΔL a Bernoulli random variable such that the probability of it taking the value 1 equals the decimal part of l

$\frac{1}{\lambda}$ mean interarrival time

$\frac{1}{\mu}$ mean service time of a single server

N number of simulation runs

w number of trials realized for each set size level in a balanced design

X_{tp} response on a target-present trial (tp), modeled as a Bernoulli random variable

π_{md} probability of responding “no” on a target-present trial (miss detection)

X_{ta} response on a target-absent trial (ta), modeled as a Bernoulli random variable

π_{fa} probability of responding “yes” on a target-absent trial (false alarm)

$y_{tp,k}$ observed number of "yes" responses on the target-present trials with set size k in a visual search experiment

$y_{ta,k}$ observed number of "yes" responses on the target-absent trials with set size k in a visual search experiment

$n_{tp,k}$ observed number of "no" responses on the target-present trials with set size k in a visual search experiment

$n_{ta,k}$ observed number of "no" responses on the target-absent trials with set size k in a visual search experiment

$L(\pi_{md,k_1}, \pi_{fa,k_1}, ..., \pi_{md,k_m}, \pi_{fa,k_m})$ likelihood of observing the response pattern $((y_{tp,k_1}, y_{ta,k_1}), ..., (y_{tp,k_m}, y_{ta,k_m}))$ in the entire experiment with set size levels $(k_1, ..., k_m)$

$l(\pi_{md,k_1}, \pi_{fa,k_1}, ..., \pi_{md,k_m}, \pi_{fa,k_m})$ log-likelihood of observing the response pattern $((y_{tp,k_1}, y_{ta,k_1}), ..., (y_{tp,k_m}, y_{ta,k_m}))$ in the entire experiment with set size levels $(k_1, ..., k_m)$

$\widetilde{l}(\alpha, \beta, a_1, a_2, b)$ log-likelihood function

Δt a (small) change in t

n number of customers in the system construed as a concrete number

λ_n effective arrival rate depending on n

μ_n effective departure rate depending on n

$\mathbf{Q}$ generator matrix for the birth-death process model of the Markovian queueing system

$P_n(t)$ probability of the number of customers in the system at t being n, given that no customers are in the system at the beginning of the queueing process

List of Algorithms

Contents

Chapter 1

Introduction

I am never content until I have constructed a mechanical model of what I am studying. If I succeed in making one, I understand; otherwise I do not.
— *William Thomson,*
Molecular Dynamics and the Wave Theory of Light

Looking for a predefined target among other objects is an important real-world task that human cognition has to deal with efficiently in everyday life. Imagine the scenario where the supermarket will be closing in fifteen minutes and you still have eight items on your shopping list that you urgently need but have not yet found, including your grandmother's favorite yogurt, a new bicycle multi tool your picky brother asked you to buy him, and tuna for your cat. Or imagine another scenario where you are looking for the stand offering the most genuine Thai snack in the alleys of Bangkok and suddenly realize that your little child is no longer in sight, disappearing in the huge crowd of strangers. In such cases, you probably just want to locate your target as soon as possible, whether it is a multi tool or a four-year old boy. How does the search process work so that we can meet the efficiency demands in different kinds of situations? During a visual search, can attention be distributed to several stimuli? Or can no more than one object be attended to at any time?

Despite a large amount of research in the last five decades, the serial/parallel debate of visual search still continues, with some empirical findings favoring one account and some the other (briefly reviewed in Section 2.2, cf. Bundesen & Habekost, 2004; Moore & Wolfe, 2001; Thornton & Gilden, 2007). Among the existing empirical findings, two directly motivated this research project. On the one hand, there are studies suggesting that it takes several hundred milliseconds to redirect attention from one stimulus to another (estimated at 200 – 500 ms, e.g., Duncan et al., 1994; Horowitz et al., 2009; Moore et al., 1996; Theeuwes et al., 2004). On the other hand, the analysis of response times (RT) as a function of the number of stimuli in the display (set size) indicates that they are processed at a rate of a few dozen milliseconds per stimulus (estimated at 20 – 60 ms/item, e.g., Treisman & Gelade, 1980; Wolfe, 1998b). If processing is strictly serial, then this should be the speed at which attention redirects from one stimulus to another, which contradicts the first finding. Moore and Wolfe (2001) proposed a hybrid model that can resolve the apparent conflict: Items are selected by attention in a serial manner at a rapid rate (of about 50 ms/item) but several items are analyzed in parallel after having gone through this attentional bottleneck, whereby the processing of each item can take several hundred milliseconds. In this way, such a mechanism can identify stimuli at a rapid rate, although the time required to identify every single stimulus can amount to 500 ms.

Although this conception is appealing because of its straightforwardness and simplicity in accommodating seemingly conflicting findings, it is rather informal and somewhat vague. There are many theoretical options and ambiguities with this idea and it remains unclear which predictions such an account makes uniquely under what conditions. Thus, it is difficult to test this conception empirically. One solution is to formalize it as a formal model that makes quantitative and testable predictions. Elaborating on the idea of Moore and Wolfe (2001), this dissertation formalizes and tests a formal stochastic model that implements their conception of a hybrid model.

This dissertation is divided into six chapters. In the first chapter, the relevant experimental approaches, theoretical concepts, accounts and models as well as empirical findings of previous studies are reviewed. In the second chapter, a formal stochastic model for the RTs on correct trials in visual search is developed.

At the beginning of the second chapter (Section 3.2), I demonstrate that queueing models have the desired properties described by Moore and Wolfe (2001), which make them conceptually suitable to model the cognitive processing during visual search. Along with the arguments for the appropriateness of queueing models, the major obstacles of applying queueing models to the study of visual search are described and the necessary tasks for a successful adaptation are outlined. Following this, I further explore the adaptability of queueing models to visual search by briefly reviewing the fundamentals of queueing theory, provide a mathematical formalization of queueing model of visual search (Section 3.3), and then elaborate the model predictions of RTs in visual search (Section 3.4). In the third chapter, the queueing model is extended by incorporating a mechanism that produces incorrect system responses so that it accounts for RTs and response accuracy simultaneously. The chapter starts with a rudimentary form of an error model based on the assumptions of imperfect processing and self-terminating and exhaustive search (Section 4.1.1). By showing the inconsistencies of the predictions of this rudimentary form with empirically observed error patterns, I conclude that the assumption of exhaustive search needs to be replaced by an assumption of premature termination (Section 4.2). The final accuracy model attributes errors to both incomplete search and imperfect processing, assuming a quasi-efficiency-maximizing termination policy of the observer (Section 4.3 to 4.5). In the fourth chapter, a simulation routine of the complete model is developed and implemented in R. The behavior of the extended queueing model is studied systematically using Monte Carlo simulation. This preliminary model assessment enables a better understanding of how changes of input variables and model assumptions influence the model predictions. In the fifth chapter, a parameter estimation procedure is developed in order to test the model empirically. Tailored to the extended queueing model, the parameter estimation procedure consists of the estimation of accuracy-related parameters as the first step and the estimation of RT-related parameters based on the estimated accuracy-related parameters as the second step. I first present the technical details of estimating accuracy-related parameters using Maximum-Likelihood Estimation in Section 6.1, and then the technical details of a hierarchical and iterative procedure for the estimation of RT-related parameters using Minimum Distance Estimation approach in Section

6.2. In the sixth chapter, the extended model is fitted to empirical data of previous studies. At the end of the thesis, the results of this dissertation are summarized and discussed.

The major achievement of this dissertation is a quantitative, testable model that accounts accuracy and RT data in visual search simultaneously and on a distributional level. It realizes a theoretical conceptualization that addresses a long-existing, fundamental question in attention research. Even though a hybrid model provides a theoretically appealing response to the enduring serial/parallel debate, we do not understand visual attention better than before as long as it remains a notion. On this matter, I share the view of William Thompson that a mechanical model is the way to truly understand a phenomena or a relation. By formalizing it as a formal mathematical model, I make the conceptualization of a hybrid model empirically testable. It can be tested empirically on a restrictive level because it makes predictions on the entire distribution of RTs. Similarly, theoretical concepts used in the construction of the mechanism accounting for error rates are quantified to express a hypothetical relation, for example, the mean reward rate from optimal foraging theory (Section 4.3.3). In fact, it was the analysis of quantitative predictions of different notions that led me to a novel account of understanding errors as a result of incomplete search and imperfect processing.

To ensure the transparency and replicability of the implementation, the development of the model and the corresponding simulation and parameter estimation methods are documented in an elaborate manner.

Chapter 2

Theoretical background

2.1 Visual search as an object of study

2.1.1 What is visual search and why study it?

Wolfe (2018) defines visual search as "the act of looking for something or a number of things" (p. 569). Visual search constitutes an important cognitive function of human beings and animals with highly developed visual system because a good search performance is essential for survival. Many activities that maintain the existence rely on visual search, such as collecting food, hunting for a prey, finding orientation in the environment, identifying enemies and threats. So do activities that build the basis of a society and of reproduction, such as finding the young or mates in the herd. Although the life of human beings in modern society has changed significantly compared to primitive or farming societies, the role of visual search is not essentially different than in ancient times. We may look for food in a supermarket instead of in a field, get necessities by shopping in the internet instead of collecting them in the nature, find orientation in big cities built with complex streets and signs seen only in civilization, and look for friends in a crowd. The objects we are dealing with and structure of the environment are different but the role of visual search as defined by Wolfe (2018) in fulfilling the basic needs has not changed. The function of visual search remains the same: finding a specific object and ignoring others, sometimes requiring filtering of more intense information.

As a basic cognitive performance, visual search has been attracting attention of numerous researchers. Understanding visual search helps us optimize search performance and/or efficiency in various contexts, identifying factors that influence the search outcomes and efficiency and guiding the design of an more performance facilitating environment or products. More importantly, the study of visual search does not only has applicational interests. Visual search as a phenomena and an experimental paradigm is sufficiently interesting and important for scientists to answer fundamental questions about visual attention. What is the nature of visual attention? What is the architecture of visual system? What are the mechanisms of visual processing? On a more general level, studying visual search helps us answer theoretical questions about human cognition, such as how the mind allocates cognitive resources.

2.1.2 The role of visual search in studying the time course of visual attention

Visual search is a specific topic and an experimental paradigm in the field of visual attention research. Research on visual attention attempts to answer fundamental questions, such as "how does attention work" and "where (or what) is the limitation of attention?" These questions motivated the development of various kinds of experimental paradigms. Basically, the allocation of attention can be viewed as a black box receiving and releasing information. To find out how the black box works, the investigator varies the task demands or the resources available and observe the performance as outcome. Ideally, the observed relations allow inference on how much resources are necessary to meet certain demands. However, finding out the amount of necessary resources that just match the demands imposed on the observer is far from trivial in empirical research. Any attempt of such inferences relies (at least implicitly) necessarily on some assumptions on the mechanisms of attention, which themselves belong to the objects that need to be examined empirically. This issue should become clearer if one takes a closer look at the existing experimental paradigms. Corresponding to the idea of varying either the demand or the available resource while keeping the other constant, existing experimental methods follow roughly two approaches.

One approach is to increase the total demand or workload unit by unit and observe the increment of resource expenditure to reach a certain performance level. The use of visual search as a research paradigm follows this concept. The demand is systematically manipulated by adding items to the set to search and the increment of resource expenditure is measured by RT. Thus the *speed* of processing constitutes the basis of inferences according to this approach. This leads to the classic slope analysis of the RT $\times$ set size function (see Section 2.3.1 and 2.3.2). However, the slope is not a direct measure of the time course of attentional shifts but rather an inference. The slope as an estimation of the time course of individual shifts of attention is only justified if visual search underlies a (stage of) strictly serial processing (see Section 2.2). In other words, inferring on the limitation of attention using the slope requires the assumption on a specific manner of the allocation of cognitive resources.

The other approach is to restrict the available cognitive resources while holding the processing demand or workload constant, and observe the reduction of performance (the outcome of the processing). Thus the *accuracy* of processing constitute the basis of inferences according to this approach. The attentional dwell time paradigm (see Section 2.5.2) and related paradigms (see Section 2.5.3) are representatives of this class. Restriction of processing capacity can be achieved by brief presentation of the stimulus to identify (i.e., presenting stimuli only for very short time period), requiring a response to another stimulus at the same time (i.e., using a dual task paradigm), or both. The study of attentional dwell time often adopts a dual task paradigm (see Section 2.5.2). Typically, two items need to be identified in such methods. One of them makes demand, taking up resources, and the impact on the identification of the other item is measured as an indicator for the interference. The processing capacity is restricted by shortening the temporal proximity of the first to the second item. The paradigms used to study attentional dwell time are explored in details in Section 2.5.

2.1.3 How has visual search been studied in laboratory?

As an important paradigm for understanding visual attention, visual search has been studied extensively (Wolfe, 1998a). Simplified and abstracted from the act of visual search in daily life, "classic search tasks involve search for a target in an

array of clearly individuated items presented on an otherwise blank background." (Wolfe, 2018, p. 574). Different from the visual field in a continuous, naturalistic scene, participants see isolated visual stimuli — usually abstract symbols and patterns — in a display in a typical visual search experiment. In each trial, the display contains either one or no occurrence[1] of a predefined target as well as a number of distractors (e.g., a single blue "T" target letter in twenty black and blue digits and letters). The task is to indicate as quickly as possible whether the target is present in the display. The stimuli in the display remain visible and unchanged until the participant makes a response. Figure 2.1 shows an example of the search display used in the experiments of Wolfe et al. (2010, see Section 7.1 for a brief description).

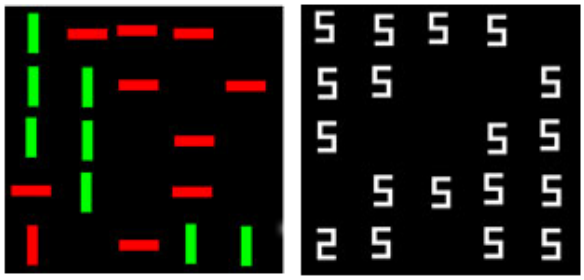

Figure 2.1: An example of the search displays used in Wolfe et al. (2010). From "Serial__Conj", by Visual Attention Lab, 2010, Retrieved from http://search.bwh.harvard.edu/new/data_set_files.html. Copyright 2019 by Visual Attention Lab.

As discussed in the last section, performance measures are the center of visual search paradigm and experiments. The most obvious, thus the most commonly used performance measures are behavioral measures such as accuracy and latency. Accuracy is measured by the proportion of correct responses in all responses and latency by the mean RT. As more advanced and sophisticated technologies become available, information beyond behavioral data can be recorded precisely as well. Among others, eye movement methods have become more convenient to apply and more popular. Although the idea of recording scan paths of fixations to reveal the process of search is appealing, making valid inferences on the deployment of attention based on eye movement data turns out to be difficult in

[1]This is referred to as standard visual search. There is a variant form using redundant or multiple targets, see e.g., Mordkoff et al. (1990), Pashler (1987).

practice. Firstly, the recorded data lags a few hundreds of milliseconds behind the deployment of attention. Secondly, fixations turn out to be neither necessary nor sufficient for the deployment of selective attention (see Wolfe, 2018, for detailed arguments). Other methods for non-behavioral measurement aim at recording brain activities. Electrophysiological techniques such as electroencephalography (EEG) and magnetoencephalography (MEG) measure the electromagnetic activity of the brain. The most useful measure for the study of visual search is visual event-related potentials (ERPs). Specific patterns have been found out to correlate with the deployment of attention, in particular the N2pc component (e.g., Woodman & Luck, 1999, see also 2.5). However, electrophysiological methods are hard to use on classic visual search tasks due to their special requirements on experimental setup and procedure to obtain meaningful results. Neuroimaging techniques, such as magnetic resonance imaging (MRI) and functional magnetic resonance imaging (fMRI) deliver structural or functional images indicating neural activity, yet their usage in tracking the concrete course of attention deployment process is limited due to their lack of spatiotemporal resolution (for detailed discussion see Wolfe, 2018). Moreover, employing non-behavioral methods comes with higher technical and financial demands in practice, which further restricts their usage. In sum, all of these types of measurement methods are used to investigate visual search, but the vast majority of the studies employing visual standard search paradigm use only behavioral measures.

Although visual search experiments often use arbitrary, isolated objects (e.g., lines with different orientations, geometrical figures, letters and so on) on blank background in order to control stimulus features and configurations precisely, mechanisms inferred from laboratory data are considered to play an important role in searching in real-world, continuous scenes (e.g., Wolfe, 1994b; Wolfe et al., 2011).

2.2 Serial vs. parallel accounts

As mentioned above, search efficiency (speed) is at the core of research on visual search. It is surprising that despite many constraints on our capacity of information processing, visual search has reached an efficiency that enables

coping with tasks in daily life successfully. What kind of mechanisms governs the allocation of attention such that the search is so efficient? Is it because we can deploy our attention to several objects at the same time? As an enduring debate in the visual search literature, the serial/parallel debate concerns the question of how attention is allocated among visual objects.

In daily life, we sometimes experience the search as so effortful that we have to check object by object to locate the target, for instance, looking for a ring in a drawer full of sundries. Sometimes we happen to spot the target with a rapidity that surprises even ourselves, for instance, locating the new album of our favorite singer among the thousands of CDs on the shelves of a store as if it had "popped out" on its own so that we did not need to check the CDs one by one. These two different experiences are respectively reflected by the serial and the parallel account, which make up two fundamental theoretical positions regarding the mechanism of object recognition. Both accounts have not only correspondence to introspection but also plausible theoretical rationales.

The notions of serial and parallel processing are rooted in the information processing approach and were transferred in the field of vision research in the 1960s (e.g., Eriksen, 1966; Eriksen & Spencer, 1969; Sperling, 1963; Sternberg, 1967, April). Since then, nearly a half-century has passed; the serial/parallel debate remains an unsolved issue in vision science despite intensive research in the past decades.

The serial and parallel accounts diverge on the question of whether accomplishing visual search tasks involves any object-by-object serial processing stage (Moore & Wolfe, 2001). The serial account holds that visual search contains at least one serial processing stage. It does not deny the existence of parallel processing stages. Rather, it claims that the visual system has a limited capacity, such that at some point in the processing it becomes impossible to process visual information in a parallel manner. That is, a bottleneck exists in the information-processing channel of the human system that requires the processing to reduce to one-by-one. This corresponds to the view of attention being a filter that picks out relevant information or a spotlight that directs to certain coordinates of a location map.

In contrast, the parallel account denies the notion of a mandatory serial selec-

tion of individual items at any level of processing. Although the parallel account also admits that humans have limited resources in information-processing, it views attentional selection as a resolution of these constraints by competition and cooperation between features and objects. Preliminary representations of visual objects compete for the constrained processing resources within the nervous system. By "winning" the competition, an object is recognized. Therefore, attention is an emergent property of such competitions rather than a "spotlight" that is directed to a certain location to integrate features.

2.2.1 Serial models

The major theoretical argument for the existence of a mandatory serial processing stage is that visual search essentially involves object recognition. To recognize an object, it appears necessary to link the visual representation to the representation of the object in memory. It is plausible that such linking involves the visual representation of only one object at any instant and thus requires a separate serial deployment of attention from object to object, selecting them to pass through the bottleneck. Through an item-by-item engagement of processing, stimuli are identified and classified as target or non-target. This is particularly the case in situations where the target is defined by a conjunction of features (e.g., black square in gray squares and black circles). Serial allocation of attention is needed because perceptions of single features have to be integrated correctly to identify an object.

2.2.1.1 Feature integration theory (FIT)

The most influential serial model is the feature integration theory proposed by Treisman and Gelade (1980). The core hypothesis of FIT is that visual perception consists of two functionally independent and sequential processing stages: a parallel feature encoding stage, followed by a serial feature binding stage.

In the first stage, basic visual features, such as color, size, shape and orientation, are registered independently and in parallel across the visual field. Feature detectors encode the presence of the corresponding features in separate feature maps. Within each map, the relational location information of the corresponding

feature, if present, is stored using the map's own coding system. In this way, processing in the first stage is assumed to be preattentive and automatic (i.e., without attention). However, there is no coordination of information across different feature maps at this early stage. Thus, it should be possible for a searcher to tell the presence of single features by inspecting the activity in the corresponding feature map (e.g., there is something red and there is something round). Yet the searcher may not be able to recognize an object (e.g., a red dot), because these separate registrations of individual features ("free-floating" features) do not form a coherent representation due to lack of coordination across different feature maps. To identify an object as a whole, features have to be bound together correctly, which requires attention according to FIT.

The integration of features takes place in the second stage. Here, focal attention is needed to function as the "glue" that coordinates activations across different feature maps. More precisely, focal attention is assumed to be a serial scan operating in a more general coordinate system, a master map. Moving within the master map, an attention window ("spotlight") recovers and combines the representations of the separate features at the current attended location via connections between the master map and the feature maps. In this way, an integrated percept of the object would be formed and the object would be identified. After the recognition, the attention proceeds to the next location to perform the integration there, if needed. In this way, processing in the second stage is assumed to be attentive and serial.

Although FIT can predict the patterns in a lot of empirical data, there are many findings that it is not able to explain (see Section 2.3). To accommodate these findings, Wolfe et al. (1989) developed the guided search model.

2.2.1.2 Guided search (GS)

The core of the original GS model (Wolfe, 1994a; Wolfe et al., 1989) is the claim that the preliminary processing outcomes guides the allocation of attention. In FIT, the parallel, preattentive stage and the serial, attentive stage are independent. FIT does not specify how the preattentive stage supports the localization of a given feature. Based on the basic structure of preattentive/attentive processing stages as in FIT, the GS model conceives a preattentive mechanism that uses the

information from the first stage to guide deployment of attention in the second stage.

The basic principle of this guidance mechanism is a combination of a bottom-up, stimulus-driven guidance to salient items and a top-down, schema-driven control reflecting the needs of the searcher. First, separate features of visual stimuli are registered and analyzed in parallel in the preattentive stage in the way FIT conceives. The outcomes of the preattentive processing are activation values for the features present in the display. These activation values are then delivered from different feature maps to the master map via broadly tuned "categorical" channels (e.g., "red", "yellow", "black", "small" and "big"). The bottom-up component of the activations responds to local contrast or salience of stimuli. It measures the differences between the value of a given location and the values of its neighborhood regarding the same feature. The top-down component of the activations responds to target-matching features, i.e., explicit task demands or implicit change in guidance. It weights or selects relevant channels from the broadly tuned channels so that they differentiate the target from the distractors as well as possible (e.g. "black" and "small"). These activation values are summed up across features at the same location. The sum gives an evaluation by the preattentive stage of how likely the object at a given location to be a target. Attention is then directed to the location with the highest total activation value. As soon as attention is deployed, the serial stage integrates different features to identify the object and then classifies it as target or non-target. Due to inherent noise, this location may contain a distractor. In this case, after the classification, attention is directed to the location with the next highest activation.

When guidance is possible in a task, search efficiency will be improved. For instance, for a small black "2" as target with distractors that are red, yellow or black and big or small digits, information from the channel "black" and "small" is enhanced. Compared with big black digits or small red digits, small black digits receive higher total activation from both channels, and therefore would be assigned higher priority. Although red digits and big digits may automatically have a relative high salience, these subsets of candidates may be ignored due to the inhibition of the respective channels.

The GS model has been successful in accommodating variations in search

performance, such as intermediate slope in conjunction search (see Section 2.3.1) and searching behavior within a certain subset of stimuli (see Section 4.2.2)

2.2.2 Parallel models

Parallel models rely on the notion that search efficiency is achieved by a unified mechanism of both selection and recognition that deals with several objects simultaneously. According to the parallel account, once the (multiple) stimuli enter the visual system, evidence of the identity of all objects accumulates over time until thresholds of identification as targets or non-targets are exceeded. There can be multiple levels of processing, but the principle of each level remains a parallel competition. Once an object is recognized, it is also selected, and vice versa. In this way, information from these objects can be processed simultaneously to classify them as target or non-target.

2.2.2.1 Attentional Engagement Theory (AET)

Duncan and Humphreys (1989, 1992) proposed the attentional engagement theory (AET) as response to FIT. AET regards the entire visual field as a continuous search surface, where object representations compete for access to visual short term working memory (VSTM). By entering VSTM, an object is selected and recognized. The competition is based on a matching mechanism that continuously cumulates evidence for the degree of similarity each stimulus shares with a "template of the target."

According to AET, stimulus objects are processed initially in a parallel stage of perceptual segmentation and analysis, which results in hierarchically structured representations across the visual array. These representations are then compared with an attentional template (i.e., a specification of the anticipated target) and gain or lose weight in the competition, depending on how well they match the template (top-down excitations and inhibitions). In this process, representations share their weights to the extent of their similarity. That is, they are linked by perceptual grouping based on their perceptual similarity and any change in weight for one would spread in parallel to the others (weight linkage). Therefore, search efficiency is determined by two factors: the target-distractor similarity

and the distractor-distractor similarity (bottom-up connections). If the target and the distractors are very similar (e.g., a rotated "T" in rotated "L"s, especially "L"s with 90 degree counterclockwise rotation are confusable with a "T" with 90 degree clockwise rotation), their representations would be all linked closely, resulting in similar weights and consequently inefficiency in determining the "winner". In contrast, if the target and the distractors are dissimilar (e.g., an "X" in "O"s), the target would gain weight quickly, resulting in an efficient search. On the other hand, if the distractors are very homogenous (e.g., an "X" in "O"s and "Q"s), they would all lose weight quickly because they are linked closely and rejected together due to the shared "unlike-features." In contrast, if the distractors are very heterogeneous (e.g., a "T" in all kinds of letters), they are linked loosely so that the inhibition of each single feature (e.g., round shape) is not sufficient to reject a large group of them (e.g. "O" and "Q" would lose weight, but not "K", "N" and so on). In this case, search would also be inefficient because additional iterations of rejection are required.

2.2.2.2 Theory of visual attention (TVA)

At the same time when AET was proposed, another influential parallel model emerged — Bundesen's (1990) theory of visual attention (TVA), a limited-capacity race model. In TVA, attentional selection and object recognition take place at the same time, by means of visual classification. That is, "the process of attentional selection is conceived as a parallel processing race among visual categorizations." (Bundesen & Habekost, 2004, p. 119), whereby a visual classification takes the form of "object x has feature i (or belong to category i)."

This process consists of two stages. In the first stage, the strength of the sensory evidence for object x belonging to category i is calculated (the η values, e.g., how well "C" and "G" matches the feature "round shape") by a massively parallel comparison. The η values are affected only by the objective perceptual properties of the visual field and the representation of the features in the long-term memory. The parallel processing in this stage is assumed to be unlimited. In the second stage, stimuli compete for selection into VSTM in a parallel stochastic race process. The storage capacity of VSTM is assumed to be limited to K (typically assumed 4) different objects, but unlimited with respect to the number of features. In other

words, when the memory space is available, the first K objects that first finish processing regarding the desired categorization (the winners of the race) become encoded into VSTM.

TVA views the classification of object x to category i as a result of the selection of x among objects and the selection of i among categories. It specifies two mechanisms of attention that affect the visual classification, i.e., the "competition". According to TVA, the "speed" of a certain visual classification in the "race" is jointly determined by the (objective) η value assigned to the object and two subjective values: the pertinence of the visual category and the bias of the categorization. The pertinence represents a mechanism called "filtering", which determines which objects are preferred. For example, if the target is red, then all red objects would have a high pertinence. The other subjective value bias of the categorization represents another mechanism called "pigeon-holding", which determines which category is preferred. For example, if the task is to find a red "S" among red and black letters and digits, then objects are more likely to be categorized regarding to alphanumeric identity instead of shape (in the sense that "S" and "5" have similar shape, so as "I" and "1"). TVA assumes independence between categorizations of different objects and between different categorizations (i.e., regarding different features) of the same object. Furthermore, it assumes that in most applications, once determined initially, the "speed" of each object remains constant during the stimulus presentation.

The serial/parallel debate simulated much theoretical elaboration of both serial and parallel models. Accordingly, there has been much work done to test these theoretical concepts and models empirically. Several experimental methods have been developed to distinguish between classes of models empirically with the focus on the distinction between serial and parallel accounts. Although these efforts has not led to conclusive evidence for or against either account so far, they provide useful experience and strategies for assessing this and related issues, pointing out the directions for subsequent research. The next section reviews the major empirical work that aimed at distinguishing between serial and parallel accounts using a latency measure (cf. Section2.1.3)

2.3 Major empirical findings regarding RT in visual search

As discussed in Section 2.1.2, to investigate the limitation of attention, visual search paradigm follows the idea of measuring the increment of the resource expenditure for increasing demands. Therefore, the relation between set size (i.e., the number of items in the search display) and RT has been studied extensively. As it is desirable that the increment of latency reflects the increment of resource expenditure, no further restriction is imposed on the cognitive capacity[2]. Since RT is considered as the indicator for efficiency under this assumption, its relation to set size under various experimental manipulations that ought to affect the search efficiency constitutes the primary empirical check for theories of visual search. A good model of visual search is expected to make predictions consistent with the findings regarding the RT $\times$ set size relation. Therefore, the patterns found in the RT $\times$ set size relation under various manipulations have become primary criteria for the assessment of the serial and parallel models reviewed in Section 2.2. These models (and other serial and parallel models) all make predictions on the time required to detect the presence or absence of the target under different conditions.

Whereas RT is the primary measure of interest, accuracy data has been attracted subordinate attention in research on visual search. Predicting the outcome of visual search seems not to be of primary interest in studies using visual search paradigm. The logic of the analysis of RT $\times$ set size treats performance as a control measure rather than a variable to predict. Even under the assumption of strictly serial processing, which allows inferences on the time course of individual shifts of attention, the claim that the increment of mean RT matches the necessary processing resources is only justified if the task is completed successfully. In theory, successful completion of the task means a perfect performance; in practice, slight error rates are tolerated. As a matter of fact, classic visual search tasks have been designed in a way that error rates could be kept low (e.g., under

[2]Note that the instruction "respond as fast as possible" serves as a measure intending to prevent an overestimation of the resource expenditure. A problem here is that the performance, i.e., the accuracy rate is not held constant when comparing the relation between demands and resource expenditure. See detailed discussions in Chapter 4, especially Section 4.2 and 4.6.

10%). Otherwise, the observed RT × set size relation will be questioned as an indicator for the search efficiency because speed could increase at cost of accuracy reduction. This issue is discussed in detail in Section 4.6. For this reason, accuracy data in visual search has been less frequently analyzed. Not only there have been infrequent attempts of predicting accuracy data, it has been rarely used as a hint for the development of theories of visual attention. Error rates have usually been seen as uninformative in respect of theory developing as they are considered not be able to differentiate between different theoretical accounts[3]. Most existing models make no explicit predictions regarding error rates.

To retain the focus of this chapter and a fluent presentation of the model development, the following sections discuss solely empirical findings regarding RT. Research work regarding accuracy is reviewed and discussed in Chapter 4.

2.3.1 RT × set size

Following the analysis method of Sternberg (1966) for memory search, the analysis of RT × set size relation was the earliest attempt to empirically examine whether visual search involves a serial component. The logic is simple: If visual search is serial, latency (reflected by the mean RT) should increase linearly with the set size because each added item requires an additional amount of time to be checked. In contrast, if visual search is parallel with *unlimited capacity*, RT should remain unchanged even if the set size becomes larger because added items can be checked simultaneously. Thus, the increase of the mean RT for every unit of increase in set size, i.e., the slope of the function relating the mean RT to set size, is taken as an index for the seriality/parallelity of the search process (cf. e.g., Treisman & Gelade, 1980; Treisman et al., 1977; Wolfe, 1998b).

As discussed last section, there is a consensus that the early stage of processing is parallel with unlimited capacity. The divergence concerns the processing stage that involves recognition. Therefore, the inference on the seriality/parallelity from the RT × set size slope depends on the type of the search task. If the target differs from all the distractors by a single basic feature (e.g., a red circle among

[3]This statement refers to differentiating theories based on the quantitative aspect of accuracy data, i.e., the error rates. Qualitative aspects, e.g., the types of the errors, have been used to test theories of visual attention. An example is the study of the phenomena illusory conjunction.

black figures), the outcome of the early parallel processing stage can already provide the diagnostic information for the presence or absence of the target — one must simply inspect the activity in this feature dimension. It follows that the completion of such a search task should be efficient and insensitive to the set size (i.e., characterized by a shallow RT × set size slope), regardless of the nature of the subsequent stage involving recognition. In contrast, if the target differs from the distractors in terms of a combination of features (e.g., a blue "L" among black "L"s and blue and black letters), the classification of an item as target or distraction requires a more elaborate processing. According to FIT and GS, this involves a serial integration processing search. It follows that search should be inefficient and more difficult with larger set size (i.e., characterized by a steep RT × set size slope). FIT labels the first kind of search tasks as feature search and the second as conjunction search. In empirical studies manipulating set size, shallow RT × set size slopes (mostly ≤ 5 ms/item) were found for feature search tasks and steeper RT × set size slopes (mostly between 5 and 25 ms/item) for conjunction search and other tasks (see e.g., Treisman, 1988; Treisman & Gelade, 1980; Wolfe, 1994a, 1998b). This has been initially seen as evidence for serial models, especially for FIT.

Furthermore, the ratio of the RT × set size slope on target-absent trials to that on target-present trials has also been regarded as relevant for the serial/parallel debate. It was argued that a strictly serial scan predict an approximate 2:1 target-absent to target-present slope ratio under the additional assumptions of exhaustive and self-terminating search. Exhaustive search means that the searcher checks all items to give a negative response on target-absent trials. Self-terminating search means that the search process ceases on a target-present trial once the target is found, resulting in a positive response. If the items are sampled in a random order with equal probability and without replacement, the target will be found after checking half the items in the display on average. This means that to give a positive response on a target-present trial, the searcher should need on average half the time of giving a negative response on a target-absent trial. In empirical studies, an approximate 2:1 ratio has been found (e.g., J. Palmer, 1995; Pashler, 1987; Treisman & Gormican, 1988; Wolfe, 1998b).

However, the research on the search-slope did not yield conclusive empirical

distinction between the serial account and the parallel account. First, neither a steep RT × set size slope in conjunction search nor a 2:1 ratio between target-absent and target-present slopes is a unique implication of serial models. A *limited-capacity* parallel model can predict a steep RT × set size slope, assuming that disjoint subsets of the entire set of stimuli in the display are sampled serially but stimuli within a subset are processed in parallel (e.g., Townsend, 1971, 1990; Townsend & Ashby, 1983; Townsend & Nozawa, 1995; Townsend & Wenger, 2004; Ward & McClelland, 1989). Similarly, based on the same assumptions of exhaustive and self-terminating search, a *limited-capacity* model can also predict a 2:1 ratio between target-absent and target-present slopes. Second, empirical data does not uniquely exhibit these two patterns. Patterns that deviate from these findings have also been observed in empirical studies. For example, some kinds of conjunction search were found to be completed very quickly (e.g., Nakayama & Silverman, 1986; Wolfe et al., 1989). Moreover, empirical RT × set size slopes do not cluster into two groups which can be labeled as "shallow" and "steep". An analysis of 1,000,000 visual search trials across different kinds of search tasks including feature search, conjunction search and other search tasks (Wolfe, 1998b) shows that the empirical slopes have not been subjected to a bimodal distribution, but rather display a continuum of search efficiency. Generally, the slopes fall into a range of 20 – 60 ms/item. Within each task type, the slopes vary rather continuously. The analysis also shows that a large proportion of empirical slope ratio is reliably larger than 2:1. On the other hand, the slope ratio is found to decrease and approximate 1:1 as the slopes themselves increase, that is, as the search task becomes more difficult (cf. e.g., Townsend & Roos, 1973). Although both serial and parallel models proposed theoretical explanations for these deviations (see e.g., Duncan & Humphreys, 1989; Wolfe et al., 1989), these attempts did not lead to a clear-cut conclusion.

2.3.2 RT × set size under additional manipulations

Following a similar logic, another cluster of studies investigated the effects of various manipulations on the RT × set size slope. Assume that a manipulation increases the processing time of a single item. If the processing involves a mandatory serial processing stage and the manipulation affects the processing

time in this stage, then the increased processing time for each single item that are processed should add to the latency. That is, the increase in the mean RT under the manipulation should be proportional to the set size (i.e., a constant ratio of the increase in the mean RT to set size is to expect)[4]. It follows that the effect of the manipulation on RT should interact with the set size: For larger sets, the difference in the mean RT under the conditions without and with manipulation should be larger than for smaller sets (over-additive effect of the manipulation). This implies a steeper RT $\times$ set size slope under a processing time increasing manipulation. In contrast, if search takes place in a purely parallel manner with *unlimited capacity*, although the processing time of each single item increases as well due to the manipulation, this increase should be reflected only once in the latency. It follows that the increase in the mean RT under the manipulation should be invariant to set size (additive effect of the manipulation). This implies that the RT $\times$ set size slopes under conditions with and without manipulation should be similar.

Empirical studies that used this approach yielded inconsistent results. For example, using stimulus quality manipulations (e.g., high-contrast vs. low contrast or no added noise vs. added noise), Pashler and Badgio (1985) observed additive effects of the manipulations, which indicates that the manipulations affected parallel processing. Egeth and Dagenbach (1991) adopted a slightly modified technique which presents only two stimuli in the display and manipulates the visual quality of the two stimuli independently (cf. Townsend & Nozawa, 1995). They reasoned that if the two stimuli had to be identified one after than the other, low visual quality of both stimuli should induce longer mean RT than low visual quality of one stimulus and high visual quality of the other. If the two stimuli could be identified simultaneously, there should be no difference between having only one and having two low-quality stimuli. In both case, high quality of both stimuli should lead to the shortest mean RT among the three conditions. The results of Egeth and Dagenbach (1991) supported parallel processing in

[4]Note that the argumentation from here on replaces "the number of items that are processed" by set size, i.e., "the number of items in the display". These two quantities equal each other only if the search is exhaustive. To ensure that all the items in the display have to be identified, Pashler and Badgio (1985, see next paragraph) used a so-called highest-digit task, in which participants were required to report the identity of the highest digit in the display.

some situations (e.g., "X" among "O"s and "T" among "L"s, or vice versa), but indicated the engagement of a serial process in other situations (e.g., a rotated "T" among rotated "L"s, or vice versa).

Apart from the inconsistent results, this approach has two major logical problems. First, *limited-capacity* parallel models are also able to predict the same over-additive effect of such manipulations, assuming that disjoint subsets of the entire set of stimuli in the display are sampled serially but stimuli within a subset are processed in parallel (cf. Section 2.3.1). Second, a serial processing stage could be a mandatory part of the entire search process, but such experimental manipulations exerted effects only on the parallel stage (e.g., the preattentive stage as postulated by FIT and GS) due to their perceptual character. In summary, this approach did not distinguish between serial and parallel models in an unambiguous manner, either.

2.4 Can RTs tell us more?

So far, it should be apparent that the major problem with using the slope of the RT × set size function to distinguish between serial and parallel accounts is that any pattern of the results is not a unique implication of either account. More specifically, processing that involves a mandatory serial stage and purely parallel processing with *limited capacity* can predict exactly the same pattern of RT × set size slope, regardless of an additional manipulation.

Another issue that adds doubt to the use of the RT × set size slope is the inconsistency of the patterns found in empirical studies. It makes the issue more difficult to deal with that the conditions under which the inconsistency occurred are not clear. Strict replications are rare among these studies on the search slope. Although a theorist may be able to adjust a model to accommodate new data, doing so without clear specifications of the constraints and careful experimentation evokes serious concerns from a epistemological perspective.

2.4.1 Previous work on analysis of RT distributions

If the analysis of simple RT × set size slope is not adequate to uncover the mechanisms underlying search processes, is a latency method still useful? One

important fact about the traditional RT × set size analysis is that it only uses the average or mean of the RT data as an estimation of the latency. If RT is regarded as a random variable, the RT × set size analysis only makes use of its first moment. From a statistical point of view, this act loses a lot of information contained in RTs, given the fact that little is known about the real distribution of this random variable. Therefore, an approach to this issue is to analyze the entire RT distribution. Although this kind of work is still rare, utilization of distributional information of RTs is getting more and more attention of researchers.

For example, based on a logic similar to Egeth and Dagenbach (1991), Sung (2008) presented four items in the display and manipulated the similarity of each distractor to the target stimulus (i.e., the display contained no, one, or two distractors that are more similar to the target than the other distractors). To examine whether the manipulations prolong the duration of the process of identification, the author checked whether the cumulative distribution functions (CDFs) of RTs under different manipulations satisfy stochastic dominance. The results supported a parallel processing in two experiments and a serial processing in one experiment.

Wolfe et al. (2010) investigated numerous aspects of three large data sets (about 112,000 trials in total, described in Section 7.1 briefly) on three visual search tasks, including the qualitative characteristics of the RT distributions. E. M. Palmer et al. (2011) fitted several functions (ex-Gaussian, ex-Wald, Gamma, and Weibull) to the RTs of the same data sets.

Although the pioneering work of E. M. Palmer et al. (2011), Sung (2008), Wolfe et al. (2010) addressed the importance of taking the entire RT distribution into account, no concrete quantitative model of RTs has been specified and tested on empirical data in their investigation.

2.4.2 A brief review of quantitative modeling of RT distributions

Bricolo et al. (2002) treated RT distributions of visual search explicitly as an object of modeling. They assumed a serial identification of stimuli and modeled the RT distribution as a convolution of a Gaussian and several exponential

distributions. The authors demonstrated via Monte Carlo simulations that their model can mimic the CDFs of the aggregated RTs over 13 subjects in a difficult (i.e., inefficient) visual search task with set size of 2, 4, 6 and 8. A parameter estimation method for their model was not presented.

Cousineau and Shiffrin (2004) investigated the termination of a search by analyzing the RTs of trials with different outcomes (hits, correct rejections and misses). The authors also assumed the search process to be a strictly serial scan of stimuli and the time of each scan to be a Weibull random variable. However, Cousineau and Shiffrin (2004) explicitly stated that the choice of a Weibull distribution was not critical because the specific shape of the RT distribution was not of interest. Rather, they focused on the comparison of RT distributions predicted by such a model assuming different termination rules. More specifically, the authors took the best-fitting Weibull distribution for RTs of each subject in the condition of set size of 1 as baseline and derived predictions for RT distributions for set size of 2 and 4 under different termination rules. These predictions were then compared with the empirical RT distribution of the subject in the corresponding conditions. The authors interpreted the results as evidence for premature termination by some of the subjects (for more details see Section 4.2.2).

Donkin and Shiffrin (2011) adopted a more elaborate modeling approach and proposed a computational model that accounts for visual search data at the level of RT distribution and choice probabilities. This model assumes a basic structure of a two-stage processing, similar to GS by Wolfe (1994a, 2007), Wolfe et al. (1989). The first stage is assumed to be an automatic parallel process (i.e., without attention) that operates across the entire display. It is modeled as a single accumulator that collects evidence regarding the presence and location of the target from all stimuli in the display in parallel. The results of the evidence accumulation process guides attention to the most likely location of the target. Then the selected item enters the second stage of processing and is compared to the mental representation of the target. This stage is assumed to be serial and modeled as two accumulators (differing from the accumulator in the first stage) that collect evidence for the item in focus being the target or being a distractor, respectively. The authors modeled the time of this evidence accumulation process based on the Linear Ballistic Accumulator (LBA) model (Brown & Heathcote,

2008). The model was fitted to the data presented in Cousineau and Shiffrin (2004) for each subject. The eleven model parameters were estimated successively in three steps[5] and by hand. The parameters that produced a visual agreement between the histograms of simulated and empirical RTs were chosen. The data fitting yield a satisfactory visual fit.

Moran et al. (2013) proposed another computational model of visual search that accounts for RT distributional data and error rates simultaneously — the competitive guided search (CGS) model. This model is also based on a structure of two-stage processing as specified by GS. It assumes that in the first, parallel stage that reflects certain guidance, stimuli compete for attention and are selected with a probability proportional to their salience value. The duration of this parallel stage, however, is assumed to be zero. The "winner" of the competition then goes through the second, serial stage of identification. The processing time of this stage is modeled as a Wald distributed random variable. The model assumes that the search process is terminated once a "quit unit" (an imaginary item) is selected (for more details, see Section 4.2.2) and attributes errors to premature termination and misexecution (motor errors). The model was fitted to the three data sets of Wolfe et al. (2010) on both individual and group (aggregated) levels using the quantile maximal probability method developed by Heathcote et al. (2002). The authors used AIC and BIC criteria to compare the fits of model variants (e.g., with vs. without guidance), though no concrete values of goodness-of-fit were reported. A precise assessment of the overall explanatory power of the model is difficult for the reader, since no direct comparison of the empirical and predicted RT distributions in the form of statistics or visualizations was presented.

To the best of my knowledge, the brief review in this section presents the majority of, if not all, computational models that accounts for RT data of visual search on a distributional level. Although there are other computational models of visual attention that fit empirical data well, such as the computational version of TVA by Kyllingsbæk (2006), they were not developed for visual search but other paradigms (e.g., for behavioral data from whole and partial report), hence may need adaptations to be applicable to visual search.

[5] A subset of parameters was estimated first and then fixed during the estimation of other parameters.

As discussed at the beginning of this section (2.4), the analysis of mean RT is not sufficient to uncover the mechanisms underlying visual search. Taking the entire RT distribution into account was one of the attempts of utilizing information beyond the mean RT. Another approach to this issue is to draw additional information from data collected using other experimental paradigms for visual attention. The findings regarding the so-called attentional dwell time have been often adduced in the serial/parallel debate (see Section 2.2) as evidence against serial models. In the next section, these findings are reviewed and discussed with the question of applicability across paradigms in mind.

2.5 The puzzle of attentional dwell time and related research

2.5.1 Temporal constraints of visual attention

To think of the applicability across paradigms critically, recall the two basic approaches of studying the time course of visual attention. As discussed in Section 2.1.2, attention limitations have been studied either by measuring latency without speed constraint or by measuring accuracy under speed constraints. The study of RT $\times$ set size slope in visual search is one of the methods that focus on latency, following the idea of measuring the incremental time cost of processing an additional stimulus. However, interpreting the slope as the rate at which attention *shifts* from object to object relies on the assumption of strictly serial processing.

In contrast, the approach measuring accuracy asks the question "what is the minimum resource to ensure the identification of a single object?" The idea is to measure identification accuracy under varying constraints on resources. A method to impose constraint on the resource is the so called dual task, i.e., requiring the identification of another object. By asking the identity of an object, attention is forced to be engaged on it. The cognitive resources available for the second object will be constrained. According to this logic, attention limitations can be inferred by measuring the interference of the first object on the second as a function of the degree of attentional demand for the first object. The paradigm

used to study attentional time is one of the many experimental methods in attention research that follow this approach.

2.5.2 Attentional dwell time

Duncan et al. (1994) defined the term attentional dwell time as the time period during which "an object that must be identified continues to occupy attentional capacity"(p. 313). They used a dual task paradigm as illustrated in Figure 2.2. Two alphanumeric characters were presented on each trial, each at one of the four locations (left, right, top, bottom) of the display in a random order and separated in time. One of the two characters was a green digit ("2" or "5") that only appeared at one of the horizontal locations. The other was a red letter ("L" or "T") that only appeared at one of the vertical locations. Either stimulus was presented for 40 to 65 ms, with a random temporal separation between 0 – 900 ms (from onset to onset, referred to as "stimulus onset asynchrony", SOA). The presentation of either stimulus was immediately followed by a masking pattern which lasted for 250 ms to interrupt further processing of the stimulus[6]. There were three conditions in the experiment: Participants were asked to report the identity of both characters, of only the digit, or of only the letter. The interference of the first target (T1) on the second (T2) was measure by the accuracy of the T2 identification. The major finding was that when both characters must be identified, the T2 identification was impaired the most severe at the SOA of 100 to 300 ms. The interference increased with increasing SOA and peaked at the SOA of about 200 ms. It than declined gradually and recovered at SOA of about 450 ms. When only one of the characters must be identified, the identification was independent of the SOA and similar to the performance for T1 in the dual task.

To enable a better comparability with visual search, Duncan et al. (1994) required the participants to indicate whether a predefined target is present or absent in a second experiment. They found similar patterns in the performance as the identification task. To differentiate the deployment of attention from shifts in space, the authors also conducted a variant of the design where both

[6]The technique of presenting a masking stimulus immediately after a brief (i.e., only presented for a very short time interval) visual stimulus (usually a target stimulus to be identified) is referred to as "backward masking."

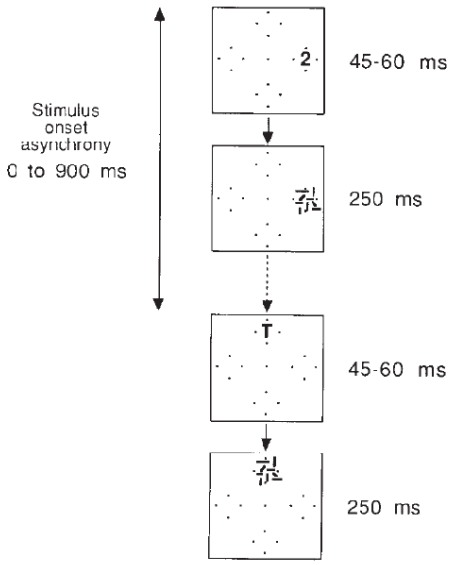

Figure 2.2: Sequence of frames on a single trial of the attentional time paradigm by Duncan et al. (1994). Reprinted from "Direct measurement of attentional dwell time in human vision" by J. Duncan, R. Ward, and K. Shapiro, 1994, *Nature*, *369*(6478), p. 313. Copyright 1994 by the Nature Publishing Group. Reprinted with permission.

stimuli were presented successively at fixation and obtained similar estimates of attentional dwell time. The authors drew the conclusion that identifying a single item occupies attentional capacity for several hundred milliseconds.

Moore et al. (1996) argued that the estimate of attentional dwell time using the dual task paradigm of Duncan et al. (1994) may not be applicable to explaining the processing time of stimuli in visual search paradigm. They questioned the use of masks following each target and pointed out that the masking makes the discrimination more difficult because stimuli in standard visual are typically not masked. In two experiments, Moore et al. (1996) employed the same dual task paradigm as Duncan et al. (1994) and manipulated the masking of T1. They compared the interference on T2 identification under the conditions a) with immediate T1 masking, b)with delayed and c) without T1 masking. The accuracy of T2 identification (measured by two alternative forced choice) was impaired at short SOAs in all conditions but the interference was most profound with T1

masking. Consistent with the findings of Duncan et al. (1994), the impairment first increased and peaked at the SOA of 200 ms in all conditions. However, the T2 identification under the condition without (or with delayed) T1 masking began recovering sooner (at SOA of 200 ms) than with immediate T1 masking (at SOA of 400 ms). At SOAs between 350 to 500 ms, the T2 accuracy was significantly higher without T1 masking than with T1 masking. The authors concluded that the estimate of attentional dwell time based on the data of Duncan et al. (1994) overestimated the time required to process a visual stimulus and suggested instead a value of approximately 200 ms based on their data.

Another finding of Moore et al. (1996) concerns the accuracy of T1 identification. Although no systematic dependence of the T1 accuracy on SOA was observed, it was significantly higher without backward masking (≥97% and ≥94%) than with backward masking (84 – 90% and 75 – 85%). This indicates that backward masking interfered with the identification of T1 backwards. This finding is not surprising insofar that brief masks following a brief target stimulus have been demonstrating the ability to reduce or eliminate the identification of the target stimulus in various applications (e.g., Breitmeyer et al., 2006).

Moore et al. (1996) also pointed out two reasons why the estimate of attention dwell time using the dual task paradigm of Duncan et al. (1994) does not necessarily correspond to the minimum required processing time of a stimulus in visual search. Both arguments concern the sequential presentation of stimuli in the dual task paradigm. First, shifts of attention used in the dual task paradigm may differ from those used in visual search. Shifts may be delayed when stimuli are presented asynchronously because it is not possible to plan a shift until the stimulus to attend to is displayed. In standard visual search, stimuli are presented simultaneously and remain visible and unchanged until a response is made, which enables the planning of a sequence of shifts on the onset of stimulus display. Second, a simultaneous presentation also enables exploiting the parallel processing capacity of the early processing stage, such that the average time attention dwells on a stimulus is shorter.

Note that under the notion of two stage processing, the attentional dwell time estimated using the dual task paradigm does not only include the time during which attention is deployed on an object, but also necessarily the time of

early parallel processing which is not supposed to involve attention. The same problem applies to the estimate based on the RT × set size slope. Even if visual search were strictly parallel and exhaustive, the slope would reflect in an ideal situation the marginal time cost for processing an item added to the search set. Yet the early parallel processing stage has to be completed for the identification of an object. Thus the corresponding time is necessarily included in the marginal time cost. However, under the assumption of parallel processing of the first stage, this time should be reflected only partially in the marginal time cost.

Theeuwes et al. (2004) also addressed the problem of applicability of the attentional dwell time paradigm to explaining findings from visual search. They argued that the attentional dwell time paradigm was a variant of rapid serial visual presentation (RSVP) from the attentional blink research (see next section), where the operation of attention in working memory and not of attention in perception plays a key role. The authors suggested using an experimental paradigm that was more similar to the situation of standard visual search to infer on the attentional dwell time. They adopted an essentially different experimental paradigm, as illustrated in Figure 2.3. The inference on attentional dwell time relies neither on measuring the increment time of processing additional items, nor the impairment of identification of a second target (accuracy under speed constraint). Rather, it is based on the assumption that the RT to a stimulus should be shorter when attention is at the same (or a nearer) location at its appearance than when it is deployed at another (or a farther) location at its appearance. More specifically, the task includes the identification of a letter stimulus and the detection of a probe stimulus and this dual task forces observer to switch attention serially from one location to the next. After the initialization of a trial, an arrow appeared in the center of the display, pointing to one of the four corners of the display. The arrow indicated the location to which attention had to be shifted and remained visible and unchanged until the end of the trial. After 400 ms of the appearance of the arrow in the center, the target display was presented. It contained four stimuli, each at one of the four corners. The target arrow was presented at the location the center arrow was pointing to and it pointed to the target letter itself. The target letter was either the letter “E” or the mirror-reversed “E”. The other two stimuli were a distractor arrow and a

distractor letter. Observer was required to identify the target letter. In this way, the identification of the target arrow was necessary for the identification of the target letter. That is, the processing of the target arrow must be completed before the start of the processing of the target letter to determine which item was the target letter. After a variable amount of time (50 to 650 ms), the target display was removed. On half of the trials, a small square probe was flashed for 33 ms at one of the four corners 50 ms after the removal of the target display, resulting in an SOA of 100 to 700 ms. Observer was required to press the space key as soon as possible once they detected the probe. At the end of the trial, observer indicated the identity of the target letter by pressing a key. The authors compared the mean RT to the probe presented at the target arrow location, at the target letter location and the distractor location depending on the SOAs. At short SOA (100 – 200 ms), probe RT at the first attended location was descriptively shorter than at the second attended location (not significant). At longer SOAs (300 – 500 ms), probe RT at the second attended location was significantly shorter. The authors inferred that attention was at the first location within the first 200 ms and at the second location after 300 ms. They concluded that the attentional dwell time should be around 250 ms.

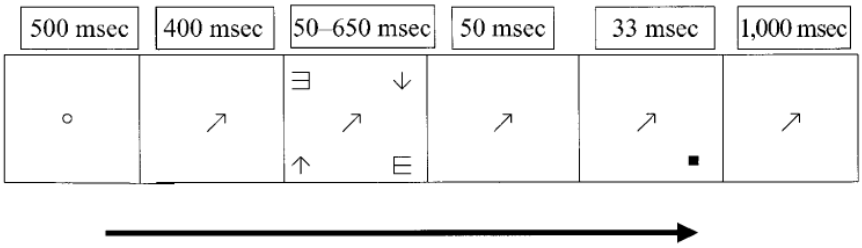

Figure 2.3: Sequence of frames on a probe-present trial in the experiment of Theeuwes et al. (2004). Reprinted from "A new estimation of the duration of attentional dwell time" by J. Theeuwes, R. Godijn and J. Pratt, 2004, *Psychonomic Bulletin and Review*, *11*(1), p. 61. Copyright 2004 by the Psychonomic Society. Reprinted with permission.

2.5.3 Related work

Although the term of attentional dwell time has been first used by Duncan et al. (1994), the study of inference or reduction in accuracy when two objects must be identified (conflict in resource) has a longer tradition than the construct of attentional dwell time. The phenomenon of interference of the identification of a first target with that of a second target has been studied with the aid of the technique of rapid serial visual presentation (RSVP; Potter & Levy, 1969). RSVP presents a stream of stimuli one after another with even lags in the same location of the screen. To estimate the time course of attention, two targets (e.g., red letters under black letters) are presented and the task is to report the identities of both targets. The positions of both targets in the stream are varied so that they are separated by differently long interval (e.g., Raymond et al., 1992; Shapiro et al., 1994). It is found that when the separation between two targets (stimulus onset asynchrony, SOA) is less than 500 ms, the identification of the second target is impaired; when the separation is more than 500 ms, both targets can be identified correctly (e.g., Broadbent & Broadbent, 1987; Chun & Potter, 1995; Reeves & Sperling, 1986). This phenomenon is referred to as attentional blink.

The simultaneous/sequential method developed by Shiffrin and Gardner (1972) compares the identification accuracy in two conditions. In the simultaneous condition, four characters were presented simultaneously in the display for 50 ms. In the sequential condition, the four characters were presented one after another (with no inter-stimulus interval), each for 50ms. in an equivalent position as the simultaneous condition. While each character was presented, the other three were masked (Experiment 1 and 3) or invisible (Experiment 2). The characters were “T”, “F”, “O” and T/F hybrid characters, whereby “T” and “F” were targets and each trial contained only one “T” or “F”. The task was to report which of “T” and “F” appeared. The idea is that if serial processing is necessary for identification, then a sequential presentation should have positive effect on the performance compared with a simultaneous presentation, which would not exist if identification involves only parallel processing.

The experiments of Shiffrin and Gardner (1972) indicated that the sequential presentation had no advantage over the simultaneous presentation. Using the highest-digit task, Pashler and Badgio (1985) obtained similar results. However,

the interpretation of the results of these two studies was more difficult than it appeared. First, both studies used a very small set size (four items) and relatively simple target-distractor discrimination. Fisher (1984) showed an advantage of the sequential presentation using a set size that is typically adopted in visual search experiments; Duncan (1987) and Kleiss and Lane (1986) also observed increased accuracy for the sequential presentation using complex target-distractor discrimination. Second, the absolute accuracy in the experiments of Pashler and Badgio (1985) and Shiffrin and Gardner (1972) was merely slightly above chance in both conditions (50% to 67%), which indicates a general identification difficulty. This is a strong hint that attention could not be reallocated to another position successfully within the time frame available in the experiments (50 ms). Therefore, the simultaneous/sequential method did not exclude the possibility of the engagement of a serial process.

2.6 The contradiction

This amount of hundreds milliseconds in theses studies is interpreted as the minimal time during which the distribution of processing resources to a certain location is maintained, i.e., the minimal time period in which attention is occupied once it is deployed to a location. This order of magnitude has been considered as inconsistent with the serial account (see Duncan et al., 1994; Ward et al., 1997). If processing involves a strictly serial processing stage, attention must be able to move from object to object at a very fast rate of a few dozen milliseconds According to an exhaustive review that included 1,000,000 trials from 2,500 sessions over a 10-year period (Wolfe, 1998b), the slopes of RT $\times$ set size functions are typically in the range of 20–30 ms/item for target-present trials and 40 – 50 ms/item for target-absent trials. All slopes from the empirical studies reviewed are less than 150 ms/item. This estimation is clearly less than the estimation of attentional dwell time.

Although it has not been discussed in the literature, I consider the findings regarding attentional dwell time incompatible with the parallel account as well. If attention can be deployed to several objects at the same time (even though this number may be limited according to limited-capacity parallel models), why

would the identification of the second target be affected by the SOA? A parallel processing allows the processing of a second object to start before that of the first one has finished. If this is the case, a switch from one to the other does not seem necessary. The interpretation that the impairment of the identification of T2 is due to the lack of attention itself contradicts the notion that attention can be distributed to T1 and T2 (and other stimuli) at the same time. Therefore, a parallel model is not able to explain the attentional dwell time findings without further assumptions.

2.7 An attempt to resolve the debate: an assembly line metaphor

Moore and Wolfe (2001) suggest that a hybrid model could explain the inconsistency between rapid processing rate and long attentional dwell time, and it could even accommodate other conflicting findings. They used a metaphor of an assembly line (later a car wash metaphor, Wolfe, 2003, 2007) to illustrate this idea. Analogous to cars on an assembly line, representations of the visual objects are selected and fed into the processing system (of the identification stage) serially, processed in parallel, and then "released" (identified) one-by-one. An assembly line may be able to deliver a car every ten minutes, but this does not mean that it takes only ten minutes to produce a car. Similarly, identifying items at a rate of 50 ms/item does not imply that it takes only 50 ms to process an item (until identification) completely. Such a mechanism accepts inputs and releases outputs in a serial manner, but its functionality is essentially parallel. That is, "items engage processing individually, but are analyzed, interpreted, and transformed, at least partially, at the same time" (Moore & Wolfe, 2001, p. 191). It is not a proposal for different processing stages being serial and parallel, but rather that visual processing can be both serial and parallel at the same time.

The assembly line metaphor conceptually provides a comprehensible architecture or mechanism of visual search that could accommodate seemingly inconsistent empirical findings to a larger extent than a serial or a parallel model. Nevertheless, it remains an idea that is not easy to examine in empirical studies

until it is formalized as a formal model that makes precise testable predictions. I propose queueing models as a suitable candidate for the formalization of the hybrid models by Moore and Wolfe (2001). In the next section, I discuss why it is reasonable to formalize them within the framework of queueing models and what has to be done to achieve this goal.

Chapter 3

A queueing model of visual search

A queueing system is a system in which customers arrive for a given service, wait in line if it cannot start immediately and leave after being served, whereby customers that enter and leave the queueing system are discrete units. Queues are formed commonly in everyday life, for instance, in traffic engineering, manufacture, telecommunication, computing, customer service and health care. Queueing theory is the discipline that studies the behavior of queueing systems mathematically, aiming at predicting the key variables of the structure and state of the queueing system of interest, such as expected queue length, expected waiting time, maximal waiting time and so on. It is an inextricable branch of probability theory from the theory side and operation research from the application side. Its results often serve as foundations for business decision making about the resource allocation to provide a service. After a century of development, queueing theory is thriving with mathematically sound models based on probability theory and statistics (e.g., Gross et al., 2008).

3.1 Pioneering work applying queueing theory to cognitive modeling

Queueing systems reflect certain order in resource allocation to demands that emerge randomly, which processing systems have to deal with constantly. The universality of queues in processing systems suggests that the understanding of

information processing systems cannot be complete without queueing theory. Besides the visual search process, it is conceivable that there are other cognitive processes displaying both serial and parallel properties which can be described by queueing systems. However, the application of queueing models to study human cognition as an information processing system is rare. So far, there are only several works attempting to outline a general framework of information processing of human cognition by queueing or queueing-network models.

Miller (1993) proposed a queue-series model for information processing in a general context. It conceives the processing of a stimulus with multiple components as a travel through a series of processing stages. One stage can process at most one component at a time and passes the component to the next stage when it finishes processing it. If a subsequent stage is still busy processing a component when the preceding one has finished and released another components, they have to wait in a line. All components have to be processed by all stages in the given order to enables an observable response to a single stimulus. Miller's (1993) model has in common with my model that the processing of a certain amount of elements has to be completed to initiate a response and that the processing of subsequent elements has to wait if the processing capacity is occupied by the processing of preceding element. But the units being processed are components of a single stimulus in his model and single stimuli in my model (see next section), and I assume only one stage that requires queueing (i.e., the identification). Moreover, Miller's (1993) model was not specified for visual search task (or for any specific empirical paradigm). It is an abstract model and was not fitted to empirical data. Thus it is dealing with a different question and a different structure.

Extending the work of Miller (1993), Liu (1993, 1996, 2013) proposed queueing network models as a tool to model human performance. Instead of a straight line as in the queue-series model of Miller (1993), queueing network models construct networks of processors in which information are transmitted and transformed as flows, with delays due to necessary waiting in the course. Queueing network models are able to predict the *mean performance* for specific multitasking tasks, as Wu and Liu (2007, 2008a, 2008b) demonstrated for driver performance, transcription typing and multitasking. Similar to the queue-series

model, queueing network models are dealing with different structures than the standard visual search task.

Fisher (1982, 1984) first proposed two model with a queueing structure for visual search, called time-dependent limited-channel model and steady-state limited-channel model, respectively. A description of these two models involves termini of queueing theory, which are explained in the following sections. To facilitate a better understanding, the similarities and differences between Fisher's (1982, 1984) steady-state limited-channel model and my model are explained in greater detail at the end of this Chapter. The steady-state limited-channel model was applied to explain the *mean* RT in the standard visual search paradigm (Fisher, 1982). The predictions for various set size levels (1 to 6) were fairly close to the empirical mean RT. The steady-state limited-channel model was also applied to explain the accuracy rate in simultaneous/sequential method (Fisher, 1984, see Section 2.5.3) and yielded good fits.

3.2 Visions and obstacles

Because the current research question deals with selective attention on the object level, that is, stimuli are seen as individual information carriers entering the visual system, a reasonable approach is to model the problem with discrete information flows. The assembly line metaphor naturally suggests the connection of the problem to the kind of problems described by queueing systems. Upon the stimulus onset of a trial, the signals of individual stimuli in the display start being received by the retina, going through the early visual processing stage and arrive in the stage of identification. Selected individually and serially into the processing system, stimuli in the display can be seen as customers arriving and waiting — if necessary — for a service (i.e., being recognized) in queues. Accordingly, the hypothetical parallel processing channels can be seen as stations delivering the desired service. The outcome of the service is then a classification of the customer as target or non-target. After serving a certain amount of customers, the observer makes a decision about the target presence in the display based on the service outcomes. The result of this decision corresponds to the response we observe, and the time elapsed from stimulus onset until the decision is observed

corresponds to the RT we measure. Obviously, a queueing process matches the conception of a hybrid model.

To achieve the goal of developing computational models that implement the conception of Moore and Wolfe (2001), we need to set up a system of assumptions that are applicable in the context of visual search, and derive computable relations from queueing theory. Specifically, the computational modeling deals with three major problems: 1) specification of a model; 2) estimation of the model parameters; 3) testing whether the models explain empirical data adequately. Though the conceptual connection between the hybrid models and queueing models is intuitively obvious, modeling visual search with queueing models is challenging. There are four major obstacles.

First, the central task of queueing theory is the study of the steady state of a queueing system. In typical applications, the key variables of interest regard to the steady state (also referred to as equilibrium) of a queue, i.e., when the probability of it being in any state does not depend on the specific instant. If the equilibrium exists, it describes the limit behavior of a queueing system. In other words, queueing systems that have an equilibrium will approach the steady state as it continues operating. A general result of queueing theory is that not-overloading is a necessary condition for possessing the equilibrium (Gross et al., 2008). Assuming no denial of customer entry, if the traffic intensity exceeds the service capacity the system can provide, the queue will get larger and larger and the queue size will never settle down. This is a situation which should be prevented in industrial applications. Hence, for industrial applications, queueing systems are usually designed or modified in a way that service requests can be processed eventually under normal conditions, which often comes along with the existence of a steady state. Another important aspect of the property of having a steady state is that such queueing systems are mathematically much easier to handle. Consequently, the existing results of queueing theory are mostly applicable to situations where the queueing system has been in operation for sufficiently long time.

A direct application of these results in the context of visual search is questionable because it is not clear whether (and when) the "queues" in visual processing would ever reach the steady state before they stop. They clearly have different

stopping rules than queues that are studied thoroughly in customer service and engineering. It seems even more plausible that they would not reach the steady state before they stop because the queue is assumed to be formed upon the stimulus onset, i.e., starts afresh at the beginning of each trial, and the number of stimuli is usually not large enough to keep the queue operating for a long time. Moreover, since the early visual processing stage is considered very efficient, and the identification stage much slower, the queueing system might be overloaded during its operation so that the condition for the equilibrium is not fulfilled.

Second, in most cases, the customer population is considered large enough that it can be treated as infinite whereas in visual search, the number of stimuli to be searched is finite and each stimulus is assumed to be searched only once at the most (but see Section 3.4.2). In queueing theory, the case of finite customer population has been studied only in the case of recurrent service demand of the customers, i.e., customers that have received service will always come into the queueing system again. Queueing models dealing with finite customer population assume that every customer goes through two alternating phases, not needing the service and being in need of the service, so that he or she will enter the queueing system again sometime after being served. In this way, the queue will not end in finite time. In contrast, in the context of visual search, the queue must terminate to initiate a response and the termination is crucial.

Third, on the empirical side, there are large constraints on the variables that can be empirically observed in visual search experiments. Traditionally, methods for statistical inferences in queueing theory, e.g., parameter estimation, have been developed under the assumption that interarrival time and service time data (see Section 3.3.2 for more details) can be directly observed. Not only the characteristics of the incoming stream, such as calls, cars and requests, can be determined via statistical methods, but also other useful variables can be measured, such as the time each customer spends in the queueing system, the time between successive departures and so on. In contrast, the data one can collect with the aid of the currently available techniques and paradigms in visual search experiments are merely the RT and the response on each single trial. As will be discussed in Section 3.3.3 and 4.5, the RT is assumed to correspond to the time required to serve a certain amount of customers, and the response to

the system response made at this time. This means that the observable of the dynamic course of the queueing process that this amount of customers have gone through reduces to one single variable — the RT, while the amount of customers involved is inferred by the system response observed. As far as I know, inferring the system parameters from such highly reduced observations has rarely been studied in the queueing literature. Thus, it is necessary to adapt general queueing models and derive model predictions that are appropriate for the RT and accuracy data with additional assumptions (see Section 3.3.3 and 4.5 for details).

For these three reasons, established models and results in queueing theory cannot be applied to visual search directly. It requires adaptation and development of queuing models and approaches that suit the current context and purpose.

A fourth issue is not about adaptation but rather about a proper modeling choice. On the one hand, queueing models are a large model family consisting of diverse members which can behave in very different ways. On the other hand, there are only indirect and limited empirical results suggesting the adequacy or inadequacy of certain model assumptions. There are many theoretical options and open questions. For example, Moore and Wolfe (2001) raised following questions: Can the same item be processed by different processors at the same time? Does the processing of one item influence that of the other? Must items remain in the same sequence as how they were loaded in the system? Given the current state of empirical research, it seems impossible to answer any of these questions satisfactorily. Therefore, the model specification is based mainly on rational considerations.

To tackle these four issues, the first step is to develop a queueing model that describes hypothetical relationships between variables that can be reflected in the data measured in a visual search experiment. I start with setting up a system of model assumptions specific for visual search and provide a formal definition of the specified model.

3.3 Model assumptions and model specification

Modeling begins necessarily with abstracting and precisionizing the modeling problem. Thus the empirical settings in which data is collected should be clarified first (Section 3.3.1). Then the question of how the variables of interest are represented by the model should be answered. To achieve this, general formal definitions for the critical characteristics of a queueing system are provided, first without reference to the context of visual search (Section 3.3.2). Applying these concepts to visual search, model assumptions clarify the general conditions of the modeling and the correspondence of model variables to empirical variables, (Section 3.3.3). Based on these assumptions, a queueing model is specified for the modeling problem (Section 3.3.4).

3.3.1 Basic empirical settings

The experiment in which the data is collected should use a standard visual search task paradigm (see Section 2.1.3). That is, participants search for a predefined target in the display and the task is to indicate whether the target is present or absent. Stimuli are presented simultaneously in the display and remain visible and unchanged until a response is made (no speed constraints). The variables of interest are accuracy (percentage of correct responses) and RTs. This chapter focuses on modeling RTs of correct responses. The model specified in this chapter is extended to account for error rates and RTs of incorrect responses in Chapter 4.

The experimental design should include two factors: target presence and set size. For the sake of simplicity, a balanced design is assumed, i.e., the target is present on 50% of the trials and each set size level appears with equal probability. Throughout this thesis, data collected under a specific combination of target presence and set size is referred to as a sub-data set.

Several ideal conditions could simplify the interpretation of the parameters estimated from an empirical data set. A unique stochastic process is assumed to characterize the visual information processing that leads to a response in visual search. Ideally, the parameters associated with this stochastic process do not depend on intrapersonal and interpersonal variations in a data set (or a sub-data set). In other words, the observations in a data set can be seen as realizations of

the same stochastic process. Several consistency conditions of the experimental settings make this prerequisite plausible. For example, the data set is collected from an individual participant (not from a group); The search task uses the same (type) of stimuli; the target is uniquely defined for at least a large number of trials (e.g., a block); the participant is cannot skip trials in a systematic manner. However, these conditions are neither sufficient nor necessary. It is possible that a participant demonstrates large inter-trial variability (due to exercise, fatigue or boredom) on the same task. It is also possible that visual processing is so universal that interpersonal variations have only negligible influences on the parameters that characterize it.

Most of the theories on visual attention do not address the issue of interpersonal variations (e.g., Bundesen, 1990; Treisman & Gelade, 1980; Wolfe, 1994a; Wolfe et al., 1989). The analysis of empirical data of visual attention and visual processing has been taking place on both individual and group level regardless of paradigm(e.g., Duncan et al., 1994; E. M. Palmer et al., 2011; Wolfe, 1998b; Wolfe et al., 2010). To my knowledge, there is not obvious evidence for the concern that analyzing aggregated data leads to essentially different results, as long as the data was collected for the same task. Therefore, the model specification and the derivation of predictions assume the empirical data collected from different individuals for the same task as realizations of the same stochastic process.

3.3.2 Elements of a queueing system

A one-stage queueing process is fully determined when six characteristics of the queueing system are known or specified. These elements are conventionally denoted by Kendall's notation in the form of $A/S/c/B/K/SD$ (e.g., Bhat, 2015): interarrival time of customers (A), service time of servers (S), the number of parallel service channels (c), the waiting room capacity, i.e., the number of buffers (B), the population size of customers (K) and the queue discipline (SD). They describe a queueing system and its behavior adequately. The interarrival time and the service time reflect the two sources of randomness; the other three elements reflect how the system is structured.

3.3.2.1 Interarrival time A

Let us denote the beginning of the queueing process by T_0. For a queueing process with no customer inside the system at T_0, it is also the beginning of the arrival process. The incoming stream is a sequence of customers arriving for service. It can be described by the sequence of instants $T_1, T_2, T_3, \ldots$ at which the customers $1, 2, 3, \ldots$ arrive. The arrival pattern is fully characterized by the sequence of times between successive arrivals, i.e., the interarrival times, defined as $A_i = T_i - T_{i-1}$, for $i \in \mathbb{N}$, assuming $T_0 = 0$. For the modeling, it is necessary to specify the probability distribution(s) of the stochastic process A_i as well as their stochastic dependence. If the probability distribution does not depend on time, the arrival pattern is stationary. The simplest arrival pattern is to have independent and identically distributed (i.i.d.) interarrival times A_i.

Alternatively, the arrival pattern can by characterized by the mean arrival rate, i.e., the average number of customers that arrive within a time unit. This characterization is useful when the arrival process is a Poisson process (see Section 3.3.3). In this case, the interarrival times are exponentially distributed with λ, which equals the mean arrival rate.

In most applications, arrivals do not happen simultaneously, and this is the mathematically simpler case. But it is possible for customers to arrive in groups (bulk or batch arrivals).

3.3.2.2 Service pattern (S and c)

The service pattern is described by the service time S and the number of parallel servers c together, representing how efficient the service can be delivered. Similar to the interarrival time, the service time is also a sequence of random variables S_i, indicating the duration of the service for the i-th customer, ordered by the arrival. Much of the discussions regarding the interarrival time also applies to the service time, although the arrival process and the service process are generally assumed to be mutually independent. Most important is the probability distribution of the service times at each single server, which is i.i.d. in the simplest case. The service mechanisms of parallel channels are usually assumed to operate independently of each other. Likewise, the service rate at a single server is an

alternative characterization for independent parallel servers with i.i.d. service times. The number of servers that are able to provide the service independently in parallel is assumed to be time independent.

Generally, a queueing system is conceived in such a way that one customer is served at a time by a given server, but there are also situations where customers may be served simultaneously by the same server, such as passengers boarding a plane, a computer with parallel processors, and so on. If both arrival and service are bulked, then the entire process is purely parallel, corresponding to the parallel account in visual search literature.

3.3.2.3 Characteristics relevant to the structure (B, K and SD)

The waiting room capacity B refers to the maximum number of customers that are allowed to wait in the queue. In many applications, there is no obvious upper bound to the queue and arrived customers do not leave the queue before getting the service. In this case, the capacity considered as infinite. If the length of the queue has a limit, no further customer is allowed to enter when the queue is at its limit. The incoming customers are forced to leave without getting a service until there is space available again.

Likewise, the customer population K is also a positive integer that can be seen as infinite if it is large enough or renewable. As mentioned in the last section, there are situations where the customer population is finite but each customer will enter the queueing system recurrently. The difference between a finite source with recurrent requests and an infinite source due to the renewability is that the arrival pattern of the customers from a finite source depends on how many customers are inside of (and consequently, also on the number of customers outside of) the system, whereas this dependence does not exist for infinite source[1]. In both cases, the queueing process is infinite in the sense that it goes on for

[1] An example for a finite source queue is a group of machines that require repair service when they become inoperative. When the defect happens to each machine independently and constantly, the more machines in the queue (waiting for repair or being repaired), the fewer machines are there that can become inoperative. Consequently, the arrival rate is lower when there are more customers in queue. In contrast, a post office in New York during the business hours shows an example for the cases in which a queue with an infinite source is an appropriate model.

infinite time (as a stochastic process). In contrast, the queueing model of visual search should be truly finite in the sense that the queueing (search) process terminates to initiate a response.

The queue discipline SD refers to the way in which customers standing in line are selected for service. The simplest and most common queue discipline is first come first served (FCFS). There are other disciplines such as last come first served (as in the case of using an elevator), selected by priority and so on.

In general, if we treat the queuing system as a black box and observe only the incoming and outgoing stream, we may see a quite irregular pattern with large variations in the time intervals between departures. The behavior of the queue depends largely on the ratio of the arrival rate to the *effective* service rate (i.e., service rate of a single server multiplied by the number of servers). This ratio is called traffic intensity. A fundamental result of queueing theory regards how to use the traffic intensity as an indicator for the long-term behavior of the queue (Gross et al., 2008). If it is larger than one, then there are more customers arriving than the system can serve in a time unit. The queue will get longer and longer. If it equals one, a steady state still does not exist unless the queue is deterministic. If the traffic intensity is smaller than one, then the queue will reach a steady state eventually. But even in this case, some customers might be delayed by waiting in the line because randomness can lead to periods in which either too many servers are free or all servers are busy.

Figure 3.1 illustrates a clip of a single-line, multiserver queueing system at a specific time point. A server is getting free due to the departure of the $j + 2$ customer and the next customer (who has arrived as the i-th at an earlier time) in line will be assigned to this server.

3.3.3 Basic assumptions

The model class of the queueing model is restricted to multiserver queueing systems fed by a single line. This consideration is based on a well-known result of the queueing theory that a single-line queue is always more efficient than a multiline queue, keeping all other conditions equal (see e.g., Gross et al., 2008). Given the fact that efficient visual search is essential for survival (see discussion in Section 2.1.1), it is reasonable to assume that human beings have evolved a

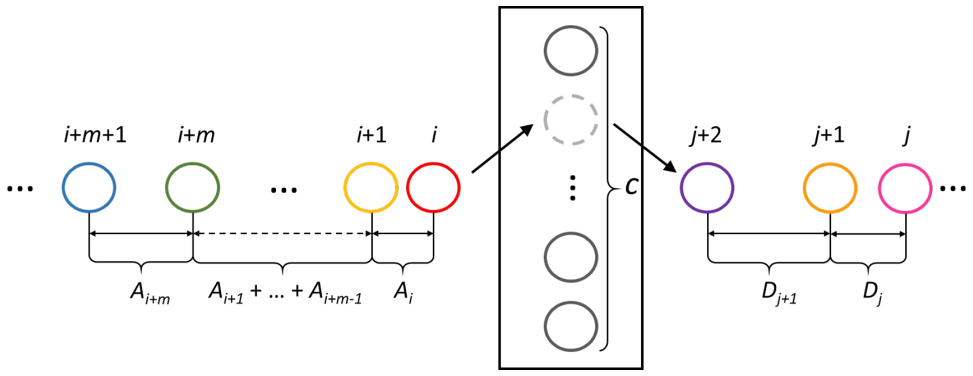

Figure 3.1: An illustration of a single-line, multiserver queueing system in operation at a specific moment in time. The colored circles represent customers, the gray circles busy servers and the dashed circles the free server.

structure that leads to a more efficient visual processing system. A single-line, multiserver queueing system also fits the conceptualization of Moore and Wolfe (2001) that a bottleneck is followed by multiple parallel processors which integrate the feature representations.

Before specifying a model from the class of single-line, multiserver queueing systems, we should clarify the basic settings and consider the question of which variables of a queueing system the empirically observable data correspond to. The first question to clarify is how elements of a single-line, multiserver queueing process match those of the process of visual search. The statement that stimuli as customers are selected serially and then processed in parallel describes a mechanism, but the connections of such a mechanism to existing theoretical concepts in visual processing are not clear yet. What does it mean that a stimulus "arrives", "is assigned" to a server and "leaves" the system?

First of all, the basic structure of visual processing is assumed to consist of a preattentive stage and an attentive stage, as in FIT (Treisman & Gelade, 1980) and GS (Wolfe, 1994a, 2007; Wolfe et al., 1989). The preattentive stage delivers feature fragments of the visual stimulus that are potentially sufficient to form a coherent object representation but not yet integrated to do so. The integration of the feature fragments, i.e., the identification of the object as a coherent representation is assumed to occur only in the attentive stage. In accordance with the notion that the preattentive stage comprises the processing before an attentional bottleneck, the arrival process is assumed to correspond to the preattentive stage. The

attentional bottleneck itself is represented by the single-line in the model. The integration or identification after the selection by attention (going through the bottleneck) is assumed to correspond to the service process. My model focuses on a mechanism on the level of how cognitive processing resources are allocated to multiple stimuli. It does not specify any mechanism of processing *within* a stage, e.g., how features are extracted and transformed, or how the feature fragments are integrated and compared to representations in memory.

Due to its hypothesized large capacity, the preattentive stage should hardly require resource allocation, so that it can be considered as massively parallel processing. Nevertheless, even a massive parallel processing stage takes time. That is, from stimulus onset to the moment when the feature fragments are ready to be integrated for the attentive stage, there is a time interval. The *arrival* of a stimulus in the queue is defined as the instant when all the signals of the features necessary to reconstruct a representation of this stimulus have gone through the preattentive stage. Even though features of different stimuli can be processed in parallel by the corresponding channels, it is unlikely that the arrival of all stimuli happens exactly synchronously. The mind is not a clockwork. The signals of features must first be received by the retina, processed, and then passed on to the responsible neurons for subsequent processing. The transformation and transmission of signals through different channels are rarely deterministic to lead to a uniform arrival pattern. There can be additional factors such as saccades that cause delays. Thus, a certain asynchrony in the arrivals of different stimuli is expected. The time intervals between arrivals of stimuli due to asynchrony correspond to the interarrival times.

In contrast to the preattentive stage, the attentive stage requires resource allocation. The fragments that belong to the same object are integrated into a coherent representation, which will be linked to the object representation in memory so that it becomes recognized. Even though the materials of an object are ready, there may be no resources available at the moment. This corresponds to the subjective experience that one perceives the existence of all the letters on this page but does not recognize them at once. The *assignment* of a stimulus to a server is defined as the moment when attention is deployed to this specific object. The assignment is assume to be instantaneous — at the exact moment of

assignment, the integration or identification begins. The process of integration, which leads to the identification of the object, corresponds to the service at the server. Once the identification is completed, i.e., a classification of this object as target or non-target is made, the service is finished and the stimulus is *released* from the system. In this way, the duration of the identification of a stimulus corresponds to its service time. For the sake of simplicity, it is assumed that there is enough inhibition of return (will be discussed in Section 3.4.2) so that released stimuli will not enter the queue again.

In this notion, an item may not be able to enter the attentive stage immediately after going through the preattentive stage. Waiting in the queue reflects the case where there is no attentional capacity available upon its arrival.

A dynamic view of the processing of visual stimuli from the perspective of the model is displayed in Figure 3.2, 3.3, 3.4 and 3.5. Assuming that the observer simply search all items in the display without making any responses, these figures illustrate the development of a complete process of identifying the four items in the display at eleven specific instances. The illustrations should be understood as symbolic and not to be interpreted as how visual information is actually transformed. For example, the red dot with a label x_1, y_1 in the processing channel means simply that the color signal of the red vertical bar with the coordinate x_1, y_1 has been received and conveyed for subsequent processing. Figure 3.2 illustrates the states of visual information at three specific moments: the stimulus onset, a time point between the stimulus onset and the first arrival, and the first arrival. Similarly, Figure 3.3 illustrates the development until the third arrival, Figure 3.4 the development until the second departure, and Figure 3.5 the last two departures.

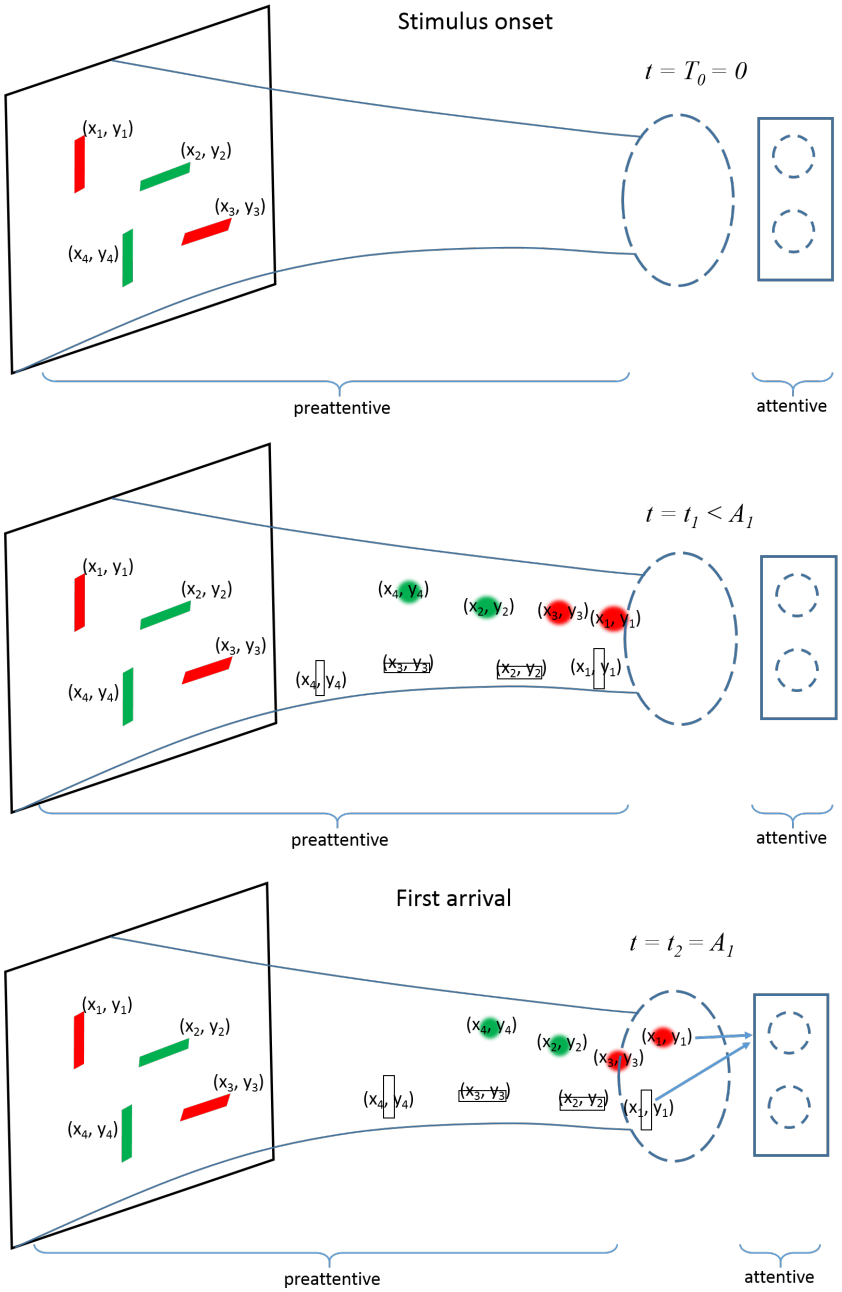

Figure 3.2: The states of visual information in the processing system at the moment of stimulus onset (top), a later time point (middle) and the first arrival (bottom). The "floating" features are going through the processing channels of the feature map. The coordinate information they carry indicates the visual object they belong uniquely. The time required to finish the preattentive processing varies for each "piece" of feature representation. At the moment when all feature representations of a stimulus have gone through the preattentive stage, it "arrives" in the queue.

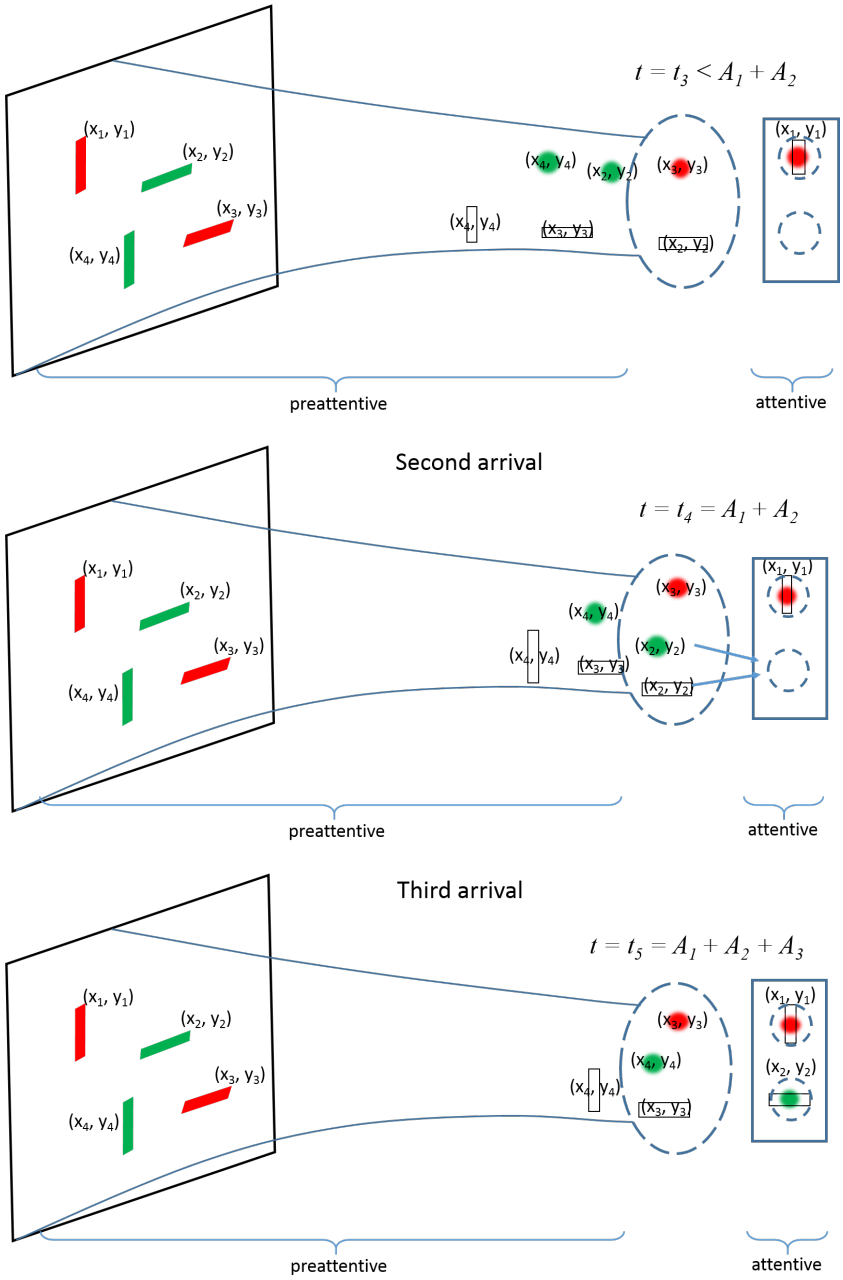

Figure 3.3: The states of visual information in the processing system at a moment before the second arrival (top), at the moment of the second (middle) and the third arrival (bottom). At t_3, because the color information of the green horizontal bar and the orientation information of the red horizontal bar are still under preattentive processing, neither of them is in the queue. Thus no assignment to the free server occurs at t_3.

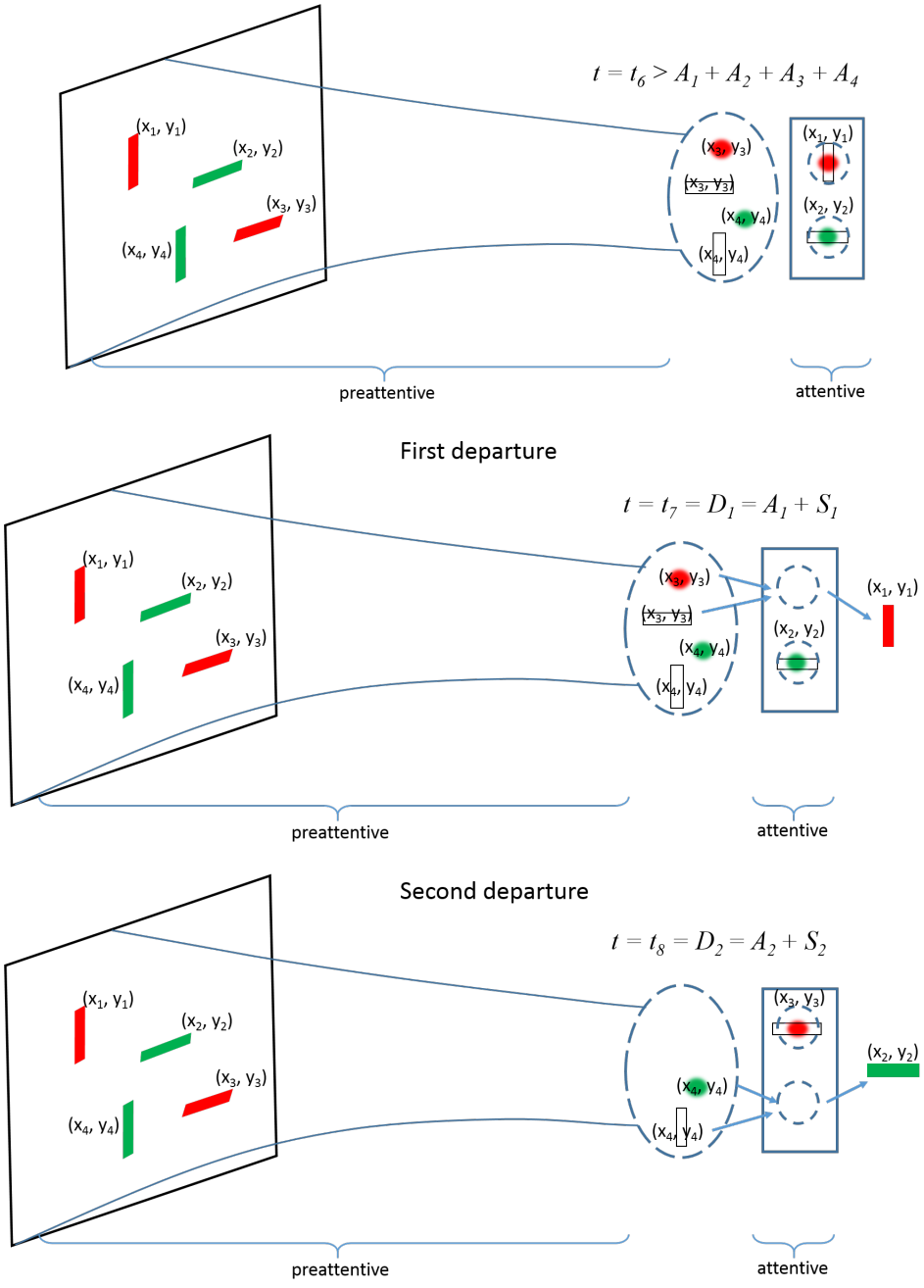

Figure 3.4: The states of visual information in the processing system at a moment after the fourth arrival (top), at the moment of the first (middle) and the second departure (bottom). At t_6, no server is available, so that the red horizontal bar and the green vertical bar have to wait in the queue although they have finished the preattentive stage. At t_7, a server becomes free. The feature representations carrying the label (x_3, y_3) arrived earlier than the others, thus they are assigned to the free server.

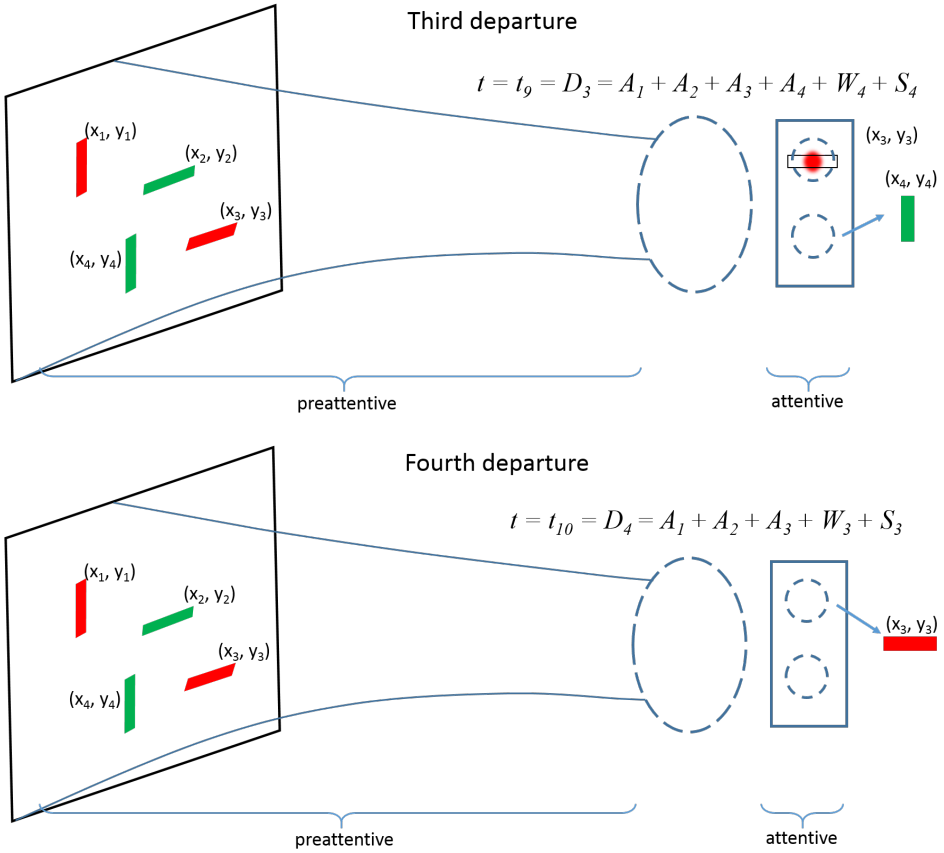

Figure 3.5: The states of visual information in the processing system at the moment of the third (top) and the fourth departure (bottom). Although the green vertical bar was assigned to a server later than the red horizontal bar, it finishes the service earlier due to shorter service time.

After clarifying the correspondence of a queueing system to visual processing from a process view, let us focus on the question of which variable of the model corresponds to which empirically observable variable. As discussed in Section 2.1.3, empirical quantities that can be observed in a visual search experiment are typically accuracy rates and RTs, which are also the objects of modeling in this dissertation (see Section 3.3.1). Queuing theory concerns only the temporal issues of queueing systems. It does not make predictions on the service outcome. Thus, modeling accuracy data is beyond queueing theory. On the other hand, in the current case, the outcome of the identification is important because it

does not only lead to observable accuracy data, but also determine whether the operation of the queue should continue. This problem is tackled in Chapter 4. For the discussions in this chapter, let us leave the incorrect responses out of consideration for a moment. In case of a correct response, the RT should reflect the time course of the queueing operation given error-free processing. Although the queueing process in the current context is truly finite (see Section 3.3.2), the six critical characteristics of a queue do not depend on when the queue terminates. Thus, we can specify a queueing model for the RT of correct responses.

For the sake of simplicity, the residual time, i.e., the time component in the RT that is not spend on collecting and analyzing visual information, such as motor reaction time, is modeled as an additive constant. The following discussion regarding the queueing process omits this residual time because its contribution is considered as a shift in the RT distribution.

The stimulus onset is assumed to be the starting point (denoted by T_0) of the queueing process, which is the instant when the arrival process begins. On a target-present trial, it appears plausible to assume that the correct response ("yes") is only initiated when the target is found. Since this stopping criterion is linked to a specific customer, it is necessary to differentiate two customer types: target and non-target. Then the critical question is on which position the target customer appears in the queue. If the preattentive processing stage does not favor any stimuli, the sequence of arrival should be random. The simplest assumption is that the target customer appears in any position with equal probability. Apparently, this assumption may be incorrect because it does not take any efficiency optimizing mechanisms of the preattentive stage into consideration, such as the bottom-up and top-down activation in GS (Wolfe, 1994a, 2007; Wolfe et al., 1989). On the other hand, it is difficult to derive a general relation regarding this issue without further theoretical specification of the preattentive stage. The simplicity of its implementation makes a equal probability assumption the modeling choice. Thus, the RT on a target-present trial is assumed to correspond to the time that has elapsed until the service of a target customer has finished, who appears in any position of the queue with equal probability. On a target-absent trial, an assumption of exhaustive search is provisionally adopted. That is, to ensure that the decision is correct, the response

is made after searching all stimuli in the display (see, however, Chapter 4 for a revision of this assumption). Thus the RT on a target-absent trial is assumed to correspond to the time that has elapsed until the service of all stimuli in the display has finished.

Denoting the set size by k, the assumptions regarding the RTs of *correct* responses are therefore:

- The RT on a target-present trial corresponds to the time elapsed from the beginning of the arrival process until the queueing system finishes serving a target customer who appears in any position of the queue with equal probability $\frac{1}{k}$, which is denoted by T_p, plus a residual time T_{resy};
- The RT on a target-absent trial corresponds to the time elapsed from the beginning of the arrival process until the queuing system finishes serving all k customers, which is denoted by T_a, plus a residual time T_{resn}.

In this way, RTs in visual search are predicted by the time elapsed until a specific customer (the target customer or the last customer) leaves the system. In queueing theory, the derivation of this time is related to the study of departure process, or output distribution. That is, how can we describe the pattern of time intervals between departures? As discussed in Section (3.2), it is not yet possible to directly observe the time intervals between departures in the context of visual search. We have to determine these times indirectly. The derivation requires a specification of the queueing model.

3.3.4 Model specification

To specify a queueing model, the six characteristics described in Section 3.3.2 need to be known or specified. It is reasonable to make assumptions that are conceptually compatible with the situation of visual search whenever possible and to choose the simplest option if little information is available about the appropriateness of any option.

3.3.4.1 Interarrival times and service times

Although an asynchrony in the arrivals is plausible since stochastic processing time is common in signal processing systems, little is known about the time intervals between the arrivals of stimuli so far. Similarly, there is no clear empirical evidence supporting a specific probability distribution of the time required for feature integration.

Existing models for visual search (especially computational models) that involve the identification of individual stimuli are mostly based on a concrete model of the identification stage. For example, GS (Wolfe, 1994a, 2007; Wolfe et al., 1989) bases the discrimination on Signal Detection Theory (SDT; Green & Swets, 1966; cf. Cameron et al., 2004; J. Palmer et al., 2000; Verghese, 2001); the CGS model (Moran et al., 2013) on an accumulation model; Donkin and Shiffrin (2011) on the Linear Ballistic Accumulator model by Brown and Heathcote (2008). Under such models, the distribution of the identification times (corresponding to the service time in my model) may be derived. It is of course desirable to have a concrete model for the identification stage, yet the distribution derived from such a commonly used model (e.g., Wald distribution) is usually too complex as the service time distribution for a queueing model. As mentioned in the previous section, my model does not specify any concrete mechanism within either of the two processing stages. Hence the assumptions on interarrival time and service time are not restricted to a probability distribution derived from a specific mechanism.

Therefore, I opt for simpler assumptions. Under the assumptions of independent and identically distributed exponential interarrival times and service times, the queueing model has a particularly simple form: The number of customers in the queue at time t as a random process is a continuous-time Markov chain. Restrictive as it might seem, the assumption of a exponential distribution turns out to be able to explain various temporal phenomena in nature adequately, and it is especially useful in the contexts where queueing models are applied. Another advantage of exponential distributions is that only one parameter is sufficient to characterize the probability distribution.

Specifically, I assume that

$$A_i \sim \text{Exp}(\lambda), \text{ and } S_i \sim \text{Exp}(\mu)$$

where A_i are the interarrival times and S_i the service times, as defined in Section 3.3.2, and λ and μ the rate parameter, respectively. In case of infinite customer source, λ and μ equal the interarival rate and the service rate, respectively. Queueing models with independent, exponentially distributed interarrival times and service times are called Markovian queueing models.

3.3.4.2 Customer population

One special characteristic of the application of a queueing model in the context of visual search is the finite customer population with non-recurrent demand, as discussed in Section 3.2. The reason is that the number of stimuli to be searched is finite and each stimulus is assumed to be searched only once at most. The assumption of finite customer source has an important impact on the arrival pattern. If the queue has an infinite customer source, the assumption of exponentially distributed interarrival times with a rate parameter λ is equivalent to the statement that the arrival process is a Poisson counting process with rate λ. Without an infinite customer source, the arrival process cannot be a Poisson process because the number of incoming customers has an upper limit.

In queueing theory, the arrival pattern of a finite source queue with recurrent demand has the property that the *effective* arrival rate is proportional to the number of customers outside the system, i.e., those who can potentially arrive in the future (see Appendix A for more details). For my model, i.e., a finite source queue with non-recurrent demand, a similar characterization of the arrival pattern is assumed. Specifically, the *effective* arrival rate[2] is assumed to be proportional to the number of customers that have not been in the system (i.e., in the queue or at a server) yet. This means that the mean interarrival time of the next arrival is inversely proportional to the number of customers that have not yet arrived.

[2]Strictly speaking, the term "rate" should not be used here because the process is not stationary. The "effective arrival rate" here is to understand as the probability that an arrival occurs in the next infinitesimal time interval, see Appendix A.

This characterization is assumed because such a arrival pattern is consistent with the theoretical concept of a preattentive stage. A parallel processing without capacity limits has the property that the processing time does not depend on the amount of inputs. In other words, for a fixed time interval, its (mean) outputs are proportional to its (mean) inputs. The specified arrival pattern as a result of the preattentive stage has exactly this property: The probability of a stimulus coming out of the preattentive stage in the next infinitesimal time interval is proportional to the number of stimuli that have not yet finished this stage by the current time.

3.3.4.3 Number of servers, capacity and queue discipline

Regarding the other three characteristics, there are also many theoretical options. Although empirical phenomena are usually very complex, it still appears advisable to start with models with the simplest options. Hence, I assume that the processing of stimuli by different processors is independent of each other (independent parallel service channels). The next question is how many parallel service channels are appropriate. Because the attentive stage involves the identification of visual objects, many theorists believe that the representations in this stage requires memory. Even parallel models assume a limited processing capacity. For example, TVA by Bundesen (1990) assumes that only four items at the most will be selected in the short term memory for further examination in parallel (final stage of the competition). However, clear empirical evidence on this issue is still lacking. Although a concrete number cannot be determined, it is very unlikely to be large (e.g., more than ten). The number of parallel servers is treated as a model parameter to be estimated, denoted by c. The number four may provide a good starting value.

The waiting room capacity concerns the question of whether there is a limit on the number of customers allowed in the system. Since the most theorists agree that the preattentive processing does not have a capacity limit, there appears to be no reason why any subsequently arriving stimuli will be discarded from the queue once a certain number of items are already in the system. Nevertheless, in the context of visual search, if there is a limit on the number of stimuli that have gone through the preattentive stage but not yet been attended to, then it is likely to be temporal fashion rather than quantitative. It seems more plausible

that such fragments of representation can co-exist in a large amount but the existence does not last long. That is, the customers in the queue may not remain in the queue after a certain amount of time. However, this kind of limit relies on the notion that the fragments of representation require memory to exist. But stimuli in a standard visual search task are constantly present before a response is made, which makes the memory of the physical features of the stimuli seemingly unnecessary. Therefore, the simpler assumption of unlimited waiting room capacity is adopted. For a given set size k, because there can be at most as many customers in the system as the population size , this is equivalent to the queue with a waiting room capacity of k.

As to the queue discipline, the question is whether to include prioritization. Prioritizing is plausible and part of some theories, e.g., GS (Wolfe, 1994a, 2007; Wolfe et al., 1989). Certain types of prioritization upon arrival can be equivalently modeled by prioritization of the incoming sequence, for example, the more similar an item to the target, the more likely it enters the queue in the first positions. I decide to reserve prioritizing as a theoretically convincing but more complex option and here assume a "first come, first served" rule (the next customer waiting in the queue will be assigned to any free server) instead.

In this way, the models are constrained to the kind of Markovian queueing model denoted by $M/M/c/\infty/k/FCFS$ (queueing model with exponentially distributed interarrival times and service times, c servers, unlimited waiting room capacity, k customers and default rule "first come, first served").

3.4 Deriving RT predictions

3.4.1 States and transitions between states

The next step is to derive the time required until the departure of the target customer (target-present trials) or all customers (target-absent trials). As discussed in Section 3.3.3, in queueing theory, this question requires the study of departure process, or output distribution. For queues in equilibrium, the properties of the departure process can be derived independent of a specific arrival or service. But the queueing model in the context of visual search will not operate in equilibrium (see Section 3.2), which makes the analysis of the departure process very difficult.

The fundament of deriving queueing-related variables is the study of the states of the queueing system. The state at any specific time t is the number of customers in the system at time t, including the customers in line and those at the servers. This is a stochastic process denoted by $Q(t)$. The behavior of the queueing system is fully characterized by $Q(t)$ (e.g., Bhat, 2015; Gross et al., 2008). Under the assumption of exponentially distributed interarrival times and service times as specified in the last section, $Q(t)$ is a continuous-time Markov chain with discrete states.

A transition is a change of the system from one state to another. When (and only when) customers enter or leave the system, transitions between the states occur. Under the assumption of independent, exponentially distributed interarrival times and service times, it can be shown that at any specific instant, no more than one customer will enter or leave the system, regardless of the customer population and the number of servers (see for example Bhat, 2015, for the proof). This means that the state transitions are of only two types: increasing $Q(t)$ by one (birth) and decreasing it by one (death). This makes the queueing model a special kind of continuous Markov chain called birth-death process.

By this notion, to derive the time required to serve a certain amount of customers (corresponding to the target-absent RT), say, k customers, is equivalent to derive the probability distribution of the random variable of the departure time of the k-th customer. This corresponds to the time when the Markov chain has the k-th jump downwards, regardless of the order in which customers leave[3]. To derive the departure time of a specific customer (corresponding to the target-present RT), say, the i-th incoming customer, is equivalent to derive the probability distribution of the random variable of the departure time of *the same* customer[4]. This corresponds to the time when the Markov chain has the jump downwards that associate to this customer[5]. In both cases, all finite combinations

[3]Note that customers do not necessarily leave in the order in which they have arrived due to the large variance in service times.

[4]Note that the same random variable gives the sum of time intervals between departures before the departure of this customer. However, even if the probability distributions of the time intervals between departures are known, it is not sufficient to derive the probability distribution of their sum because they are not stochastically independent.

[5]The i-th arrival and the i-th departure do not necessarily associate to the same customer.

of "ups" and "downs" that can lead to the event of interest have to be studied. That is, how the system changes from one state to another.

Hence, the analysis of the behavior of a $M/M/c/\infty/k/FCFS$ queueing process is equivalent to the analysis of the corresponding continuous-time Markov chain $Q(t)$ (e.g., Gross et al., 2008). In the terminology of Markov chains, this is a case with discrete state space $(0, \ldots, k \in \mathbb{N}_0)$ and continuous parameter space $(\lambda, \mu \in \mathbb{R}^+)$. The analysis techniques appropriate for such Markov processes involve two key steps: the determination of the infinitesimal transition rates, and the solution of the resulting forward Kolmogorov equation.

Following this theoretical approach, the infinitesimal transition rates and the resulting forward Kolmogorov equations are presented in Appendix B). Unfortunately, obtaining an explicit anylitical form of the CDFs or PDFs of the these variables using this approach turns out to be technically intractable.

Although the analysis of a finite-time queue is analytically intractable, it is convenient to simulate when the infinitesimal transition rates are given. Therefore, a numerical approach based on Monte Carlo simulation is applied for deriving RT predictions and for parameter estimation. Details of the simulation and parameter estimation method are presented in Chapter 5.1 and 6.

There is still a modeling decision that has an impact on the infinitesimal transition rates. By assuming an inhibition of return (Klein, 1988), the effective arrival rates differ from those presented in Appendix B.

3.4.2 Considering inhibition of return

A Markovian queueing model of the type $M/M/c/\infty/k/FCFS$ assumes that every customer outside of the system potentially comes in the system (again), regardless of whether the customer has been in the system once (see Section 3.2 and 3.3.2). Conceptually, this corresponds to the case where attention is allowed to move (back) to any object that it is currently not deployed to. That is, the observer does not differentiate which objects have been searched and which have not. However, this does not appear to be consistent with our subjective search experience, especially when the set size is small. There is also empirical evidence for the existence of an inhibition of return (Klein, 1988, for discussion see 4.2.1). This means that there is a mechanism that ensures objects that have

been searched once should not be examined again. On the other hand, there is evidence showing that the inhibition of return is not perfect (e.g., Horowitz & Wolfe, 1998). It is possible that attention will be directed to an object more than once during the search process. This is especially the case for large set size levels.

Specifying a model with *no* inhibition of return ($M/M/c/\infty/k/FCFS$) is of mathematical-theoretical advantage because there is a theoretical approach to deriving the CDFs of the variables of interest. However, it loses utility in application since an analytical closed form of the CDFs or PDFs of them is technically intractable (see Appendix B).

A perfectly functioning inhibition of return means that once a customer has left the system after receiving service, he or she is discarded from the system and will never return to the queue again (non-recurrent demand). A queueing model with finite customers and non-recurrent demand is strictly speaking no longer a Markov chain. The transition rates do not only depend on the number of customers in the system at the moment, but also on the cumulative number of arrivals (the history). On the other hand, a perfect inhibition of return is technically easier to implement using Monte Carlo simulation based approaches and the prediction is less complex so that less numerical difficulties are to expect (see Chapter 5.1 and Section 6.2.3).

For the sake of simplicity, I opt for the assumption of a perfect inhibition of return. In light of the empirical findings, neither the assumption of perfect inhibition of return nor the assumption of no inhibition of return describes visual search precisely. However, because previous studies mostly implemented small set sizes (less than 20), a perfect inhibition of return appears closer in describing the search behavior investigated in these studies. A psychologically meaningful modeling of inhibition of return can be a topic of future research (see discussion in Section 8.3.1).

The queueing model specified is thus revised to an $M/M/c/\infty/k/FCFS$ queue with *non-recurrent* demand. Accordingly, the infinitesimal transition rates are given as follows: For a specific time t, let n ($n = 0, 1, \ldots, k$) be the number of customers in the system, n_A ($n_A = 0, 1, \ldots, k$ and $n_A \geq n$) be the cumulative number of arrivals until t and λ_n the *effective arrival rate*, then

$$\lambda_n = (k - n_A)\lambda. \tag{3.1}$$

Let μ_n be the *effective departure rate*, then

$$\mu_n = \begin{cases} n\mu & \text{if } n = 0, 1, \ldots, c-1 \\ c\mu & \text{if } n = c, \ldots, k \end{cases} . \tag{3.2}$$

Now that a queuing model has been specified, it should be comprehensible to compare my model with the time-dependent limited-channel model and steady-state limited-channel model by Fisher (1982) mentioned in Section 3.1. The time-dependent limited-channel model is developed for standard visual search. It assumes a sequence of processing stages: encode, queue, scan and compare. The encoding is assumed as preattentive and massively parallel. A stimulus that has gone through the encoding stage has already a coherent representation but has not been mapped to a representation in memory yet. The mapping takes place in the comparison stage, which is assumed to be limited by four parallel channels. Due to the capacity limitation, encoded feature fragments have to wait in a queue if all channels are busy. Once a comparison at a channel finishes, the channel becomes free and the scanner assigns the next stimulus in the queue to the channel. Only the comparison stage is assumed to require attention. The steady-state limited-channel model is developed for visual search with sequentiall, brief presentation of multiple displays (see Section 2.5.3). It differs from the time-dependent limited-channel model in assuming no waiting room capacity (i.e., no waiting queue) and a constantly steady state of the system. If there is no comparison channel free upon the arrival of a stimulus, the encoded representation gets lost. In both models, interarrival times and service times are assumed to be exponentially distributed.

Apparently, the time-dependent limited-channel model by Fisher (1982) assumes the same structure of the queueing system as my model, namely a single-line, multiserver structure with limited numbers of channels/servers. There are three major differences between the time-dependent limited-channel model and my model. First, my model is based on different theoretical constructs of the underlying processing stages and maps them to stages of the queuing process differently. In my model, the arrival process (corresponding to the preattentive stage) results in feature fragments that need to be integrated, whereas it lead to

an integrated representation according to the time-dependent limited-channel model. My model assumes that integration of the feature fragments and identification are the same process that takes place in the service stage (corresponding to the attentive stage). For the time-dependent limited-channel model, identification is essentially mapping the stimuli representation to representations in memory. Second, my model assumes a finite customer source, whereas the time-dependent limited-channel model implicitly assumes an infinite customer population by specifying the arrival process as a Poisson process. Third, to characterize the preattentive stage as massively parallel, my model assumes an effective arrival rate that is proportional to the number of stimuli that has not been in the queue yet. The time-dependent limited-channel model assumes a stationary arrival process, which does not match the concept of parallel processing without capacity limitation.

So far, a queueing model for the RTs in visual search is developed. As mentioned in Section 3.3.1 and 3.3.3, it is restricted to accounting for RTs of correct responses. Although errors in a typical visual search experiment are rare, both error rates and error RTs contain important information about the search mechanism. More importantly, modeling errors in visual search has a relevance to the modeling of correct RTs. Recall that in Section 3.3.3, exhaustive search is assumed to derive the prediction on RTs on target-absent trials. As will be shown in the next chapter, the analysis of errors indicates that the assumption of exhaustive search needs to be revised, which changes the RT predictions.

Chapter 4

Modeling errors in visual search

In this chapter, a mechanism attributing errors in visual search to incomplete search and imperfect processing is proposed. Both false alarm probability and miss probability are derived from this mechanism. After being incorporated in the queueing model specified in the last chapter, it is also able to predict RTs of incorrect responses. In this way, the queueing model is extended to a model accounting for accuracy and RT data of visual search simultaneously.

4.1 Requirements on a error rate model and basic assumptions

Error rates observed in visual search exhibit some particular patterns consistently found in previous studies (Wolfe et al., 2010): 1) The miss rate increases as the set size gets larger; 2) The false alarm rate stays on roughly the same level regardless of the set size; 3) Holding the set size constant, the miss rate is usually much higher than the false alarm rate. These three patterns have been found for various task types and materials, for example, the conjunction search and spatial configuration task in the studies of Wolfe et al. (2010), as shown in Figure 4.1 (aggregated across all participants). The notable asymmetry in the behavior of the false alarm and the miss rate as a function of set size indicates that errors are probably not a result of pure guessing or motor errors. As discussed at the beginning of Section 2.3, while predicting RTs has always been strongly focused

on in models of visual search, the analysis of accuracy data has been rarely attracting attention of researchers. Although there are several attempts to model error rates in visual search (presented in Section 4.2), none of them appears to fully explain all three patterns characterized above in a systematic way.

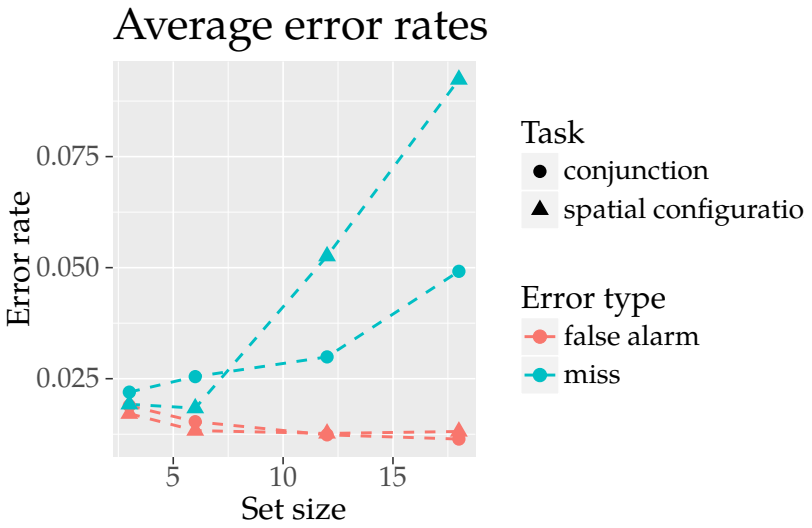

Figure 4.1: Average error rates as a function of set size, calculated from the data published by Wolfe et al. (2010).

I start with a naive conception of a mechanism that produces incorrect system responses and derive the false alarm and miss detection probability predicted by this mechanism using parsimonious assumptions. Then I show by monotonicity analysis that such a mechanism in its rudimentary form is inadequate to predict the patterns found in empirical data. After providing arguments for incomplete search, I present an approach predicting the search termination of human observers based on a quasi-optimization of the overall search efficiency under an accuracy constraint. Finally, showing the necessity of taking both genuine processing errors and premature termination into account, I propose a comprehensive error generating mechanism and derive error probabilities that such a mechanism predicts. After being integrated in the queueing model described in the last chapter, this accuracy model is also able to predict RTs of error trials.

Since the three patterns mentioned above have been reliably observed in accuracy data of previous visual search studies, it is reasonable to formalize

them as requirements on a model accounting for error rates in visual search. The proposed error generating mechanism should be able to reproduce three patterns:

Requirement a) The miss rate increases with increasing set size;

Requirement b) The false alarm rate remains roughly constant (or slightly decreasing) with increasing set size;

Requirement c) For the same set size, the miss rate is higher than the false alarm rate.

The servers in the queueing model specified in Chapter 3 are assumed to be independent of each other. In queueing theory, this refers to the stochastic independence of the service times at each server. Although queueing theory does not deal with service outcomes, we can assume a stochastic independence of the processing outcomes (i.e., the identification of a stimulus) at each server. This implies that identification errors also occur independently at each server. When the final system response is determined by a series of service outcomes at different servers, the probability of each response option should reflect the overall service quality of the system. Using the terms of queueing theory, three basic assumptions that characterize an error producing mechanism can be formulated. Among these three assumptions, assumption 2 and 3 are in a rudimentary form. I first explain why Assumption 3 needs revision in Section 4.1.2 and how it is revised to meet the requirements in Section 4.2 and 4.3. The reason why Assumption 2 needs alteration and how it is altered is presented in Section 4.4.

Assumption 1. Servers provide imperfect service. They make two types of mistakes:

- **1.1** Mistaking an ordinary customer for the target (system reports "target" given a distractor, denoted by "T" | D);
- **1.2** Mistaking the target for an ordinary customer (system reports "distractor" given the target, denoted by "D" | T).

4.1.1 A rudimentary model and its predictions on error probabilities

Assumption 2 (rudimentary). These two types of mistakes occur at each server independently with the same (conditional) probabilities p_1 and p_2 respectively, i.e., $\mathbb{P}(\text{“}T\text{”} \mid D) = p_1$ and $\mathbb{P}(\text{“}D\text{”} \mid T) = p_2$. This means that the categorization of an item is assumed to depend only on the processing outcome of this item, and to be independent of the processing outcomes of all other items in the display.

Assumption 3 (rudimentary). The service process terminates as soon as either of the following two events occurs:

3.1 A customer is judged as “target” (self-terminating), which initiates a positive response (pressing “yes”);

3.2 All customers have been served and judged as “distractor” (exhaustive search), which initiates a negative response (pressing “no”).

Based on these three assumptions, the predicted probability of a “yes” response on a target-absent trial and the predicted probability of a “no” response on a target-present trial can be calculated by:

$$\begin{aligned}\mathbb{P}(\text{false alarm}) &= \mathbb{P}(\text{“yes”} \mid \text{target-absent})\\ &= 1 - \mathbb{P}(\text{examining and rejecting all } k \text{ distractors})\\ &= 1 - (\mathbb{P}(\text{“}D\text{”} \mid D))^k\\ &= 1 - (1 - \mathbb{P}(\text{“}T\text{”} \mid D))^k\\ &= 1 - (1 - p_1)^k\end{aligned} \tag{4.1}$$

and

$$\begin{aligned}\mathbb{P}(\text{miss}) &= \mathbb{P}(\text{“no”} \mid \text{target-present})\\ &= \mathbb{P}(\text{examining and rejecting all } k-1 \text{ distractors and the target})\\ &= \mathbb{P}(\text{“}D\text{”} \mid D)^{k-1} \cdot \mathbb{P}(\text{“}D\text{”} \mid T)\\ &= (1 - p_1)^{k-1} p_2.\end{aligned} \tag{4.2}$$

Note that Assumption 3 plays the role of connecting Assumption 1 and 2, which describe an error invoking mechanism on the level of servers, to the probabilities of observing false alarms and misses on an experimental trial.

The model characterized by Assumption 1 to 3 (rudimentary) is essentially equivalent to a model postulated by Zenger and Fahle (1997), assuming that their "Strategy 1" is applied in the visual search. The largest difference is that Zenger and Fahle (1997) adopted Signal Detection Theory (Green & Swets, 1966) to account for the categorization of every single stimulus as a target or a distractor. Applying "Strategy 1" means setting a constant discrimination threshold across different set sizes, which necessarily leads to constant p_1 and p_2 *regardless of set size*. The model I propose is embedded in the framework of a queueing model with systematized model assumptions and elaborate derivation.

4.1.2 Plausibility check

Analyzing the monotonicity of the predicted $\mathbb{P}$(false alarm) and $\mathbb{P}$(miss) as functions of set size k enables a quick evaluation of the applicability of the assumptions from a mathematical point of view. Despite its apparent simplicity, this rudimentary model consisting of Assumptions 1 to 3 is not able to predict the three patterns observed in empirical data.

- Requirement a) cannot be met because it follows from Equation (4.2) that $\mathbb{P}$(miss) is a decreasing function in k.
- Requirement b) cannot be met because it follows from Equation (4.1) that $\mathbb{P}$(false alarm) is an increasing function in k.
- Since

$$\begin{aligned}\mathbb{P}(\text{miss}) - \mathbb{P}(\text{false alarm}) &= (1-p_1)^{k-1}p_2 - (1-(1-p_1)^k) \\ &= (1-p_1)^{k-1}(1+p_2-p_1) - 1,\end{aligned}$$

  Requirement c) can be met if and only if $p_2 - p_1 > \frac{1}{(1-p_1)^{k-1}} - 1$. As the empirical error rates are usually very low (less than 0.1), p_1 and p_2 would be estimated at very small values according to the model. Consequently, for every fixed k, the set of p_1 and p_2 fulfilling this inequality is sufficiently

large. However, for fixed p_1 and p_2 which fulfill the inequality, if we let k increase, the inequality will not hold once k exceeds a certain value, because $\frac{1}{(1-p_1)^{k-1}}$ gets larger as k increases. This means that for constant p_1 and p_2, the model predicts that the false alarm probability will always outpace the miss probability as k increases and thus exceed it when k gets large enough. This is associated with the behavior that as k increases, the predicted discrepancy between miss probability and false alarm probability, i.e., $(1-p_1)^{k-1}(1+p_2-p_1)-1$, declines because $(1-p_1)^{k-1}$ gets smaller. Both are inconsistent with the empirical observation that miss rate dominates false alarm rate with increasing ascendancy as k increases.

The inconsistency between the implications of Equations (4.1) and (4.2) and Requirement a) and b) motivated Zenger and Fahle (1997) to introduce further assumptions of strategy use by observers. Their approach is to relax the constraint of constant p_1 and p_2, assuming that observers shift their discrimination thresholds so that a certain quantity (can be understood as "risk") is minimized. They proposed that this quantity can be the probability of misclassification (the sum of $\mathbb{P}(\text{“}T\text{”} \mid D)$ and $\mathbb{P}(\text{“}D\text{”} \mid T)$)) weighted by the incidence probability of distractors and the target $\frac{2k-1}{2k}\mathbb{P}(\text{“}T\text{”} \mid D) + \frac{1}{2k}\mathbb{P}(\text{“}D\text{”} \mid T)$, referred to as "Strategy 2", or the expected overall error rate ($\mathbb{P}$(miss) + $\mathbb{P}$(false alarm)), referred to as "Strategy 3". Since p_1 exerts a larger influence on both quantities than p_2 does as k increases (reflected in the factor $\frac{2k-1}{2k}$ and the power k), both strategies require p_1 to become smaller. Consequently, both strategies predict a threshold shift in the more conservative direction and p_2 has to become larger according to SDT. In this way, the assumptions of strategy use in Zenger and Fahle (1997) are essentially equivalent to a relaxation of Assumption 2 in my rudimentary model, such that p_1 decreases whereas p_2 increases with increasing k. The authors claimed that the modified model with this sort of threshold adaptation ("Strategy 2" and "Strategy 3") alone is sufficient to reproduce the patterns of increasing miss rates and relatively flat false alarm rates. They further conclude that we should refrain from attributing errors to multiple sources such as premature termination, as per Occam's razor.

Zenger and Fahle (1997) doubtlessly provided an original approach to modeling error rates in visual search. They showed via simulation that their model

with threshold adaptation could predict increasing miss rates and relatively flat false alarm rates. However, their results appear insufficient to forgo taking other sources of errors into account when modeling. First of all, there are two systematic differences between the empirical data and model predictions. For the purpose of illustration, Figure 3 in Zenger and Fahle (1997) is reprinted here as Figure 4.2. It plots the miss and false alarm rated predicted by their model for set sizes between 1 and 12, assuming the application of "Strategy 1", "Strategy 2" and "Strategy 3."

The first systematic difference is that the growth of the empirical miss rate speeds up with increasing set size and does not appear to be limited in the data, whereas it slows down and is limited in their model. In other words, the data show miss rates as a convex, increasing function of set size, whereas their model predicts this relation as a concave, increasing function. Although the authors considered this as a consequence of second-order effects, neglecting it may conceal important information carried by the data because the convex pattern appears consistently in visual search data whereas the concavity seems to be an intrinsic property of their model.

The second systematic difference is that the empirical false alarm rates do not show any upward trend but rather occasionally a slight downward trend with increasing k, whereas the false alarm rates predicted by their model always ascend slightly despite of a generally flat pattern. These differences become more profound when the value of the error rate becomes larger (reflecting more difficult search tasks). Another notable point is that the set size in the empirical data used in Zenger and Fahle (1997) consists of merely three levels with relatively small distance (2, 8, 14). This limits the inspection of patterns in the data and may undermine or even conceal critical discrepancies between the data and model predictions.

The analysis above demonstrates the limitations of the sort of models that assume classification errors (or strictly speaking, genuine processing errors) to be the only source of errors in visual search. Now one could argue that these limitations might result from other assumptions of the models. Nevertheless, in addition to the mathematical arguments discussed above, there are rationality arguments as well as empirical evidence for a substantial role of incomplete

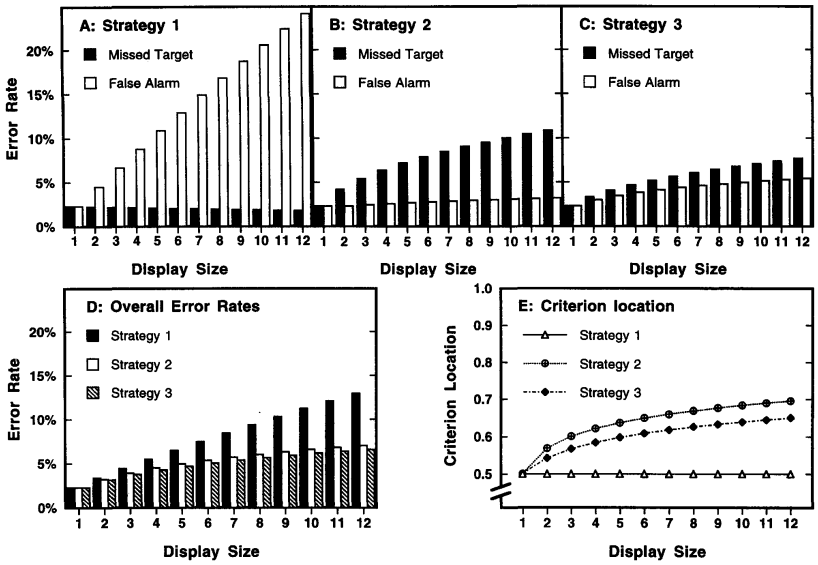

Figure 4.2: Error rates predicted by the model of Zenger and Fahle (1997) for set sizes between 1 and 12, assuming the application of "Strategy 1", "Strategy 2" and "Strategy 3." Reprinted from "Missed targets are more frequent than false alarms: A model for error rates in visual search," by B. Zenger and M. Fahle, 1997, *Journal of Experimental Psychology: Human Perception and Performance*, *23*(6), p. 1787. Copyright 1997 by the American Psychological Association.

search. In the next section, I will elaborate the issue of how observers decide when to terminate a search.

4.2 Incomplete search

4.2.1 Theoretical and empirical arguments for incomplete search

As mentioned in the last section, Assumption 3 specifies how a search is terminated. The self-terminating nature of visual search (Assumption 3.1) has usually been taken for granted. Since the instruction of a standard visual search experiment explicitly provides the valid information that each trial contains either no or exactly one target, there is no reason for observers to continue searching once they have classified an item as the target. So far, there is also hardly any empirical finding that questions this assumption.

In contrast, exhaustive search (Assumption 3.2) appears to be a very strict assumption. Do people search through all items in the display if they fail to classify any item as the target? They should if they do not want to falsely make a "no" response by omitting any item in the display, which could have been the target. However, an exhaustive search on every target-absent trial requires both the willingness and ability to execute. The data of standard visual search experiments mostly reflect how untrained observers perform tasks without serious, real consequences — as distinct from the settings in which airport security looks for threats or radiologists look for signs of cancer. An ordinary participant may lose patience after completing several hundreds of monotonous visual search tasks and would rather sacrifice some accuracy in order to finish the remaining trials sooner. Moreover, exhaustive search relies on perfect memory for rejected items. Many studies have been conducted to investigate the question of whether visual search has a memory (e.g., Dodd et al., 2003; Gilchrist & Harvey, 2006; Horowitz & Wolfe, 1998, 2003; Klein, 1988; Klein & MacInnes, 1999). Wolfe (2012) summarized these findings and concluded that there is no perfect memory for rejected items but enough inhibition of return to the searched locations directing attention to new items. In other words, even if observers are willing to perform an exhaustive search, they may not necessarily succeed in executing it, especially when the set size is large or the display is cluttered.

Let us leave aside individual variations in persistence and ability to memorize searched locations from the discussion for a moment and consider how a prototypical (or average) observer would perform standard visual search tasks. In every standard visual search experiment, participants are instructed to "respond as fast and accurately as possible." These two demands unavoidably result in a speed-accuracy trade-off because achieving the maximum of one necessarily leads to a deviation from the maximum of the other. Ideally, to adjust his or her behavior, an observer should follow one single measure which quantifies the extend to which both conflicting demands are fulfilled simultaneously, that is, the search efficiency (the measure of search efficiency will be discussed in Section 4.3.4). Hence, from an economic rationality perspective, efficiency optimization underlies incomplete search. This also makes up the theoretical basis for my approach towards explaining error rates in visual search.

From the empirical perspective, there are several lines of findings considered as evidence for incomplete search. One finding considered as indirect evidence is that in most visual search studies, the variance of RTs on target-absent trials is found to be larger than that of target-present trials of the same set size (Chun & Wolfe, 1996; Ward & McClelland, 1989). This lets some researchers cast doubt on an exhaustive search. They argue that target-present trials should produce larger RT variance than target-absent trials under the assumption of exhaustive search because on target-present trials, the number of items examined when the search stops can vary from 1 to the set size k, whereas it should always be k on target-absent trials. However, this reasoning does not take the stochastic properties of processing times for a fixed number of items into consideration. As Townsend and Ashby (1983, Propositions 7.7 and 7.8, pp. 194-196) show, if the variance of the time required to process a single item is large enough, RT variance on target-absent trials can exceed target-present trials even under an exhaustive search. Since the proposed queueing models take account of these stochastic aspects, they do not imply the superiority of RT variance of either trial type a priori. The prediction depends on the concrete values of parameters. Thus, findings regarding RT variance cannot provide clear evidence for the issue.

Another line of argument is also based on an implication of RT data analysis, more precisely, on results from model comparisons. Models that assume an incomplete search often turn out to perform better in explaining empirical RT data than models assuming exhaustive search (e.g., Chun & Wolfe, 1996; Cousineau & Shiffrin, 2004; Wolfe, 2012).

Studies using an eye-tracking method provide perhaps the most convincing evidence. Rich et al. (2008) recorded eye movements of participants during standard visual search tasks. By analyzing fixations on miss trials, they found out that failing to gaze on the target made up the prime source of miss errors. They report that on up to a half (if target prevalence is high) to two third (if target prevalence is low) of the miss trials, the trajectory of eye movement did not include the target or even its near areas, indicating that search had been probably abandoned before the target could be examined at all. Subsequent studies (e.g., Godwin et al., 2015; Hout et al., 2015; Peltier & Becker, 2016) replicated this finding and confirmed that omissions caused the majority of miss errors in

standard abstract visual search, regardless of target prevalence.

In summary, exhaustive search (Assumption 3.2) does not describe the termination of search correctly and needs modification. Having seen the necessity of taking omissions into account, we now come to the discussion on how to model an incomplete search. Several approaches are proposed in the literature (e.g., Moran et al., 2013; Wolfe, 2012).

4.2.2 Approaches to modeling incomplete search

4.2.2.1 Preselecting by feature

This kind of models modifies the assumption of exhaustive search such that the search is terminated after an exhaustive search through a preselected subset of stimuli. It is assumed that which elements are contained in the subset has been determined before attention is directed to any item, i.e., the selection of those items that will be inspected occurs before any item is examined and identified. For example, Egeth et al. (1984) showed that searchers were able to search through only a subset of stimuli defined by a shared feature of the target (e.g., searching only among red objects when the target is a red "O"). However, such models appear not to be able to describe the behavior in conjunction search generally. As Friedman-Hill and Wolfe (1995) showed, the patterns of both RT and accuracy data in standard conjunction search differ clearly from those for the same material when subjects were explicitly instructed to search in relevant subsets.

4.2.2.2 Preselecting by activation threshold

As a variant of restricting subsets to be searched, Wolfe (1994a) proposed in Guided Search 2.0 that only those items whose activation is above a threshold will be examined. An activation map is formed upon the stimulus onset. The activation arises from both bottom-up activity and top-down guidance and determines also the order in which items will be examined. When all items above the activation threshold are examined and classified as distractors, the search will be terminated. The threshold is assumed to be set adaptively and automatically. As Wolfe (2012) pointed out, because the search within the subset

is assumed to be exhaustive, implementing this kind of mechanisms requires a (nearly) perfect memory of which items have been rejected.

4.2.2.3 Random quitting

Cousineau and Shiffrin (2004) proposed a simple approach according to which the searcher quits with a certain probability after each non-target judgment (see Section 2.4.2 for a description of the RT modeling approach of their model). This (conditional) probability can vary at each time when a "quit or stay" decision has to be made (i.e., when an item is examined and rejected), so that it is a free parameter to be estimated. This means that the number of free parameters needed to characterize the quitting behavior is $k - 1$ on target-absent trials, if there are k items in the display. This property makes this approach difficult to apply to data with large set size.

An advanced version of this approach is proposed by Donkin and Shiffrin (2011), which relates the (conditional) probability of quitting positively to the proportion of rejected items so far (see Section 2.4.2). More precisely, the (conditional) quitting probability after each non-target judgment is assumed to be a logistic function of the proportion of the cumulative non-target judgments to the set size. Since a logistic function is characterized by its location and scale, all quitting probabilities can be determined once these two parameters are fixed. In this way, only two free parameters are necessary regardless of the set size k. This kind of approach can be viewed essentially as a survival analysis of the search. Note that although this approach assumes a relation between the proportion of rejected items (i.e., the search experience) and the probability of quitting, it does not assume any threshold of the number of items searched. The search can be terminated after the identification of any item.

4.2.2.4 Quitting by selecting a "quit unit"

The concept of quitting randomly is comprehensible and appealing as it can be interpreted as the dynamic tendency of terminating a search. However, the implementations by Cousineau and Shiffrin (2004) and Donkin and Shiffrin (2011) do not reveal much about the mechanism that governs the quitting tendency. The

CGS (Moran et al., 2013, see Section 2.4.2) deals with the problem by introducing the notion of a "quit unit". The quit unit is a hypothetical entity that carries activation weight (in the same sense as in the preselecting by activation threshold approach) and competes with all display items for attention. Once it is selected by attention, the search is terminated. The proportion of the activation weight of the quit unit to the sum of all activations determines the probability of selecting the quit unit. Moreover, it is assumed that the activation of an item is inhibited after it has been rejected and that rejecting an additional item adds an increment to the activation weight of the quit unit. Both lead to a dynamic change in the activation map. The probability of selecting the quit unit thus increases as the number of rejected items increases.

4.2.2.5 Quitting by timing or counting

Unlike the preselecting mechanisms described above, quitting by timing or counting does not specify which items will be examined ahead of the search. According to this approach, the search is terminated once an internal time limit expires or a certain quantity that changes as the search progresses reaches a certain threshold. Timing or counting mechanisms reflect the idea that observers accumulate information about the current trial continuously and use it to determine the time to quit when the information is judged as enough. For instance, when the searcher expects that on most target-present trials (say, 90%), the target is found within time t, then he or she may think that a trial is unlikely to contain the target if he or she has searched until t and found no target. By the same token, the termination criterion could be a counting threshold of the number of items inspected until the target is found.

4.2.2.6 Why quitting by counting is appropriate

Because the queueing model does not take an activation into account as postulated in Wolfe (1994a) and Moran et al. (2013), preselecting by activation threshold or quitting by selecting a quit unit are not applicable without further, specific assumptions. For the random quitting approach (the advanced version), not the set size, but the proportion of the cumulative non-target judgments to the

set size is critical for the termination. Thus, it predicts the same pattern of quitting behavior for different set sizes. It is unclear how such a mechanism is compatible with the changes observed in empirical error rates (miss rates above all) depending on the set size. Thus, only the approach of quitting by timing or counting is elaborated here.

As Wolfe et al. (2010) pointed out, a timing account makes predictions that are inconsistent with the empirical RT data. They reason that when an observer sets a time threshold such that only a tolerable percentage of targets will be missed if he or she quits at this time point, the median target-absent RT should be much longer than most target-present RTs. For example, if the observer accepts a 5% miss rate, the median target-absent RT should lie roughly on the 95 percentile of the target-present RT. However, the empirical target-absent RT overlaps largely the target-present RT, so that its median lies much closer to the median of target-present RT than such a timing mechanism would predict.

Besides this argument, I shall point out another observation that contradicts the notion of terminating once an internal time limit expires. If the search is always aborted when no target has been found by an internal time limit $t_{threshold}$, the target-absent RT must be restricted so that it will not exceed a certain time $t_{threshold} + t_{residual}$[1]. This means that the distribution of target-absent RT will be truncated for a given set size k, or at least the target-absent RT will have a small variance, when considering the variability of the residual time. But the empirical target-absent RT distribution looks nothing like truncated; it is right-skewed with a long, fat tail (see Section 7.2), and the RT has a large variance. Hence, quitting based on elapsed time is not plausible in light of the empirical data[2].

What about a counting mechanism? The prediction of a counting mechanism on the RTs depends on the progress of the processing. Since the time a queueing system needs to process a fixed number of stimuli is variable and its mean and variance increase as the number of stimuli to process increases, it is possible to predict a large overlap of the target-absent and target-present RTs and a positively

[1] According to the authors, the residual time includes decision time and motor execution time, which may be variable.

[2] These arguments are based on the assumption that quitting upon the time limit always initiates a negative response. The observer is of course free to response "yes" after the time limit expires. However, a positive response in this case, even if it is a guess, is irrational and counterintuitive, see Section 4.3.2 for details.

skewed distribution with a long, fat tail. Therefore, a counting mechanism appears to be the most appropriate choice for an error model that can be integrated into the Markovian queueing model of visual search.

Based on the considerations above, I adopted a counting approach to revise Assumption 3. For the sake of simplicity, I retain Assumption 2 in the discussion of revising Assumption 3 to model incomplete search. In Section 4.4, I will come back to the issue of altering Assumption 2 and discuss the adjustment of p_1 and p_2 for the set size k.

4.3 Revising Assumption 3: Quitting by counting

Assumption 3 is revised as follows:

Assumption 3 (revised). The service process terminates as soon as either of the events occurs:

3.1. A customer is judged as "target" (self-terminating), which initiates a positive response (pressing "yes");

3.2. l customers have been served and judged as "distractor", which initiates a negative response (pressing "no"), whereby $1 \leq l \leq k$

The revised Assumption 3 assigns the response options ("yes"/"no") to the search outcomes (finding the target/no target has been found) in this way because for any given l, such an assignment maximizes the conditional probability of correct response to the current trial given the information collected by search. In other words, responding "yes" only when the target is found regardless of how large l is chosen (as long as $0 \leq l \leq k$) yields a better mean accuracy than any other combination of response option and search result (I will give a detailed reasoning for this statement in Section 4.3.2). Denote the event "target-absent" as TA and "target-present" as TP. Under the revised Assumptions 1 to 3, the probability of a "yes" response on a target-absent trial and the probability of a

"no" response on a target-present trial are given by

$$\begin{aligned}\mathbb{P}(\text{false alarm}) &= \mathbb{P}(\text{“yes”} \mid \text{TA}) \\ &= 1 - \mathbb{P}(\text{examining and rejecting } l \text{ distractors} \mid \text{TA}) \\ &= 1 - (\mathbb{P}(\text{“}D\text{”} \mid D))^l \\ &= 1 - (1 - p_1)^l\end{aligned} \tag{4.3}$$

and

$$\begin{aligned}\mathbb{P}(\text{miss}) &= \mathbb{P}(\text{“no”} \mid \text{TP}) \\ &= \mathbb{P}(\text{examining and rejecting } l \text{ distractors} \mid \text{TP}) \\ &\quad + \mathbb{P}(\text{examining and rejecting } l-1 \text{ distractors and the target} \mid \text{TP}) \\ &= \frac{\binom{k-1}{l}}{\binom{k}{l}} \cdot (1-p_1)^l + \frac{\binom{k-1}{l-1}}{\binom{k}{l}} \cdot (1-p_1)^{l-1} \cdot p_2 \\ &= \frac{(k-1)!}{(k-1-l)!l!} \cdot \frac{(k-l)!l!}{k!} \cdot (1-p_1)^l \\ &\quad + \frac{(k-1)!}{((k-1)-(l-1))!(l-1)!} \cdot \frac{(k-l)!l!}{k!} \cdot (1-p_1)^{l-1} \cdot p_2 \\ &= \frac{k-l}{k} \cdot (1-p_1)^l + \frac{l}{k} \cdot (1-p_1)^{l-1} \cdot p_2,\end{aligned} \tag{4.4}$$

respectively.

4.3.1 Necessary condition under constant p_1 and p_2

Assume that p_1 and p_2 are invariant of k. The question of how the counting threshold l is determined arises. One might easily think of several simple options, for example, a constant threshold or a fixed ratio of l to the set size k etc. Nevertheless, a second thought disproves their appropriateness of being the counting threshold. Consider first the case in which the search is terminated after examining a fixed proportion of display items. Let $r = \frac{l}{k} = \text{const.}$, then

$$\begin{aligned}\mathbb{P}(\text{miss}) &= \frac{k-rk}{k} \cdot (1-p_1)^{rk} + \frac{rk}{k} \cdot (1-p_1)^{rk-1} \cdot p_2 \\ &= (1-r)(1-p_1)^{rk} + r(1-p_1)^{rk-1} \cdot p_2\end{aligned}$$

It implies that $\mathbb{P}(\text{miss})$ should be decreasing for increasing k if $p_1 > 0$, and constant if $p_1 = 0$, because r, p_1 and p_2 in the equation are assumed to be constant. This prediction is inconsistent with Requirement a). Now consider the case in which l is constant, i.e., remains the same for all k. We can assess the monotonicity of $\mathbb{P}(\text{miss})$ as a function of k by calculating its derivative:

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}k}\mathbb{P}(\text{miss}) &= \frac{k-(k-l)}{k^2}(1-p_1)^l - \frac{l}{k^2}(1-p_1)^{l-1}\cdot p_2 \\ &= \frac{l}{k^2}(1-p_1)^{l-1}(1-p_1-p_2) > 0, \text{ if } p_1+p_2 < 1.\end{aligned}$$

Thus, Requirement a) can be met with the constraint $p_1 + p_2 < 1$, which is not quite restrictive. However, this inference is valid only under the condition $l \leq k$. For the case $k < l$, we should assume that simply all items will be examined, i.e, an exhaustive search. Considering both cases together, the mere idea of aborting the search after examining a fixed number of items is inconsistent with the empirical data: If the fixed threshold l is larger than all set size levels realized in the experiment, decreasing miss rates should be observed; if some of the realized set size levels exceed l, i.e., l lies between two set size levels, the miss rates should first decrease then increase. This means that l must be smaller than all set sizes to be able to reproduce a pattern of constantly increasing miss rates. Since many studies include small set sizes (less than 10), the case that l is smaller than all set size levels realized in the studies is very implausible. Besides, the concept of a counting threshold that is invariant of the set size itself does not seem convincing.

Although these two options have been disproved, the discussion is not in vain. It provides a hint for finding out conditions that l needs to satisfy so that the miss probability is increasing. An observation is that assuming l is constant, the ratio $\frac{l}{k}$ decreases as k increases. One might speculate that decreasing $\frac{l}{k}$ could be a necessary condition. A closer look at Equation (4.4) for the miss probability provides a rational argument for this speculation. The term $\frac{k-l}{k}\cdot(1-p_1)^l$ is the probability of the case where the searcher misses the target because he or she did not look at it at all; the term $\frac{l}{k}\cdot(1-p_1)^{l-1}\cdot p_2$ is the probability of the case where although the target is looked at, the searcher failed to identify it as such. The ratio of the former to the latter is thus $\frac{k-l}{l}\cdot\frac{1-p_1}{p_2}$. Because the empirical false

alarm rate and miss rate are usually very low (under 10%), p_1 and p_2 must be very small (≤ 0.05). This means that the fraction $\frac{1-p_1}{p_2}$ should be a large value (≥ 19). Consequently, the ratio $\frac{k-l}{l} \cdot \frac{1-p_1}{p_2}$ can only fall below 1 if $\frac{k-l}{l}$ is smaller than about 0.05. In other words, the case where the target is *not* included in the searched set makes up the dominating source of miss errors, unless the search is almost exhaustive (i.e., more than 95% of the items must be examined).

From this observation, we can infer that the option of the increasing ratio $\frac{l}{k}$ with k should be excluded. If $\frac{l}{k}$ was increasing with k, then it would be the only increasing factor in Equation (4.4). If $\frac{l}{k}$ is below 0.95, $\mathbb{P}$(miss) would be decreasing with increasing k because the dominating part $\frac{k-l}{k} \cdot (1-p_1)^l$ is decreasing. If $\frac{l}{k}$ is above 0.95, $\mathbb{P}$(miss) would hardly be increasing with increasing k because the monotonic behavior of the factor $(1-p_1)^l$ (and $(1-p_1)^{l-1}$) is faster than a linear increase[3] and the influence of the only increasing factor $\frac{l}{k}$ is very limited (more slowly than linear, from 0.95 to 1 at the most, and then shrunk by the small p_2). Even if this influence was strong enough (for specific value configuration of the variables) to compensate all other decreasing factors, the increase in miss probability would be minuscule. In other words, under the assumption of increasing $\frac{l}{k}$, we would not be able to observe such monotonic and substantial increases in empirical miss rates. Any model that predicts an increasing proportion of the searched subset with increasing set size would fail to predict the empirical pattern in Requirement a).

On the other hand, if we assume $\frac{l}{k}$ to be decreasing with increasing k, as one might speculate from the previous discussion, increasing miss probability can be a natural consequence. If $\frac{l}{k}$ decreases with increasing k so fast that even l decreases, then itself would be the only decreasing factor in Equation (4.4). Even if it starts from 1, it would fall below 0.95 very soon. Due to the dominance of $\frac{k-l}{k} \cdot (1-p_1)^l$, its influence can be easily compensated and overtaken by the exponential increase of the factor $(1-p_1)^{l-1}$. Nevertheless, searching even less items when there are more appears less plausible than searching a fixed number of items regardless of set size. If l remains increasing as k increases, then all

[3]Note that for very small p_1, the increase of $(1-p_1)^l$ is approximately linear, if $\frac{l}{k}$ is constant and l is relatively small. Under the condition of increasing $\frac{l}{k}$, l would increase even faster as k increases.

factors in Equation (4.4) would be decreasing except $\frac{k-l}{k}$. Because $\frac{l}{k}$ is decreasing, the factor $(1 - p_1)^{l-1}$ decreases more slowly than it does if $\frac{l}{k}$ is constant (as discussed above), i.e., more slowly than linear. This means that the increasing behavior of $\frac{k-l}{k}$ can be of similar magnitude as the decreasing behavior of all other factors so that it determines the monotonic behavior of $\mathbb{P}$(miss) eventually. In other words, depending on the properties of $\frac{l}{k}$, the miss probability can be increasing. This is the only relation that can fulfill Requirement a).

Explained in plain language without reference to formulas, the critical part of the reasoning above is to consider the relative impact of the two situations where a miss error occurs: The target was not included in the searched subset in the first place, although all examined items are rejected correctly; the target was included in the searched subset but rejected, like the other examined items. If the occurrence of a miss error requires an erroneous processing, i.e., a misidentification of the target, the miss rate cannot rise when there are more items in the display. Saying "no" implies that all distractors, once attended to, must be rejected correctly. Rejecting all of them correctly becomes harder when more distractors are attended to. Because there is only one target on each target-present trial of a standard visual search task, including it in a random sample — and subsequently misidentifying it — also becomes harder when there are more items in the display. Therefore, the extra occurrences of miss errors for larger set sizes must result largely from the omission of the target rather than its misidentification. To ensure a sufficiently large relative contribution of the omission, the proportion of the items selected to examine must become smaller as the set size gets larger. Only this condition can predict increasing miss rates as the set size increases.

If this is the case, what is the mechanism behind it? Can we describe the relation between the proportion of items to search and the set size more precisely than "it seems to be negative?" It may come to one's mind that the purpose of looking at stimuli is to collect information to make a decision. What about viewing the problem from the perspective of "how the observer decides which response option to choose based on her search experience?"

4.3.2 Posterior probability of the target absence

Let us focus on the search process on one single trial. As the search of the currently displayed items progresses, the observer collects information about the current trial. Once the target is found, the search can be terminated immediately, resulting in a "yes" response. If this does not happen after a certain number of items has been examined, search experience suggests that the posterior probability of the current trial being a target-present trial decreases (and the posterior probability of the target absence increases accordingly). At some point of the search, the probability that continuing searching provides new information that will improve the accuracy (i.e., finding the target) drops below a threshold. Now the observer may decide the timing of termination according to such a rule: "If the posterior probability of finding the target drops below a certain value, I quit and say 'no'." By setting such a criterion, the observer can restrict the probability of missing the target. In the following, for the sake of convenience, I focus on the posterior probability of target absence and discuss how to determine a confidence criterion above which the observer would feel safe to quit and respond "no".

Obviously, the posterior probability of target absence is connected to l, the number of items searched. To calculate the posterior probability of target absence, the key event to consider is "the l searched items are all distractors" (denoted as SAD(l)). At the very beginning of the search, no posterior information is available because no item has been searched yet. The prior probability that the current trial is a target-absent trial corresponds to the prior probability, i.e., $\frac{1}{2}$. If no target has been found after finishing examining l-th item, then by Bayes' theorem, the posterior probability of the current trial being target-absent can be calculated as follows:

$$\begin{aligned}
\mathbb{P}(\text{SAD}(l)) &= \mathbb{P}(\text{SAD}(l) \text{ and TA}) + \mathbb{P}(\text{SAD}(l) \text{ and TP}) \\
&= \mathbb{P}(\text{SAD}(l) \mid \text{TA})\mathbb{P}(\text{TA}) + \mathbb{P}(\text{SAD}(l) \mid \text{TP})\mathbb{P}(\text{TP}) \\
&= 1 \cdot \frac{1}{2} + \frac{\binom{k-1}{l}}{\binom{k}{l}} \cdot \frac{1}{2} \\
&= \frac{1}{2} + \frac{k-l}{k} \cdot \frac{1}{2} = \frac{2k-l}{2k}
\end{aligned}$$

$$\mathbb{P}(\text{TA} \mid \text{SAD}(l)) = \frac{\mathbb{P}(\text{TA and SAD}(l))}{\mathbb{P}(\text{SAD}(l))} = \frac{k}{2k-l} \tag{4.5}$$

$$\mathbb{P}(\text{TP} \mid \text{SAD}(l)) = \frac{\mathbb{P}(\text{TP and SAD}(l))}{\mathbb{P}(\text{SAD}(l))} = \frac{k-l}{2k-l} \tag{4.6}$$

Equation (4.5) shows that the posterior probability of target absence given SAD(l) increases with increasing l and becomes 1 when l reaches k. It is complementary to the posterior probability of target presence given SAD(l), which decreases with increasing l. It is easy to see that once the search has begun, the conditional probability of the current trial being target-present given SAD(l) is always lower than the corresponding conditional probability of it being target-absent, regardless of how large l is. This is the reason searchers should always response "no" when they quit the search without finding the target. By always responding "no", they will achieve a higher accuracy on average than responding in any other way.

Then, to ensure the decision ("no" response) made upon leaving the current trial with a posterior probability of at least s being correct ($\frac{1}{2} \leq s \leq 1$ represents thus the confidence criterion that provides a sense of safety for the searcher), it follows from $\mathbb{P}(\text{TA} \mid \text{SAD}(l)) = \frac{k}{2k-l} \geq s$ that the number of items that need to be examined must satisfy

$$l \geq 2k - \frac{k}{s}.$$

Furthermore, we can infer that the proportion of items to search must satisfy

$$r = \frac{l}{k} \geq 2 - \frac{1}{s}.$$

If r decreases with increasing k, then increasing k should also lower the confidence criterion s. This means that people may relax their confidence criterion when facing more stimuli in the display. In other words, the pattern of increasing miss rates with increasing set sizes is consistent with the account that people search according to a stricter confidence criterion when the set size is small while they use a lax confidence criterion when it is large.

If this is the case, why do people relax their confidence criterion when there are more items in the display? Recall the discussion at the beginning of Section

4.2 on behavioral reasons for an incomplete search. If we accept the notion that efficiency optimization underlies incomplete search, then this principle should also guide the behavior of search termination. Intuitively, searching a smaller proportion for a larger set size is in accordance with efficiency optimization. The more items searched, the more information collected, yet the more time and effort spent. Small or large, each trial requires only one response, but for larger trials, collecting enough information to make a correct response costs more time than for smaller trials. Then the more efficient way to search would be to ensure more correct responses on small trials and to spend less time on large ones. To derive the optimal size of the subset to search, a single measure of the search efficiency is necessary, as discussed above. In the following sections, this problem is solved by using a similar approach as in optimal foraging theory. I begin with outlining the concept of optimal foraging theory and its application in studying search termination of human subjects.

4.3.3 Optimal foraging

Let us rethink the problem an observer faces. A human observer sees trials containing varying numbers of items and must decide whether a predefined target is present or not. During the search, he or she tries to answer as many trials as possible correctly within as short as possible time. It is not difficult to see the analogy of this situation to that where an animal searches for its food in patches of different sizes. Imagine a bee searching for nectar in clumps of flowers, each of which contains only one or no flower with nectar. Checking all the flowers of each clump requires different amounts of time because the number of flowers on each branch varies. The bee has to decide when to leave the clump it has been dwelling on and when to fly to the next clump. The bee in this role faces a problem with similar structure as the human observer confronted with a visual search task. Given this analogy, one may speculate that patch leaving and search abortion obey similar principles. After all, foraging was the major task that visual search was originally applied to solve. It is possible that visual search still relies on the same programmed mechanisms as for the purpose of foraging.

Whereas the study of the termination problem in visual search just began about two decades ago, animals' foraging behavior has been studied extensively

for more than five decades (for a review, see Stephens & Krebs, 1986). Today, foraging study has been developed into a fruitful field of behavioral ecology, distinguishing itself with established theory systems proven to be able to predict foraging behavior of various species. Until anthropologists invoked the marginal value theorem (Charnov, 1976) to explain decision making of humans (e.g., Metcalfe & Barlow, 1992), foraging theory has been solely tested on animals. In visual search literature, foraging theory was first used by Cain et al. (2012) to predict the time when human subjects decide to quit a multiple-target search. Since then, its theoretical value for the termination problem in visual search (above all multiple-targets or hybrid search) has attracted the attention of many researchers (e.g., Ehinger & Wolfe, 2016; Wolfe, 2012; Wolfe et al., 2018; Zhang et al., 2015).

Although foraging theory succeeds in predicting foraging behavior across a wide range of species, it is intended to describe the formative behavior instead of the normative. The heart of foraging theory is to use rational analysis to formally answer the question "Given the environment, when should a forager stop looking for food in a patch and move to another?" Beginning with a series of assumptions on the environment structure (e.g., how are the prey items distributed within and between patches) and the forager (e.g., energy cost of traveling between patches; how is the time needed to find the next prey item distributed), it calculates the mean reward rate (net energy gain per time unit) under all possible giving-up policies (a decision rule which specifies when to give up). Then the optimal giving-up policy is the one which maximizes the mean reward rate. One of the key results is expressed by the marginal value theorem (Charnov, 1976): The optimal time to leave the current patch is the moment when the instantaneous intake (reward) rate of the current patch falls below the average intake rate of the environment. Later, McNamara (1982) generalized this result to find the optimal foraging policy in a stochastic environment and formalized it using the concept of potential functions: Continue foraging while the potential is positive and leave the patch when it falls to zero. The potential can be understood as "the relative advantages of being in the various possible states given that the future behavior is optimal." (McNamara, 1982, p. 283)

Appealing as the idea of optimal foraging, there are still several open questions

when applying optimal foraging theory to the context of (standard) visual search. Here, I point out three major differences between the standard visual search task and a foraging scenario (or assumptions of foraging theory). First, the visual scene in laboratory often consists of well-defined, discrete visual units (items) which observers can count, whereas a forager often faces continuous visual (or regarding whichever senses) scene so that the discrete units in a counting mechanism cannot always be well-defined (Imagine a bird looking for nymphs on a tree, what should it count as units that it identify as not-a-nymph?). As a consequence, using the time elapsed as the termination criterion is necessary in foraging theory and the major results are all expressed in a timing account. As discussed in Section 4.2.2.6, a timing mechanism is inconsistent with the empirical data. Second, observers are aware of the fact that there is either one or no target in the display, while animals do not necessarily have this information. It implies that the search will be terminated within finite time, no matter the display contains a target or not (at the latest when all items have been searched). In foraging theory, however, the dwell time on a patch is allowed to be infinite (although not quitting at all will never be the optimal policy). Third, responding "no" on a target-absent trial is also a correct response and thus contributes to accuracy ("reward" in visual search task) to the same extent as finding a target on a target-present trial. In other words, a correct rejection should be viewed as a reward providing the same "energy value" as a hit. In contrast, animals do not score for correctly classifying a patch as containing no food. These differences make the mathematics of foraging theory not directly applicable to standard visual search. However, we can derive the optimal termination rule using the same approach as in foraging theory.

4.3.4 The mean reward rate under a stopping policy

The first step is to calculate the mean reward rate. Because searchers are instructed to "respond as accurately as possible", using the accuracy as the indicator for reward appears to be natural. Although the other requirement is "respond as fast as possible", the number of examined items (counting) appears to be a better measure of effort than the time spent so far (timing) when modeling visual search termination, as discussed in Section 4.2.2.6. Therefore, the number of examined

items is used as the indicator for effort and the mean reward rate is defined here as the ratio of the latter to the former.

Since the target prevalence is 50% in standard visual search, it is assumed a target prevalence of 50% in the following reasoning for the sake of convenience. Similar calculations can be done to find the optimal quitting rule in situations where the target prevalence deviates from 50%. Furthermore, for the sake of simplicity, the observer is assumed to be not aware of his or her processing accuracy (represented by p_1 and p_2). The observer's quitting behavior is assumed to be determined by the reward rate calculated from his or her perspective using only information the observer has. That is, the reward rate is calculated without taking genuine processing errors (associated to p_1 and p_2) into account. This assumption is insofar justified as genuine processing errors in simple visual search tasks are very rare (see Section 4.4). The observer may feel at eased to ignore them.

Based on the revised Assumption 3, a giving-up policy can be formulated as follows: "Stop and response 'yes' once the target is found, otherwise stop and response 'no' after examining l items ($0 \leq l \leq k$)." In this way, the set of the stopping policies is bijectively mapped to the set of all possible l. After specifying the giving-up policies, we can now calculate the expected number of correct responses under each giving-up policy.

Consider first target-present trials. Due to the assumption of perfect processing, the observer will response "no" only if he or she finishes searching l items and all of them are distractors. In all other situations he or she will response "yes" (including the situation where the l-th item is the target), because he or she will not search more than l items. Therefore, the probabilities of "yes" and "no" responses on target-present trials are

$$\begin{aligned}\mathbb{P}(\text{“yes”} \mid \text{TP}) &= \sum_{i=1}^{l} \mathbb{P}(\text{the target is the } i\text{-th item examined} \mid \text{TP}) \\ &= \sum_{i=1}^{l} \frac{1}{k} = \frac{l}{k}\end{aligned}$$

and

$$\begin{aligned}\mathbb{P}(\text{“no”} \mid \text{TP}) &= \mathbb{P}(\text{examining and rejecting } l \text{ distractors} \mid \text{TP}) \\ &= \frac{\binom{k-1}{l}}{\binom{k}{l}} = \frac{k-l}{k},\end{aligned}$$

respectively. Let $N_k(l)$ denote the random variable of the number of items examined until termination (regardless of the outcome response) under this policy on a trial with k items. The mean number of items searched under this policy on target-present trials is then

$$\begin{aligned}\mathbb{E}(N_k(l) \mid \text{TP}) &= \sum_{i=1}^{l} \mathbb{P}(\text{quitting after examining } i \text{ items} \mid \text{TP}) \cdot i \\ &= \sum_{i=1}^{l} \frac{1}{k} \cdot i + \frac{k-l}{k} \cdot l \\ &= \frac{1}{k} \cdot \frac{l(l+1)}{2} + \frac{(k-l)l}{k} \\ &= \frac{l^2 + l + 2kl - 2l^2}{2k} = \frac{2kl - l^2 + l}{2k}.\end{aligned}$$

Consider now target-absent trials. Due to the assumption of perfect processing, every target-absent trial will always be terminated by a “no” response after searching l items. That is,

$$\begin{aligned}\mathbb{P}(\text{“yes”} \mid \text{TA}) &= 0; \\ \mathbb{P}(\text{“no”} \mid \text{TA}) &= 1.\end{aligned}$$

Accordingly, the mean number of items searched under this policy on target-absent trials is

$$\mathbb{E}(N_k(l) \mid \text{TA}) = l.$$

Let $R_k(l)$ denote the random variable of the number of correct responses under the giving-up policy l on a trial with k items. The expected number

of correct responses on any arbitrary trial of the experiment with 50% target prevalence is then given by

$$\begin{aligned}\mathbb{E}(R_k(l)) =& \mathbb{P}(\text{“yes”} \mid \text{TP}) \cdot \mathbb{P}(\text{TP}) \\ &+ \mathbb{P}(\text{“no”} \mid \text{TA}) \cdot \mathbb{P}(\text{TA}) \\ &= \frac{l}{k} \cdot \frac{1}{2} + 1 \cdot \frac{1}{2} = \frac{l+k}{2k}.\end{aligned} \tag{4.7}$$

The expected number of items searched under this policy on any arbitrary trial of the experiment is given by

$$\begin{aligned}\mathbb{E}(N_k(l)) =& \mathbb{E}(N_k(l) \mid \text{TP}) \cdot \mathbb{P}(\text{TP}) \\ &+ \mathbb{E}(N_k(l) \mid \text{TA}) \cdot \mathbb{P}(\text{TA}) \\ &= \frac{2kl - l^2 + l}{2k} \cdot \frac{1}{2} + l \cdot \frac{1}{2} = \frac{4kl - l^2 + l}{4k}.\end{aligned}$$

It follows that the mean reward rate under the policy “stop once the target is found, otherwise stop after examining l items” is

$$\gamma_k(l) = \frac{\mathbb{E}(R_k(l))}{\mathbb{E}(N_k(l))} = \frac{l+k}{2k} \cdot \frac{4k}{4kl - l^2 + l} = \frac{2(k+l)}{4kl - l^2 + l}. \tag{4.8}$$

For each given set size k, the reward rate $\gamma_k(l)$ is a function of l. The optimal policy is the one whose l maximizes the reward rate $\gamma_k(l)$. Following the standard procedure of finding extrema, we can derive $\gamma_k(l)$ with respect to l:

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}l}\gamma_k(l) &= \frac{2(4kl - l^2 + l) - (4k - 2l + 1)(2k + 2l)}{(4kl - l^2 + l)^2} = \frac{-8k^2 + 4kl + 2l^2 - 2k}{(4kl - l^2 + l)^2} \\ &\leq \frac{-8k^2 + 4k^2 + 2k^2 - 2k}{(4kl - l^2 + l)^2} = \frac{-2k^2 - 2k}{(4kl - l^2 + l)^2}.\end{aligned}$$

Obviously, the derivative is strictly negative for all $k \in \mathbb{N}$. This means that for every natural number k, $\gamma_k(l)$ is a monotone decreasing function in l and is maximal when l takes its smallest possible value. This result might seem surprising at first glance but it makes sense: If the observer does not search at all but simply always responses “no”, then he or she will achieve an accuracy of 50% while having no item to search at all! In this case, the reward rate is — not surprisingly — by definition infinite. This seemingly absurd result is due to the

assumption that a correct rejection has the same value as a hit and it does not necessarily require any search effort at all.

But this is surely not what is usually observed in a visual search experiment. For simple tasks, most participants complete the task conscientiously so that they make under 10% errors. Obviously, they understand that the experimenter is interested in their search behavior and they did perform visual search in order to provide relevant behavioral data.

A possibility is that participants may not be told about or not completely rely on the information of target prevalence. If this is the case, they may update their belief of the target prevalence continuously throughout the experiment by their search experience. To collect information about target prevalence, it is necessary to perform search. Previous studies on the target prevalence effect show that people are sensitive to fluctuations of target prevalence and adjust their search behavior accordingly without being explicitly informed about the prevalence (e.g., Ishibashi et al., 2012; Wolfe & Van Wert, 2010). Thus, in experiments with a large number of trials, participants should be able to learn about the true target prevalence in the several trials at the beginning without being told explicitly. Their behavior should not differ notably from that when they know the information from the beginning on. Thus, we can assume that target prevalence has an influence on the search termination.

Another possibility is that a correct rejection may have a lower subjective value than a hit, which prevents searchers giving up the target-present trials. After all, responding merely "no" regardless of what is shown in the display is hard to be considered as a performance.

Finally, it is possible that searchers have a subjective standard of acceptable performance, i.e., how high their average accuracy at least must be to allow themselves to "get it done quickly." Such a speed-accuracy trade-off is often observed in behavioral data of cognitive tasks.

It should be noted that the number of items examined underestimates the actual effort of completing visual search tasks. This measure neglects for instance the effort for executing the response (key press) and waiting for the next stimulus display to appear. This kind of effort (residual effort) corresponds to the travailing time in optimal foraging theory. If this kind of effort is taken into account as

part of the total effort, $\gamma_k(l)$ will have a upper bound because the denominator will not be zero for any l in $[0, k]$. In this case, always responding "no" without searching any item may not yield the maximum reward rate. The optimal l will depend on the relation between the residual effort and the effort of examining visual stimuli. Once this relation is specified, the optimal stopping policy can be derived. This is a topic reserved for future research. In this dissertation, I follow another approach.

4.3.5 Optimal termination rule

Because the speed-accuracy trade-off account appears simplest and most plausible, I adopt this account for modeling here. That is, searchers try to maximize the mean reward rate under the restriction of achieving an acceptable accuracy. Each individual may have a different subjective standard of acceptable accuracy. The question is, given this minimal performance requirement, how does it influence the termination rule? Remember that we derived the reward rate as a function of the maximal number to search before quitting to find the l that maximizes the reward rate $\gamma_k(l)$. Taking a minimal accuracy requirement into account is to place a restriction when looking for the l which maximizes γ_k, and this restriction should also be a function of l.

Let z denote the minimum acceptable accuracy. Consider first a visual search experiment with only one set size level. For any $k \in \mathbb{N}$ given, finding the maximum of $\gamma_k(l) = \frac{2(k+l)}{4kl-l^2+l}$ under the restriction of $\mathbb{E}(R_k(l)) = \frac{l+k}{2k} \geq z$ is easy because $\gamma_k(l)$ is decreasing in l while $\mathbb{E}(R_k(l))$ is increasing in l. Thus, $\gamma_k(l)$ takes its maximum at the smallest possible l, i.e., when $l = k(2z-1)$. It follows that the proportion of items to search is a linear transformation of z: $r = \frac{l}{k} = 2z - 1$. This ratio will be constant if z is constant. But this result only applies to situations in which k is constant throughout the entire experiment. In standard visual search experiments, the set size k is typically manipulated at more than one level, which makes the situation different. An important fact is that if the observer always searches a constant proportion of items $r = \frac{l}{k} = const.$, the reward rate is lower for larger k, since $\gamma_k(l) = \frac{2(k+l)}{4kl-l^2+l} = \frac{2(1+r)}{k(4r-r^2)+r}$ is decreasing in k for constant r. This means that in order to achieve the same accuracy, the search is less efficient on larger trials. If there are m different set size levels $(k_1, \ldots, k_m)$ realized in the

experiment and the observer searches according to the stopping rule $(l_1, \ldots, l_m)$, the total expected number of correct responses is the sum of expected correct responses across trials with different set sizes

$$\mathbb{E}(R_{(k_1,\ldots,k_m)}(l_1, \ldots, l_m)) = \frac{1}{m} \sum_{i=1}^{m} \mathbb{E}(R_{k_i}(l_i)).$$

The same applies to the total expected number of items to search:

$$\mathbb{E}(N_{(k_1,\ldots,k_m)}(l_1, \ldots, l_m)) = \frac{1}{m} \sum_{i=1}^{m} \mathbb{E}(N_{k_i}(l_i)).$$

The maximum of the mean reward rate of the entire experiment will not be achieved at $(l_1, \ldots, l_m)$ if each of $l_1, \ldots, l_m$ alone minimizes the reward rate on those trials with the corresponding set size under the restriction of the same acceptable reward rate z. The reason is that one can always trade for a larger increase of the reward rate on larger trials by relaxing the acceptable accuracy on larger trials while restricting that on smaller trials. To illustrate this phenomenon, take the simple example of a visual search experiment realizing two levels of set sizes: $k_1 = 10$ and $k_2 = 20$. Assume that there are 100 trials for each set size and the total acceptable accuracy is 0.9. Consider two stopping rules: $(l_1, l_2) = (8, 16)$ (which has a fixed ratio $r = 0.8$) and $(l_1, l_2) = (10, 12)$. Both stopping rules will obtain an expected total accuracy of 0.9 according to Equation (4.7). However, the total mean reward rate under the stopping rule $(8, 16)$ is about 0.092, while it is about 0.099 under the stopping rule $(10, 12)$, according to Equation (4.8). On one small trial and one large trial, the total expected number of correct responses is the same under these two stopping rules (1.8). Using the stopping rule $(10, 12)$, the observer will have to search more items on the small trial (7.75 instead of 6.6) on average, but in exchange, he or she will have to search much fewer on the large trial (10.35 instead of 13).

The discussion above reveals the essence of the problem: In an experiment containing trials with different set sizes, the accuracy goal z and consequently its linear transformation r, i.e., the proportion of items to search, must be adjusted to the set size level to achieve a higher search efficiency. Therefore, the key is to allocate the acceptable accuracy to trials with different set sizes.

Let us formalize the problem of optimizing the mean search efficiency throughout the entire experiment as a mathematical programming problem. The objective function (i.e., the quantity to be optimized) is the mean reward rate of the entire experiment

$$\gamma_{(k_1,\ldots,k_m)}(l_1,\ldots,l_m) = \frac{\mathbb{E}(R_{(k_1,\ldots,k_m)}(l_1,\ldots,l_m))}{\mathbb{E}(N_{(k_1,\ldots,k_m)}(l_1,\ldots,l_m))} = \frac{\frac{1}{m}\sum_{i=1}^{m}\mathbb{E}(R_{k_i}(l_i))}{\frac{1}{m}\sum_{i=1}^{m}\mathbb{E}(N_{k_i}(l_i))} = \frac{\sum_{i=1}^{m}\frac{l_i+k_i}{2k_i}}{\sum_{i=1}^{m}\frac{4k_il_i-l_i^2+l_i}{4k_i}}$$

The most important constraint is the acceptable expected accuracy

$$\mathbb{E}(R_{(k_1,\ldots,k_m)}(l_1,\ldots,l_m)) = \frac{1}{m}\sum_{i=1}^{m}\mathbb{E}(R_{k_i}(l_i)) = \frac{1}{m}\sum_{i=1}^{m}\frac{l_i+k_i}{2k_i} \geq z$$

In addition, there are natural constraints on l_i: $0 \leq l_i \leq k_i \, \forall i = 1,\ldots,m$. The optimization problem in the general form is thus

$$\begin{aligned}
&\min f(l_1,\ldots l_m) = -\frac{\sum_{i=1}^{m}\frac{l_i+k_i}{2k_i}}{\sum_{i=1}^{m}\frac{4k_il_i-l_i^2+l_i}{4k_i}} \\
&\text{s.t. } g(l_1,\ldots l_m) = -\sum_{i=1}^{m}\frac{l_i+k_i}{2k_i} + mz \leq 0 \\
&\quad l_i - k_i \leq 0 \\
&\quad -l_i \leq 0, i = 1,\ldots m.
\end{aligned}$$

The essential result is that the optimal solution of this problem depends on the range of the acceptable accuracy z and the number of set size levels m. Here I focus on the case $mz-(m-1) \geq 0.5$. Since the visual search task in the laboratory usually employs stimuli that are easy to identify, the overall performance of participants is usually very high (over 90% correct). It is plausible to assume that participants set a high subjective acceptable accuracy when facing such easy tasks. Furthermore, the number of set size levels realized in an experiment is usually very limited (not more than 4 levels). Thus, the condition $mz-(m-1) \geq 0.5$ should be fulfilled in most cases. Under this condition, the optimal solution is

$$(l_1^*, l_2^*, \ldots, l_m^*) = (k_1, k_2, \ldots, k_{m-1}, k_m(2mz-2m+1)), \text{ for } k_1 \leq k_2 \leq \ldots \leq k_m$$

The optimal solution can be equivalently characterized by the proportion of items to search

$$(r_1^*, r_2^*, ..., r_m^*) = (1, 1, \ldots, 1, 2mz - 2m + 1),$$

by the expected accuracy for each set size

$$(z_1^*, z_2^*, ..., z_m^*) = (1, 1, \ldots, 1, mz - (m - 1))$$

or by the confidence criterion

$$(s_1^*, s_2^*, ..., s_m^*) = \left(1, 1, \ldots, 1, \frac{1}{1 + 2m(1 - z)}\right).$$

The optimal solution can be interpreted in the following algorithmic manner: First guarantee the maximal possible accuracy on all trials except those with the largest set size by performing an exhaustive search on these trials. If one still needs to achieve an accuracy of more than 50% on the largest trials to fulfill the acceptable total accuracy, then search on those trials can be terminated so early that the minimum necessary contribution to the total average accuracy can be achieved.

The optimal solution under other conditions follows a similar structure. In case $mz - (m - 1) < 0.5$, meaning that the observer tolerates lower total accuracy, but not too low such that $mz - (m - 2) - 0.5 \geq 0.5$ still holds, the optimal solution is to search exhaustively on all trials with the smallest $m - 2$ set size levels, to give up all trials with the largest set size level (always responding "no"), and to search only the necessary proportion on all trials with the second largest set size level. For other ranges of z and other values of m, the optimal solution follows analogous polarization of the proportion to search.

It does not take us a second thought to realize that no human subject behaves exactly according to the optimal termination rule in reality. An observer adopting the optimal stopping rule would have perfect performance on small trials and we would observe a jump of the miss rate for the largest set size (if the overall performance is high). Even if we take genuine processing errors into account, performance on small trials would be close to perfect, the miss rates would decrease rather than increase for the smaller set sizes and jumps would still be expected.

Is it wrong then that the principle of efficiency optimization underlies incomplete search? It seems too early to jump into that conclusion. In behavior ecology, the exact theoretical optimum is hardly observed empirically. Although optimality is rarely obtained, the behavior animals display is still considered optimal in the sense that it shows statistical tendency that is in line with the prediction of the optimal solution. After all, the theoretical optimal solution is the result of a formal analysis aiming to describe a formative behavior.

4.3.6 Approximatively optimal termination rule under uncertainty

Many factors may contribute to a deviation from the optimal behavior. As McNamara (1982) commented, "It is the experimenter who must solve the mathematical problem, not the animal (p. 286)". In fact, most participants would not deliberately think about the optimal termination rule. Even if they try to find out the most efficient way to respond, the exact optimal solution cannot be obtained without some calculation effort. People might just follow the intuition that spending the same amount of time on small trials is more beneficial than on large trials, and reduce the proportional effort on large trials accordingly. Moreover, if the searcher pursues a perfect performance on small trials, the responses on these trials will not be "as quickly as possible" anymore, which may make her hesitate to do so, resulting in premature termination on the medium-sized trials.

More importantly, the searcher does not possess complete information ahead of and during the experiment. Participants may be told about the target prevalence and the total number of trials, but it is unconventional to inform them about the number of set size levels and how they are distributed. This knowledge is necessary to derive the optimal termination rule *before* the game is on. Although the searcher can form her estimations after a few trials, this remains uncertain for her until the end of the experiment. If the set size levels are not evenly distributed, the optimal termination rule will no longer be the optimal solution. Moreover, the searcher may not be able to always keep her current accuracy in mind accurately; he or she may also raise the acceptable accuracy to make

up for genuine processing errors. In sum, participants do not seem to stick to a predefined termination criterion but adjust their behavior in a dynamic way during the experiment. This is in line with the observation that subjects spent more time on the subsequent trial, when they had missed the target on the previous trial (Chun & Wolfe, 1996).

To explore the relation between the actual behavior and the optimal behavior, it is helpful to study the characterization of the optimal solution by the confidence criterion s derived in Section 4.3.2. Here I keep focusing on the most common case of very high overall performance. The major pattern of the optimal confidence criterion $(s_1^*, s_2^*, ..., s_m^*) = (1, 1, ..., \frac{1}{1+2m(1-z)})$ is a flat line extending through the smaller set sizes with a kink at the end. This pattern can be captured by the upper part of a mirrored "S"-shaped curve. Using a smoothing approach, we can approximate the optimal confidence criterion by a sigmoid function. I illustrate the idea using the study of Wolfe et al. (2010) as an example. In their experiment using conjunction search task, four set size levels were realized: $(3, 6, 12, 18)$. Assuming that an observer sets her acceptable accuracy at $z = 0.95$, the theoretically optimal confidence criterion for these set sizes are $(s_1^*, s_2^*, s_3^*, s_4^*) = \left(1, 1, 1, \frac{5}{9}\right)$. As shown in Figure 4.3, these points lie very close to the upper part of the graph of a horizontally reflected logistic function $f(x) = \frac{1}{1+\exp(0.45*(x-20))}$. As the steepness or/and the inflection point of the reflected logistic curve change, it deviates more or less from the optimal confidence criterion.

The upper part of a horizontally reflected logistic function displays a smooth approximation of the optimal confidence criterion. The nearer it approaches the optimum, the larger the extent to which an observer adopts an efficiency maximizing termination rule. Such a reflected logistic function ranging in $(0, 1)$ is defined by the formula

$$f(x) = \frac{1}{1 + \exp\big(\beta x + \alpha\big)} = \frac{1}{1 + \exp\Big(\beta(x + \frac{\alpha}{\beta})\Big)}, \text{ whereby } \beta > 0, -\frac{\alpha}{\beta} > x \text{ for all } x.$$

It is thus characterized by two parameters: the steepness of the curve β and the inflection point $-\frac{\alpha}{\beta}$. The restriction $\beta > 0$ makes the "S"-shaped curve horizontally reflected and the restriction $-\frac{\alpha}{\beta} > x$ for all x guarantees that the

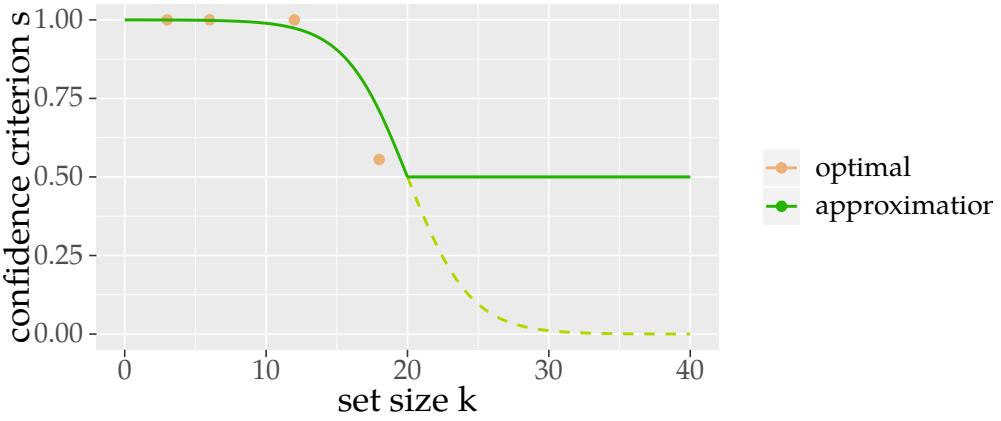

Figure 4.3: The theoretical optimal confidence criterion and the approximation by a logistic function.

inflection point locates to the right of the largest set size level, so that the function at the largest set size takes a value larger than 0.5.

The steepness can be interpreted as the extent to which the search is efficiency-oriented. The larger β, the steeper the curve, i.e., the more the confidence criteria are polarized. The inflection point can be interpreted as the set size level from which the observer would give up search on all equal sized or larger trials, skipping them with a negative response. Thus, the confidence criterion must be set constant to $\frac{1}{2}$ from the inflection point on.

An advantage of this approximation is that it can be generalized to approximate the optimal solution in all other situations where the range of the acceptable accuracy is different. If the acceptable total accuracy is lower, i.e., z satisfies $mz - (m-1) < 0.5$, the optimal solution changes its form. Nevertheless, if we focus on the smallest $m - \tilde{m}$ set sizes such that $\tilde{m}$ is the smallest non-negative integer satisfying $mz - (m - \tilde{m} - 1) \geq 0.5$, then the optimal solution restricted on them has exactly the same structure as the optimal solution in the most common case. This pattern can be captured by moving the inflection point of the reflected logistic curve to the left and setting all the values on the right of the inflection

point to $\frac{1}{2}$. In this way, I propose to model the confidence criterion by the function

$$s(k) = \begin{cases} \dfrac{1}{1+\exp\big(\beta k+\alpha\big)}, & \text{if } 1 \le k \le -\dfrac{\alpha}{\beta}, \\ \dfrac{1}{2}, & \text{if } k > -\dfrac{\alpha}{\beta}, \end{cases} \quad \text{whereby } \beta > 0. \tag{4.9}$$

Accordingly, I model the termination rule by

$$l(k) = \begin{cases} k\big(1-\exp\big(\beta k+\alpha\big)\big), & \text{if } 1 \le k \le -\dfrac{\alpha}{\beta}, \\ 0, & \text{if } k > -\dfrac{\alpha}{\beta}, \end{cases} \quad \text{whereby } \beta > 0. \tag{4.10}$$

Note that in the most common case where z satisfies $mz - (m-1) \ge 0.5$ (the inflection point is located to the right of the largest set size), my model predicts the proportion of the items to search to be $r(k) = 1 - \exp\big(\beta k + \alpha\big)$, which is an decreasing function in k. Thus, it satisfies the necessary condition of the error modeling. For α, β, p_1 and p_2 in normal ranges, it is convex and increasing in k.

So far, I have provided a solution to the problem of modeling search termination. For the miss rate, we can now substitute l in Equation (4.4) by $k\big(1-\exp\big(\beta k+\alpha\big)\big)$ and obtain

$$\begin{aligned} \mathbb{P}(\text{miss}) =& (1-p_1)^{k-k\exp\left(\alpha+\beta k\right)-1} \\ & \cdot \Big((1-p_1)\exp\big(\alpha+\beta k\big) + p_2\big(1-\exp\big(\alpha+\beta k\big)\big)\Big). \end{aligned} \tag{4.11}$$

Despite some technical issues (will be discussed in Section 4.5), it appears to work well. Simulations show that it is able to predict miss rates that are close to the miss rates observed in the experiments by Wolfe et al. (2010), capturing the basic pattern of a convex, increasing form. However, the predictions seem to contain a minor incompatibility. The growth in the miss rate simulated by the model is slightly slower than its growth in the empirical data, such that it often predicts higher miss rates than observed in the data for medium set sizes. This might be considered as a second-order effect, but taking Requirement b) into consideration, one may view this issue as an indicator for the necessity of modeling the genuine processing errors and revising Assumption 2.

For Requirement b) to hold, the empirical false alarm rates have to remain on roughly the same level, or even decline slightly as k increases. Under the termination rule proposed here, the false alarm rate is

$$\mathbb{P}(\text{false alarm}) = 1 - (1 - p_1)^l = 1 - (1 - p_1)^{k\left(1-\exp\left(\beta k+\alpha\right)\right)}. \tag{4.12}$$

It is possible (and also likely) that l increases with increasing k. This means that the proposed termination rule predicts false alarm rates to be increasing as the set size increases. This growth is not negligible, even though p_1 is usually very small such that the growth is approximately linear. This inconsistency indicates that incomplete search alone is not sufficient to explain the error patterns observed in visual search experiments.

4.4 Imperfect processing

The key of fulfilling Requirement b) of the empirical accuracy data is to find out how the probability of failure in processing one single distractor, i.e., $\mathbb{P}(\text{“}T\text{”} \mid D) = p_1$, changes depending on the set size k. Unlike the miss errors, whose occurrence relies on both types of processing errors and the termination rule (as reflected in Equation (4.11)), false alarm errors have nothing to do with the processing error of classifying the target as an distractor, because there is no target at all on a target-absent trial. Thus, from the pattern of false alarm errors, we can draw better inferences about the genuine processing errors.

Why do false alarm rates remain on the same level for different set sizes? If the occurrence of a false alarm error on a single trial is based on one single processing failure of classifying a distractor as the target (because the search will be terminated immediately the first time a distractor is classified as the target, there will not be a second chance for taking another distractor for the target mistakenly), then assuming an i.i.d. nature of these failures regardless of the set size necessarily leads to the prediction of an increasing function of false alarm probability on set size — except for the case where all failures occur with probability zero. The reason is that under a set-size-invariant probability of processing failure, the number of items to search also gets larger as the set size gets larger (assuming the set sizes are in a range for which the observer is not

willing to give up searching completely on trials with a certain set size) and the probability that the processing of one of these distractors fails also gets larger.

If we reject the assumption of i.i.d. failures, what could be an alternative assumption? There is no obvious theoretical or empirical argument against the stochastic independence. Moreover, the assumption of stochastic independence is simply basic — it would be very difficult to do any calculation without it. If we retain the notion of stochastic independence, the only remaining option is to relax the assumption that p_1 and p_2 are invariant to the set size.

4.4.1 Threshold adaptation

Allowing p_1 and p_2 to change depending on k is also the approach that Zenger and Fahle (1997) adopted to modify their model (described in Section 4.1.1) in lights of the inconsistency of predicted and empirical false alarm rates. Using SDT to describe how a single stimulus is classified, they proposed that observers may adjust their discrimination thresholds to the set size so that it minimizes certain risk (the probability of the misclassification weighted by incidence probability of distractors and the target, or the expected overall error rate). Both lead to a shift in the more conservative direction when k rises, such that p_1 becomes smaller and p_2 larger.

Such an adaptation is theoretically appealing and convincing, yet difficult to solve for two reasons. One is that the two free parameters in their model cannot be estimated simultaneously. It is necessary to hold the discriminability constant to estimate the optimal position of the discrimination threshold. The second is that even under a fixed discriminability, there is no analytical method to find the position of the discrimination threshold that minimizes the objective function. Estimating the discrimination threshold here is equivalent to solving equations involving the density functions of the target and the maximal non-target signals, which can only be done using numerical methods. Thus, the estimation contains two steps. With some computational expenses, this estimation method may work well given that only these two parameters need to be determined. However, as pointed out in Section 4.1.1, there are systematic deviations of their model predictions from the empirical patterns, which is one of the reasons why we need to model the search termination. Note that because the critical equations for the

estimation of the threshold are not analytically solvable, the model of Zenger and Fahle (1997) is not able to express p_1 and p_2 (and consequently, the predicted error rates) as a function of its two parameters explicitly. If it was able to do this, we would be able to substitute p_1 and p_2 in Equation (4.11) and (4.12), obtaining an explicit relation between all parameters, especially the termination rule l, and the predicted error rates. This means that if we take premature termination into account and apply the approach of Zenger and Fahle (1997) to model genuine processing errors, the parameter estimation would be computationally expensive to implement and there is no guarantee for the quality of the estimates.

Therefore, I adopt another approach instead. Ideally, the probabilities of genuine processing errors p_1 and p_2 can be described as a function of k explicitly. As discussed in the last section, Equation (4.12) indicates that for constant p_1, the predicted probability of false alarms $\mathbb{P}$(false alarm) increases as k increases, because the power is increasing. Holding the power constant, $\mathbb{P}$(false alarm) is increasing in p_1. Thus, to compensate the increase caused by the increasing k, p_1 must decrease. This inference is in line with the predictions of the model of Zenger and Fahle (1997) that the discrimination threshold shifts in the conservative direction, resulting in a smaller p_1. For very small, positive p_1, as it is usually the cases in empirical studies, $(1-p_1)^l \approx 1 - lp_1$ holds. This means that the reduction of p_1 must be at least as fast as $\frac{1}{l}$, i.e., p_1 must be inversely proportional to l or smaller.

4.4.2 Power law relation

To explore the relation between p_1 and k, I use the datasets of Wolfe et al. (2010). First, I obtain a naive estimate of p_1 for each set size level by solving Equation (4.3) for p_1, whereby l is an estimate obtained from the empirical miss rates. If p_1 is inversely proportional to l, then the product of the naive estimates of p_1 and l must be approximately constant. However, this product declines systematically. Even if we substitute l by k, the result only changes to a negligible degree. This observation indicates that an inverse proportional relation underestimates the speed at which p_1 declines with k.

Considering that the speculation of an inversely proportional relation essentially describes a power law, I plot the log-transformed naive estimates of p_1

as a function of the log-transformed set size k. On the log-log plot, the naive estimates of p_1 clearly form a straight line (Figue 4.4). This is observed for both conjunction search and spatial configuration tasks, with only minimal deviation especially for conjunction search tasks. If the log-transformed of a variable can be described by a straight-line of the log-transformed of another variable, i.e., $\log(v_1) = a + b\log(v_2)$, then the former itself can be described by a monomial of the latter, i.e., $v_1 = e^a v_2^b$. Thus, this observation leads me to hypothesize that the relation between p_1 and k follows a power law, i.e.,

$$p_1 = a_1 k^{-b}, \text{ whereby } a_1 > 0, b > 0. \tag{4.13}$$

I restrict the power of k to be negative because p_1 should decline as k increases, as concluded before. The slope of both straight-lines in Figure 4.4 is slightly smaller than -1 (-1.28 and -1.15, respectively), which is in line with the speculation that p_1 declines at a higher speed than an inversely proportional would predict.

The observation of a power law relation here has not been documented in the literature. Although both data sets demonstrate patterns that are highly consistent with a power law, it should be born in mind that this observation is based on data for only four set size levels. Although the error rates of these data sets are supposed to be reliable because each single value is calculated based on about 5,000 trials of each task type, they reflect the aggregated data of 10 or 9 participants. When the log-log plot is created using individual data (each value is based on ca. 500 trials), a straight line is observed for only about a half of the participants (6 out of 10 for conjunction search, 4 out of 9 for spatial configuration). However, zero or extraordinarily low false alarm rates (< 0.005) for *some* of the set size levels in case of the other participants can explain that no straight line is observed. They lead to missing values or outliers in the estimate of p_1 (when the false alarm rate is zero, p_1 is estimated at 0 according to Equation 4.12 and its logarithm is not defined). In sum, a large amount of empirical data is needed to answer the question of how universal this observation is for visual search.

Furthermore, this discovery is an ad hoc analysis and not sufficient to conclude a convincing theoretical explanation. Power laws are observed in numerous natural and social phenomena of all kinds of contents but not fully

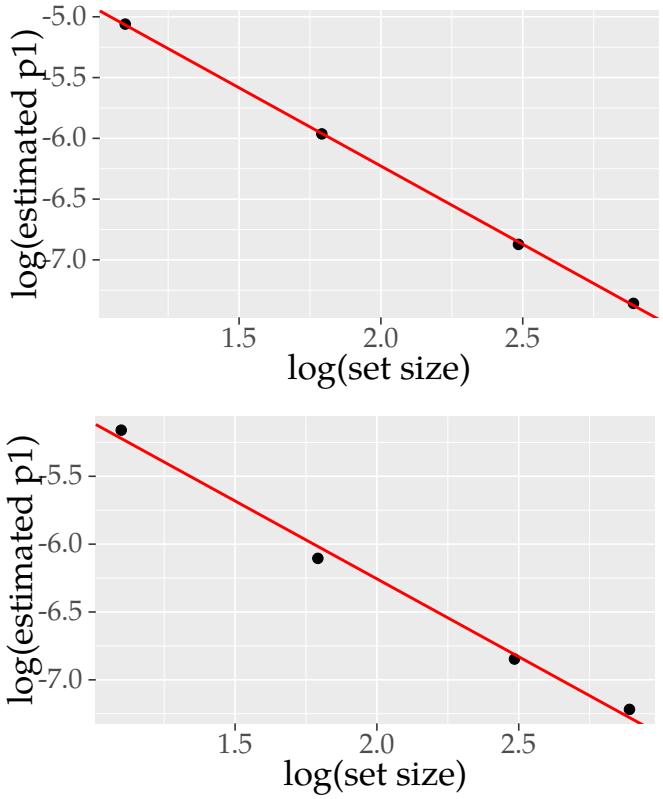

Figure 4.4: Log-log plot of the naive estimates of p_1 as a function of the set size k for conjunction search (top) and spatial configuration (bottom) in the study by Wolfe et al. (2010).

well understood. There are many mechanisms that can lead to a power law but so far no theorems that prove the necessary and sufficient conditions for power laws to be found. Although some power laws in physics point to specific mechanisms conclusively, it is not the case for most phenomena in other disciplines. Therefore, even if the power law can be found for various visual search data generally, additional evidence is required to allow justified inferences on the underlying mechanism.

Nevertheless, there are several possibilities worth considering. One might think of Stevens' power law and speculate that changes in p_1 may result from a distortion of the magnitude of the set size. In fact, set size must be an estimate based on our ability to roughly enumerate larger number of items (Dehaene, 2011; Krueger, 1984). But this speculation is not consistent with the

data. According to Krueger (1984), magnitude estimation is roughly $\propto k^{0.8}$. It is more difficult to discriminate large set sizes. That would lead to the prediction that p_1 will decline slower than inversely proportional to k, contradicting the observation. Considering that there may be some optimization principles driving the discrimination process of visual stimuli, another thinkable connection could be that many complex systems optimizing certain quantities generate power laws[4]. For example, maximizing entropy (Mandelbrot, 1953) and maximizing a risk-neutral utility function under certain constraints (e.g., Carlson & Doyle, 1999).

Considered from a signal detection perspective, the observation of a power law may indicate an adaptive mechanism to maintain detection performance in a background wit irregular and cluttered noise. Recall that p_1 is conceived as the false alarm probability of discriminating a single stimulus and its dependence on set size k is interpreted as an adaptation of a discrimination threshold. What could be the reason for keeping a flat false alarm rate in the decision regarding multiple stimuli by such an adaptation? In engineering, the property of maintaining an approximately constant false alarm rate does belong to an ideal binary classifier that is adaptive in general contexts. This property requires a threshold adaptation following a power law in some circumstances. For instance, Constant False Alarm Rate (CFAR) is an important principle for designing detection scheme according to which the radar decides whether or not the target is present. It is applied in an environment of varying background noise and clutter. If the power threshold is set constant, the radar's performance will be affected by the background noise level, resulting in increased number of false alarms in the presence of strong noise or interference. By adapting the power threshold above which the echo signal is classified as originating from the target, the radar's performance can be maintained regardless of the level of noise or interference. When the background is an even, regular surface, the background noise is often modeled as Gaussian noise. However, when the background is a rough, irregular, or cluttered surface, Gaussian noise does not provide an accurate model. Rather, the background noise is commonly modeled by a (generalized) Pareto distribution (e.g., Newman, 2005). Since the tail of a Pareto distribution (i.e., the survival function) follows

[4]However, the power laws in those contexts mostly refer to power law distribution of variables.

a power law[5], a threshold adaptation that follows a power law is necessary to achieve CFAR.

It is conceivable that the set size k correlates with the level of background noise, degree of clutter or interference, which can be characterized by a Pareto distribution. Observers adapt their discrimination threshold in a way such that their detection performance remains independent of the level of background noise, similar to the CFAR principle for radars. Perhaps it is even not necessary that a Pareto distribution describes the background noise in the display in a visual search experiment accurately. After all, it is not a typical environment in which humans perform visual search. Human observers are confronted with a visual search task primarily in natural, continuous, noisy and cluttered scenes. It is possible that a Pareto distribution characterizes the background noise in such a scene accurately, so that observers, learned or programmed, stick to the detection scheme that works well in everyday life even when facing an environment that is better characterized by another distribution (e.g., Gaussian noise).

4.4.3 Conjecture of another power law

Now we have a model for p_1. Together with the model for the termination rule l, we can solve Equation (4.11) for p_2 to obtain a naive estimate of p_2 because all other variables in the equation can be predicted so far. Unfortunately, no clear pattern is found in these naive estimates, as showed in Figure 4.5.

However, it is not to expect that solving Equation (4.11) for p_2 provides valid estimates of p_2 after all. This method treats l and p_1 as known, which are in fact not known or measured but rather estimated. Their estimates are based on the empirical miss rates and false alarm rates, which again are estimates of $\mathbb{P}$(miss) and $\mathbb{P}$(false alarm), respectively. Because the absolute value of p_2 is on a very small scale, even tiny fluctuations in the empirical miss rates will lead

[5]The cumulative distribution function of a Pareto distributed random variable is given by

$$F(x) = \begin{cases} 1 - \left(\frac{x_{min}}{x}\right)^{\alpha}, & x \leq x_{min} \\ 0, & x < x_{min} \end{cases}$$

The tail is given by $\mathbb{P}(X > x) = (x_{min})^{\alpha} x^{-\alpha}$.

to relatively large fluctuations in the naive estimate of p_2. To understand this, consider the partial derivative of p_2 with respect to $\mathbb{P}(\text{miss})$ and $\mathbb{P}(\text{false alarm})$ followed from Equation 4.11 and 4.12:

$$\frac{\partial p_2}{\partial \mathbb{P}(\text{miss})} = \frac{1}{1 - \exp\big(\alpha + \beta k\big)};$$

$$\frac{\partial p_2}{\partial \mathbb{P}(\text{false alarm})} = \frac{\exp\big(\alpha + \beta k\big)}{1 - \exp\big(\alpha + \beta k\big)} = 1 - \frac{1}{1 - \exp\big(\alpha + \beta k\big)}.$$

It follows from the assumption $1 \leq k \leq -\frac{\alpha}{\beta}$ (see Equation (4.10)) that

$$\frac{1}{2} \leq \frac{1}{1 - \exp\big(\alpha + \beta k\big)} \leq 1.$$

This implies that the partial derivative of p_2 with respect to $\mathbb{P}(\text{miss})$ and $\mathbb{P}(\text{false alarm})$ should be bounded by $[1, 2]$ and $[-1, 0]$. The naive estimates of p_2 are on a scale of 10^{-3}, as shown in Figure 4.5. The empirical miss rates are on a scale of 10^{-2}. This means that a fluctuation of about 5% to 10% in the empirical miss rates is sufficient to cause a fluctuation of 100% in the naive estimates of p_2. This imprecision can conceal the essential pattern of p_2. Note that the analysis above assumes that the estimation of l is uniquely determined by $\mathbb{P}(\text{miss})$. But in fact, the l used to obtained the naive estimates of p_2 is a heuristic value that leads to an approximate pattern to the empirical miss rates according to Equation (4.11).

Without an explicit theoretical assumption on the relation between p_1 and p_2, such as SDT, it is difficult to specify a relation between p_2 and the set size. The SDT approach does not provide an analytical solution, as discussed in Section 4.4.1. Thus, I opt for another approach. If p_1 decreases as a function of k, it is reasonable to assume p_2 to be increasing. Although we cannot observe p_1 and p_2 directly, we know that the Receiver Operating Characteristic (ROC) curve of a binary classifier usually has a positive tangent, i.e., the Detection Error Trade-off (DET) graph has negative slope. That is, for most classification system, the performance of identifying a target correctly decreases as its performance of rejecting a non-target correctly increases. If p_2 rises always at a faster speed

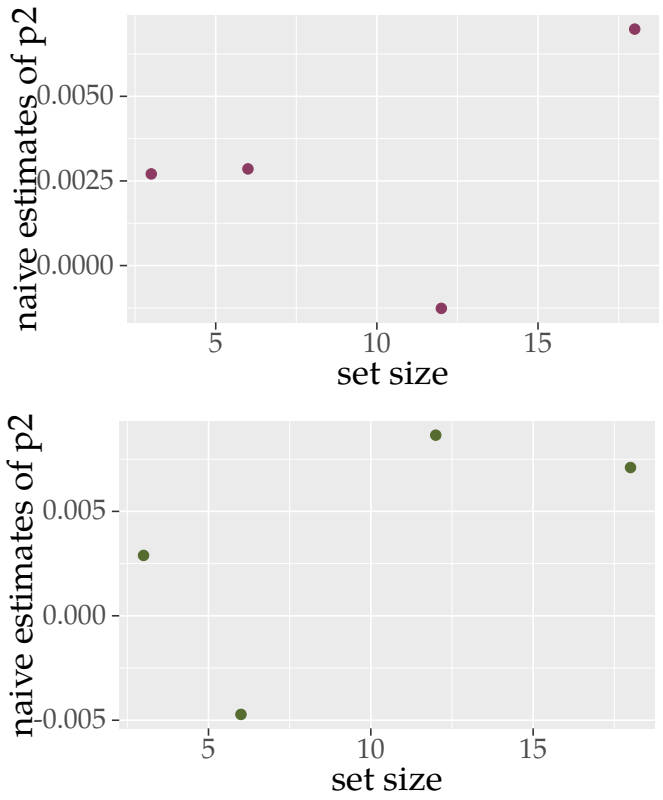

Figure 4.5: The naive estimates of p_2 as a function of the set size k for conjunction search (top) and spatial configuration (bottom) in the study by Wolfe et al. (2010).

than p_1 declines, the adaptation scheme observed for p_1 (power law) will lead to a much higher p_2 as the price for the adaptation. If p_2 rises always at a slower speed than p_1 declines, the ROC curve must be relatively flat in the neighborhood of the origin, which is not possible for a continuous ROC curve. Thus, in the area in which p_1 declines according to a power law, it is reasonable to assume that p_2 rises at a similar speed, i.e., following a power law with the inverse power. An additional reason to model p_2 in this way is that it reduces the number of parameter by one. I propose to model p_2 by

$$p_2 = a_2 k^b, \text{ whereby } a_2 > 0, b > 0 \tag{4.14}$$

In this way, we obtain analytical expressions for p_1 and p_2. Simulations show that the power law model incorporated in my termination rule model can mimic

the empirical error rates in the data of Wolfe et al. (2010) precisely (see 7.2 for the detailed results of fitting the model to these data).

From a modeling perspective, the advantage of the power law model is that it provides a simple analytical formula. For p_1, it appears reasonable because Equation (4.13) results in a small, positive value, in a range a probability is restricted to. But for p_2, the predicted value is not limited, which makes it inappropriate as a model for a probability. Therefore, Equation (4.14) may not be applicable for very large set size k. In fact, if the adaptation of p_1 still follows a power law when p_1 is very tiny, it must be followed by a cost of large p_2, or it will be very expensive to maintain the detection performance. In that case, the observer may switch to another adaptation scheme.

4.5 The complete error model

Based on the discussions so far, we can now formalize a complete model for errors in an visual search characterized by three assumptions:

Assumption 1. Servers provide imperfect service. They make two types of mistakes:

- **1.1.** Mistaking an ordinary customer for the target (server reports "target" given a distractor, denoted by "T" | D);
- **1.2.** Mistaking the target for an ordinary customer (server reports "distractor" given the target, denoted by "D" | T).

Assumption 2. For a fixed set size level, each server makes these two types of mistakes stochastically independently with the same (conditional) probabilities p_1 and p_2 respectively, i.e., $\mathbb{P}(\text{"}T\text{"} \mid D) = p_1$ and $\mathbb{P}(\text{"}D\text{"} \mid T) = p_2$. This means that the categorization of an item is assumed to depend only on the processing outcome of this item, and independent of the processing outcomes of all other items in the display. p_1 and p_2 are determined by

$$p_1 = a_1 k^{-b} \text{ and } p_2 = a_2 k^b, \text{ where } a_1, a_2, b > 0$$

Assumption 3. The queueing process terminates as soon as either of the events occurs:

3.1. A customer is judged as target (self-terminating), which initiates a positive response (pressing "yes");

3.2. $l(k)$ customers have been served and judged as distractor, which initiates a negative response (pressing "no"), whereby $l(k)$ is a function of k given by

$$l(k) = \begin{cases} k\big(1 - \exp\big(\beta k + \alpha\big)\big), & \text{if } 1 \leq k \leq -\dfrac{\alpha}{\beta} \\ 0 & \text{if } k > -\dfrac{\alpha}{\beta} \end{cases} \text{ whereby } \beta > 0$$

These three assumptions combined imply the error probabilities

$$\mathbb{P}(\text{false alarm}) = 1 - \big(1 - a_1 k^{-b}\big)^{k\left(1-\exp\left(\alpha+\beta k\right)\right)} \tag{4.15}$$

and

$$\begin{aligned} \mathbb{P}(\text{miss}) = &\big(1 - a_1 k^{-b}\big)^{k\left(1-\exp\left(\alpha+\beta k\right)\right)-1} \\ &\cdot \Big(\big(1 - a_1 k^{-b}\big)\exp\big(\alpha + \beta k\big) + a_2 k^b \big(1 - \exp\big(\alpha + \beta k\big)\big)\Big). \end{aligned} \tag{4.16}$$

Since the observable variables (false alarm and miss rates) are expressed as analytical formulas of the parameters by the model, the parameters can be estimated using MLE (see Section 6.1).

Some technical issues still need clarification when incorporating this error model into the Markovian queueing model of visual search. First, my model allows the termination rule l to be non-integers and the prediction will mostly be a non-integer. A non-integer l will not affect the modeling of error rates, but the queueing model can stop and respond only when the service for an integer number of customers is finished. To solve this problem, I adopt a probabilistic interpretation of l. That is, I model the stopping criterion as a random variable $L = \lfloor l \rfloor + \Delta L$, whereby $\Delta L \sim \text{Bernoulli}\big(l - \lfloor l \rfloor\big)$[6]. Then the random stopping criterion L defined in this way has the mean l. To estimate the parameters of the error model using MLE, the formulas (4.15) and (4.16) are used (see Section 6.1.

[6]An example: If $l = 17.23$, then the actual stopping criterion will be 18 with the probability of 0.23 and 17 with the probability of 0.77.

Another issue regards the number of set size levels implemented in an experiment. To obtain a reasonable parameter estimation, the experimenter should include at least four levels of set size the experiment. Otherwise, there will not be enough data for the parameter estimation. For each set size, two error rates can be observed. The error model contains five parameters. This means that the parameter estimation is impossible if less than three set sizes are realized.

On the other hand, reducing the number of trials of each set size level to offer capacity for more set size levels has a disadvantage. Since p_1 is very small, it requires a large number of trials of each set size level to obtain an estimate. At least 300 target-absent trials per set size are required to guarantee a small probability of observing no false alarms.

Between the two sources of errors, the termination rule is important for the RT modeling, especially the target-absent RTs. My model does not assume that genuine processing errors affect the RT at the first place. Nevertheless, it is plausible that speed-accuracy trade-off takes place on the level of identification of single items, reflecting in shortened service time for each item. This can be a future direction to improve the model.

4.6 Taking RTs of incorrect responses into account

Strictly speaking, the error model proposed here does not rely on the specifics of a queueing system as the underlying structure of processing. The mechanism combining the termination rule and the threshold adaptation makes the same prediction on error rates, regardless of a serial, parallel, or hybrid processing. Nevertheless, integrating this mechanism in the framework of the queueing model specified in Chapter 3 has important impacts on the RT predictions.

First, it revises the assumption of exhaustive search, which the prediction of correct RTs by the queueing model was originally based on. Incorporating the error model means replacing the exhaustive search assumption with the termination rule specified by the error model. Since the termination rule depends on set size according to the error model, doing this has a non-trivial influence on the prediction of correct RTs. The implementation of the termination rule in the simulation of the queueing model is presented in Section 5.1.2.

Second, it enables the prediction of error RTs. Assumption 3 maps the response "yes" and "no" uniquely to the mental state of confirming seeing the target or disapproving finding it, respectively. A judgment may conflict the information given by the environment sometimes. In this case, it is a misjudgment, but it matches the mental state[7]. Regardless of whether an error arises from imperfect processing or incomplete search, this mapping between the response option and the mental state remains the same. It follows that the RT of a false alarm should be a realization of the same random variable as the RT of a hit. Similarly, the RT of a miss detection should be a realization of the same random variable as the RT of a correct rejection.

The second aspect has further implications for the model evaluation. Given that variables of interest have been derived from a model, a parameter estimation method usually searches for parameter values that minimize the discrepancy between model and data, or maximize the similarity between model and data. As long as the model does not make predictions regarding data defined as errors, such an estimation method applies only to the ideal situation where the empirical data is "clean", i.e., representing the random variable of interest.

In other disciplines, errors occur in empirically collected data as well. They may be caused by noise, imprecision of the measurement tool or other interfering factors. As long as data points can be identified as errors, the common method of dealing with them is to exclude them from the analysis or modeling. This approach is justified when little or nothing is known about the origin of the errors. It is in particular faultless, when there are sufficient data collected and the error rate is sufficiently low.

As discussed throughout this chapter, error trials in visual search are systematic and not a result of noise. They carry relevant information that is not redundant to correct trials. More importantly, an explicit model for errors has been specified here, which not only makes predictions on error rates but also on the RTs of errors. Given that it is able to explain empirical error rates well (see Chapter 7), simply excluding error RTs may lead to unnecessary loss of valuable information.

[7]The error model does not take motor errors into account, as stated at the beginning of this chapter.

A concern about making use of incorrect responses is that their infrequency constrains the amount of information they carry. If the entire data set is not large enough, e.g., data collected from one single subject containing less than 100 trials for each condition, it is probable that no single false alarm will be observed in one of the target-absent conditions. The issue of sample size is discussed in Section 8.2.1.

Assuming that there are sufficiently large amount of data collected such that the information in incorrect responses can be revealed, the approach to taking incorrect responses into account is natural. According to Assumption 3, the empirical RTs recorded under a specified experimental condition (set size $\times$ target presence) are a mixture of realizations of two different random variables. Correct and incorrect RTs (indicated by "yes" or "no" responses) should be separated. Since the model predicts both RTs of correct and incorrect responses, a parameter estimation method should seek for the parameter values that optimizes a criterion that takes both random variables into account. In the current case, the distance between predicted and observed RTs is calculated for correct and and incorrect responses separately. The parameter estimation method minimizes the overall distance (see Section 6.2.2 for details).

Chapter 5

Simulation and preliminary model assessment

5.1 Simulation of responses and RTs

To simulate realizations of the random variables of interest, responses mimicking the responses of a human observer under each specified condition (set size $\times$ target presence) and the associated RTs should be generated. That is, the system's judgment on the presence of the target customer and the time elapsed from stimulus onset until the judgment should be documented. Accordingly, the simulation outputs should include single system responses with the associated system response time given the experimental conditions and specified model parameters as inputs. The generated RT and response should underlie the stochastic process characterized by the queueing model specified in Section 3.4. The empirical settings in which the data should be collected are described in Section 3.3.1.

5.1.1 Goals and basic settings of the simulation

Simulation enables a structured and thorough investigation of the model behavior depending on the changes in parameters as well as an assessment of the robustness of model assumptions. The investigation and assessment should help

us understand the model better in terms of "what does the model do?" It is useful for assessment of quality of estimators.

More importantly, simulations build the foundation of parameter estimation for my model. Since a mathematically closed form of a parameter estimator is not available, not even an explicit analytical form of the CDF or PDF of the random variables of interest is known, the theoretical distribution of RTs has to be approximated by simulating a large number of single responses with RTs, called Monte Carlo integration. By the Glivenko-Cantelli theorem, the empirical CDF of the simulated RTs converges uniformly to the theoretical CDF of the random variables of interest. Thus, the comparison of experimental data to the theoretical CDF predicted by the model can be replaced by the comparison of experimental data to simulated data (details see Section 6.2.1).

As described in Section 3.4, the queueing model makes predictions depending on two experimental factors: target presence (target-absent vs. target-present) and set size k. These factors are represented in the model by the presence or absence of a specified target customer and the total number of customers k, respectively. The model parameters are considered intrinsic[1]: the speed of perception characterized by the interarrival rate λ, the speed of recognition characterized by the service rate μ, the residual times T_{resn} and T_{resy}, as well as the accuracy-related parameters $(\alpha, \beta, a_1, a_2, b)$. The experimental factors and the parameters constitute the inputs of the simulation[2].

5.1.2 The algorithm

The algorithm used here is developed based on the principles and algorithms in Stewart (2009) (pp. 682-685), where algorithms are designed to stimulate a single-server queueing system ($M/M/1$) with a fixed termination criterion (time threshold or number of served customers). To develop an algorithm directly

[1]This does not imply that they cannot be influenced by external conditions. Their values can change under different experimental conditions, e.g., perceptual difficulty and payoff for correct identification.

[2]Here, the assumptions on the laws of interarrival times and the service times as well as the relation between the accuracy-related parameters are considered implicitly given as specified in Section 3.3.4 and 4.5. Therefore, the parameters are sufficient to constitute the input data. In more general contexts, the laws of the random variables involved in the queueing process are also part of the inputs

applicable to the queueing model specified in the current context, two major issues have to be solved:

1) Integration of the system response mechanism described in Section 4.5 into a multiple-server system;

2) Adding (independent and parallel) servers such that it is in accordance to the queueing discipline "first come first served."

In this section, a detailed description of the algorithm is given, addressing these two issues.

5.1.2.1 Simulation of a system response

To solve the first issue, I added a procedure generating a system response according to the mechanism described in Section 4.5. The model assumes that the classification of a single item as target or non-target only depends on its actual status of being the target or a distractor and whether its processing is subject to a genuine processing error. In other words, the processing outcome of a single item (the "target/non-target" -judgment regarding this one item) is assumed to be independent of its processing time. It follows that the system response after the termination of the queueing process depends only on the outcomes of the classification, the position of the target entering the system, and the termination rule. In this model, the input data regarding these three factors, i.e., accuracy-relevant parameters ($\alpha, \beta, a_1, a_2, b$) and the condition parameters (set size k and target presence), are sufficient to predict the outcome, since the termination rule is based only on counting, not timing. The operation time of the queueing system, however, relies on the outcome of the system response and the concrete termination criterion in a run. As a consequence, the system response of one single trial (one queueing process) can be simulated ahead of the simulation of the temporal dynamics. For the convenience of programming, the part of the algorithm responsible for the simulation of a system response is designed as a procedure running ahead of simulating the queueing process and independently of the operation time of the queueing system.

The working of this procedure is straightforward since every system response is a logical consequence of the conditions and realizations of the relevant random

variables, which should be apparent from the discussion in Section 4.5. In case of target absence, the system responds "no" as soon as the number of departures reaches the stopping criterion L if no genuine processing error occurs to the items positioned prior to L, and "yes" otherwise, stopping at the position of the first misidentified item. The case of target presence is a bit more complicated due to the additional possibility of misidentifying the target. If no genuine error occurs to any of the items positioned prior to L, the response and termination depends on whether the target is included in the set of searched items. If this is the case, the system responds "yes", stopping at the position of the target, and "no" otherwise, stopping at the position L. If the target is misidentified but all distractors in the set of searched items are identified correctly, the system responses the same way as in the target-absent case with no genuine processing error. If there is at least one misidentification of the distractors in the set of searched items, the system responses "yes" and stops at position of the first misidentified distractor or the target, whichever happens earlier.

To solve the second problem, I introduced several variables enabling the tracking of the activity state of the servers and the order of arrival and departure of each customer. To make the modifications comprehensible, I will first outline the method I adopted to simulate the operation time of the queueing system. The procedure simulating the queueing time takes the result of response generating procedure, i.e., the system response and the termination position, as part of the inputs. Note that the time resulted from simulating the queueing process does not correspond to the RT I am trying to model, but rather the pure operation time of the queueing system, i.e., the time required to process the visual stimuli until a decision is made on the basis of the information collected during the processing. In other words, the RT is assumed to be the sum of the system operation time and an additive constant T_{resn} or T_{resy} representing the residual time.

5.1.2.2 Discrete event systems

The simulation is based on the method of discrete event systems, which is commonly used for simulating general queueing systems (e.g., Stewart, 2009). This method is used to model and simulate systems with a discrete set of states. The dynamics of the systems (i.e., transitions between states) is driven by events

which occur at asynchronous discrete moments of time. The concept of states of discrete event systems is more general than the term state applied to queueing systems (i.e., number of customers in the system) as defined in Section 3.4.1. Rather, it refers to a set of variables that characterize the system at any specific time. Here, the variables constituting the states are the counter values for the relevant events (see the next paragraph for details). The discrete events triggering a state transition are the arrivals and departures of customers. The arrivals and departures occur at random times and instantaneously (i.e., the events themselves are modeled as having duration zero). The development of the states of the system over time determines the outputs of the system on the one hand, depends itself entirely on sequence of the occurrences of the events on the other hand. The logic of discrete event systems is to simulates the operation of a queue by tracking the sequence of the arrivals and departures as the state development, updating the state and the relevant variables upon the occurrence of every event (i.e., "what to do after a customer has arrived/departed?") until a pre-defined stopping condition is fulfilled.

5.1.2.3 Key components and construction of the simulation of system operation time

The general principles of discrete event systems are described in detail in Stewart (2009). The description below is modified and adapted to the current context. The simulation of the system operation time for the queueing model includes the following key components (Stewart, 2009).

State descriptor. The system state at time t is a set of variables representing the information at time t such that the evolution of the system from t to the occurrence of the next event is uniquely determined from this information and the inputs. For the queueing model, this set of variables includes above all the "internal clock" of the simulation (t itself) indicating the passage of time since the start of the arriving process (i.e., stimulus onset), and the number of customers in the system (denoted by n_S), which plays a key role in controlling the development of the system. Furthermore, the counter values for the relevant events, i.e., the cumulative number of arrivals (n_A) and departures (n_D), also have an influence

on the occurrence of the next consecutive event (see the discussion in Section 5.1.3) and are thus part of the state.

Event list. The event list consists of concrete events that are going to take place in advance time. These events are "pending" from the view of the queueing system because they resulted from a previously simulated event but have not yet occurred themselves according to the "internal clock". For instance, after an arrival, if there are still customers that have not yet been in the system, the interarrival time between the current and the next arrival should be generated and thereby the time at which the next arrival will happen needs to be "scheduled". The event list thus collects all the events "scheduled" for the "future" due to the occurrence of events that have happened previously. The elements in the event list are characterized by the time at which they will happen and the type of the event. Since it is essential to schedule these events which succeed each other correctly, they are sorted by the event time.

Generating random numbers. The dynamic of the queueing system is uniquely characterized by the transitions. The transitions are in turn characterized by the infinitesimal transition rates, which are determined by the parameter λ, μ and the current state (see Section 3.4). Thus, the random times at which arrivals and departures occur must be simulated in accordance with the infinitesimal transition rates. Here, exponentially distributed random numbers with the proper rate need to be generated whenever an event triggers the subroutine that requires the determination of interarrival or service time.

Bookkeeping list. The bookkeeping list records the intermediate results and statistics of interest, computes performance measures and provides information to the executive for the termination of the queue if necessary.

Executive. To incorporate the components described above into a functional entirety, there must be an executive acting like a director, controlling the "internal clock" and the chronological arrangement, as well as executing events and the corresponding subroutine. At the beginning of the simulation run, the executive

initializes all relevant variables, in particular the time of the first arrival, and advances the "clock" to this time. It then monitors the events in the event list, updates the state and the event list and jumps to the time at which the next event is "scheduled" (since the events are modeled as instantaneous) after the subroutine is finished. Critical for the controlling are the subroutines after an arrival or a departure. The subroutine given that the next upcoming event is an arrival includes following actions:

- Update the state. Advance the "clock" to the time of the upcoming arrival and increment both the number of arrived customers and the number of customers in the system by one.
- Log the arrival time in the bookkeeping list.
- If there are customers that have not yet been in the system, generate an interarrival time according to λ and the current state and add the next arrival to the event list; otherwise, no arrival will happen before the next transition.
- If there are idle servers, assign the arriving customer to an idle server, generate the service time for this customer and add the departure to the event list.

The subroutine given that the next upcoming event is a departure includes following actions:

- Update the state. Advance the "clock" to the time of the upcoming departure and increment the number of departed customers and decrement the number of customers in the system by one.
- Find out the arrival order of the departing customer as well as the index of the server at which the departure is happening. Log the departure time in the bookkeeping list and mark the server as free.
- If there are customers in the queue waiting for service, assign the first customer in the queue to the idle server with the smallest index, generate the service time for this customer and log the departure to the event list; otherwise, no departure will happen before the next transition.

The executive will terminate the operation of the updating process of the state and the event list once the stopping criterion is met. The stopping criterion in case of a "no" response is the value of L and in case of a "yes" response the departure of the item judged as "target customer", as described in Section 4.5. By then, the results and statistics of interest can be returned. In this way, the times of arrival and departure can be simulated and recorded for every customer.

Now let us return to the second problem described at the beginning of this section (5.1.2). The solution is already disclosed in the subroutines described above. From the perspective of simulation, the difficulty of extending the simulation from single-server to multiple servers is the management of the assignment of customers to servers. It arises from the fact that different from a single-server system, in which customers come and leave in exactly the same order, the departure in a multiple server system does not need to happen in the same order as arrivals. Incoming customers as well as the servers must be labeled uniquely such that each assignment of a customer to a server can be well documented. Otherwise, the arrival time and the service time of the same customer cannot be mapped correctly, causing problems in calculating and recording the departure time of the customer.

The approach I adopted is to index the customers in the order of arrival $1, \ldots, k$ and the servers simply by $1, \ldots, c$. In order to keep track of service history of each customer and server, the event list is extended by an additional slot for each server. Each slot represents the state of the corresponding server: The value zero indicates that the server is currently idle; a value i in $1, \ldots, k$ indicates that the service for the i-th arrived customer is running at this server. Initially, all slots are filled with zero. Whenever a customer arrives, the executive checks whether any of the slots has the value zero. If this is the case, the customer will be immediately assigned to the free server with the smallest index and the incoming order will be registered in the corresponding slot. As soon as the service is finished, the value will be replaced by zero and the executive checks whether there is a customer waiting in line and executes an assignment if this is the case. In this way, the service by multiple parallel servers according to the queueing discipline "first come first served" is realized.

As the system operation time is assumed to differ from the RT by an additive

constant of the residual time T_{resn} or T_{resy}, we obtain the system response time predicted by the model by adding T_{resn} or T_{resy} to the simulated system operation time. The simulation of the complete queueing model with response as specified in Section 5.1 is implemented as a function. The algorithm designed for the entire simulation is presented as pseudocode in Algorithm 1. To keep the structure clear, procedures executing different functionality are presented separately.

Algorithm 1 (part 1): Simulation of a system response

Input: $(\alpha, \beta, a_1, a_2, b, k, \text{pr})$
Output: (response, termination)

set $l = k(1 - \exp(\alpha + \beta k))$;
generate $\Delta L \sim \text{Bernoulli}\left(l - \lfloor l \rfloor\right)$;
set $L = \lfloor l \rfloor + \Delta L$;
set $p_1 = a_1 k^{-b}$, $p_2 = a_2 k^b$;
generate $F_{1,\ldots,k} \overset{\text{i.i.d.}}{\sim} \text{Bernoulli}(p_1)$, $G \sim \text{Bernoulli}\left(p_2\right)$;
if pr = FALSE **then**
    if $F_i = 0 \; \forall i = 1, \ldots, L$ **then**
        set termination = L ;
        set response = FALSE ;
    else
        set termination = min$\{ j \mid F_j = 1 \}$;
        set response = TRUE ;
    end if
else
    generate $P_{target} \sim \text{unif}\left\{1, \ldots, k\right\}$;
    set $F_{P_{target}} = 0$;
    if $F_i = 0 \; \forall i = 1, \ldots, L$ **then**
        if $G = 0$ **then**
            if $P_{target} \leq L$ **then**
                set termination = P_{target} ;
                set response = TRUE ;
            else
                set termination = L ;
                set response = FALSE ;
            end if
        else
            set termination = L ;
            set response = FALSE ;
        end if
    else
        if $G = 0$ **then**
            set termination
              = min$\{ P_{target}$, min$\{ j \mid F_j = 1 \}\}$;
            set response = TRUE ;
        else
            set termination = min$\{ j \mid F_j = 1 \}$;
            set response = TRUE ;
        end if
    end if
end if

Algorithm 1 (part 2) Simulation of a system response time (main body)

Data structures:
input = $(\lambda, \mu, T_{resn}, T_{resy}, c, k,$ pr, response, termination)
state = (t, n_A, n_D, n_S)
out = $\begin{pmatrix} A_1 & \dots & A_{n_D} & A_{n_D+1} & \dots & A_{n_A} \\ D_1 & \dots & A_{n_D} & +\infty & \dots & +\infty \end{pmatrix}$
event_list = $(t_A, t_{D_{S1}}, t_{D_{S2}}, \dots, t_{D_{Sc}})$
service_id = $(x_1, \dots, x_c)$
Random variables:
$I_{1,\dots,k} \sim \text{Exp}\left((k - n_A)\lambda\right)$
$S \sim \text{Exp}\left(\mu\right)$
Output: (system_rt, response)

initialization
  set state = $(0, 0, 0, 0)$;
  set out = $\begin{pmatrix} \cdot \\ \cdot \end{pmatrix}$;
  set service_id = $(0, \dots, 0)$, ;
  set dep_id = 0, dep_server= 0 ;
  generate $I_1 \sim \text{Exp}\left(k\lambda\right)$;
  set $t_A = I_1$;
  set event_list = $(t_A, +\infty, \dots, +\infty)$;
  set STOP = FALSE ;
simulation loop
  while *not STOP* **do**
    if $t_A = \min\{ t_A, t_{D_{S1}}, t_{D_{S2}}, \dots, t_{D_{Sc}} \}$ **then**
      execute subroutine UPDATE_ARRIVAL ;
    else
      execute subroutine UPDATE_DEPARTURE ;
    end if
    if $t_A = t_{D_{S1}} = \dots = t_{D_{Sc}} = +\infty$
    or (response = FALSE *and* $n_D \geq$ *termination)*
    or (response = TRUE *and dep_id = termination)* **then**
      STOP = TRUE ;
    end if
  end while
set system_rt= $t + T_{resn} +$ response $\cdot (T_{resy} - T_{resn})$;
set output = (system_rt, response) ;

Procedure: UPDATE_ARRIVAL

set $t = t_A$, update n_A by $n_A + 1$, update n_S by $n_S + 1$;
set $A_{n_A} = t$, out $= \left(\text{out}, \begin{pmatrix} A_{n_A} \\ +\infty \end{pmatrix}\right)$;
if $n_A < k$ **then**
  generate $I_{n_A+1} \sim \text{Exp}\left((k - n_A)\lambda\right)$;
  set $t_A = t + I_{n_A+1}$;
else
  set $t_A = +\infty$;
end if
set free_servers $= \{\, j \mid x_j = 0 \,\}$;
if $n_S \leq c$ *and* free_servers $\neq \emptyset$ **then**
  generate $S \sim \text{Exp}\left(\mu\right)$;
  set $i =$ min(free_servers) ;
  set $t_{D_{Si}} = t + S$, $x_i = n_A$;
end if

Procedure: UPDATE_DEPARTURE

set $t = \min\{\, t_A, t_{D_{S1}}, t_{D_{S2}}, \ldots, t_{D_{Sc}} \,\}$, n_D++, n_S-- ;
set $D_{n_D} = t$;
set dep_server $= j$ such that
  $t_{D_{Sj}} = \min\{\, t_A, t_{D_{S1}}, t_{D_{S2}}, \ldots, t_{D_{Sc}} \,\}$;
set dep_id $= x_{\text{dep\_server}}$, $\text{out}_{2,\text{dep\_id}} = D_{n_D}$;
set $x_{\text{dep\_server}} = 0$, free_servers$= \{\, j \mid x_j = 0 \,\}$;
if $n_S \geq c$ **then**
  set $i =$ min(free_servers) ;
  set $x_i = n_D + (c - |\text{free\_servers}|) + 1$;
  generate $S \sim \text{Exp}\left(\mu\right)$;
  set $t_{D_{Si}} = t + S$;
else
  set $t_{D_{Sj}} = +\infty \quad \forall j \in$ free_servers;
end if

5.1.3 Implementation and speed-up

Algorithm 1 is implemented as a function receiving the arguments ($\frac{1}{\lambda}$, $\frac{1}{\mu}$, T_{resn}, T_{resy} α, β, a_1, a_2, b, c, k, N, pr, seed) and returns a $N \times 2$ matrix with the simulated system response time and the response as columns. "Pr" represents the experimental factor target presence. The value FALSE indicates a target-absent trial and TRUE indicates a target-present trial. N denotes the number of simulation runs and "seed" the random seed of the random number generator.

Although the relevant transition rates λ and μ are standard for presenting the abstract formulation of queueing models, $\frac{1}{\lambda}$, $\frac{1}{\mu}$ are usually more convenient to use in simulations because they are numerically easier to handle and lead to results that are easier to communicate. This is especially the case here because the time unit is millisecond and the findings regarding the RT in visual search and the attentional dwell time suggest that $\frac{1}{\lambda}$ and $\frac{1}{\mu}$ must be on the scale of a few dozen to a few hundred milliseconds. This means that the estimates of λ and μ are expected be very tiny numbers. The parameters of $\frac{1}{\mu}$ and $\frac{1}{\lambda}$ can be understood as mean service time and a kind of mean interarrival time, respectively. The latter refers to the mean interarrival time when the factor of the number of customers who have not been in the system is discounted.

The implementation pursues two goals: reducing computing time and numerically undesirable characteristics. For simulations of complex models, such as the queueing model, computing time is doubtless an important factor. Its significance becomes even more noticeable as the parameter estimation relies on simulating a large amount of realizations. To find the best-fitting parameters, the distance between the empirical data and the simulated realizations needs to be minimized (see Section 6.2 for details). At each iteration of the optimization process, evaluating the objective function means calculating the distance between the empirical and simulated RTs given a set of parameter values. In fact, we need about 10^5 realizations to obtain a good approximation of the theoretical CDF for a single set of parameter values. Now, for finding the best fit, the parameter space has to be searched. Even under the most ideal conditions where the optimization algorithm works efficiently and the functional surface in the parameter space has no undesirable characteristics, it could take 20 to 30 iteration steps to converge. If such propitious conditions are not given, the optimization may need a hundreds

of iteration steps. Additionally, for stability reasons, one of the parameters (c) has to be profiled (see Section 6.2.3 for details). Thus, specific attention to efficient implementation is necessary.

Stewart (2009) presents codes that implement his algorithms in a straightforward way. The implementation is exemplary from a didactic point of view, optimally designed for facilitating students' understanding. However, this merit comes along with higher computational costs. Implementing Algorithm 1 using R in a similar manner results in codes that require up to about 40 minutes[3] to complete 10^5 simulation runs. This means that it would take a few weeks to execute a single parameter estimation process.

To reduce the computing time, factors that prolong the computation were identified and some strategies were applied to mitigate the influences of these factors. Both the general and fine structures were modified. For example, function calls were avoided whenever it was possible because they have a high overhead cost, especially in R. Repeated calculations were also avoided by storing intermediate results in additional variables. After these modifications, the R code (see Appendix C) may appear less comprehensible for the reader to follow, yet the gain on speed outweighs the need of presenting clean and simple code.

The most important modification is centralizing the generation of random numbers so that it is separated from the execution of all the other actions. That is, all random numbers required for simulating N runs of the queueing process are generated at once and stored in a large pool prior to the start of the queueing process. A random number is selected from the pool whenever required. From a simulation point of view, this implementation generates realizations of the same random variable as the "generation-on-demand" approach described in Algorithm 1. Every call of the random number generator comes with an overhead cost, whereas generating a larger number of random numbers in a call contributes to the computational cost of this call only marginally. Therefore, applying the strategy of centralizing random number generation reduces the overall computing time of a simulation run.

[3]The simulation time depends on the values of k and pr. The larger k, the longer the simulation run. Simulating the case of target absence takes longer than target presence. Thus the longest simulation time is usually observed for $k = 18$ and pr = FALSE.

Most importantly, centralizing the generation of the random numbers avoids numerically undesirable characteristics of the parameter estimation based on the simulation. This aspect is related to a weakness of parameter estimation methods that rely on Monte Carlo simulations. Unlike the theoretical model characteristics, their approximation based on simulated data has certain fluctuation depending on the concrete outcome of the simulation. Even if the simulation generates a very large sample, e.g., 10^5, such that the discrepancy between the approximation and the theoretical model prediction is negligible, this fluctuation cannot be eliminated completely. This means that if we repeat the calculation, the approximation based on another sample of the simulated data will be slightly different and will lead to a different value of the estimated parameters when fitted to the same empirical data set. Consequently, the objective function (see Section 6.2.1) will no longer be a mapping — it does not map the same argument to a function value uniquely. This will lead to unwanted side effects during the optimization. The optimizer will be traveling through a constantly changing "landscape." To avoid these fluctuations, a common method is to fix the random seed on the same value during the entire optimization process. Due to the algorithm implemented in `R` (or any other software using a deterministic method) for generating (pseudo) random numbers, the simulation will lead to exactly the same outcome (replicable sample) when repeated. There will be no fluctuations but a small bias due to the usage of a specific random seed. This method works well in practice, if the number of simulation runs is large enough.

However, fixing the random seed causes another unwanted side effect when applied to simulation of the queueing model. This side effect originates from an interplay of the discrete event systems approach (more specifically, the way how the internal "clock" of the system proceeds) and the way the algorithm in `R` generates (pseudo) random numbers. If the random seed is fixed, the sequence of numbers generated under the assumption of exponential distribution with rate parameter λ has the following property: If the simulation is repeated for a new rate parameter $a\lambda$, the result will simply be the original sequence multiplied by a^{-1}. For example, the sequence with mean 20 ($\lambda = \frac{1}{20}$) will simply be twice the sequence with mean 10 ($\lambda = \frac{1}{10}$). As explained in Section 5.1.2, the development of the system (and thus the outcome) is driven by the order of the events and

the order is determined by the simulated times. Due to the property of the generated random numbers described above, small changes in the parameter value could result in large changes in the outcome of a simulation run if random numbers are generated on demand. Consider a simple case where $\frac{1}{\lambda} = 50$ and $\frac{1}{\mu} = 120$ for $k = 12$ and $c = 4$. For the sake of simplicity, let us focus on a part of the development and assume that the current clock time t is 100 and the event occurring is the departure of the first customer, followed by the arrival of the ninth customer with a lag of less than 10 ms[4]. This is a situation quite likely to happen given the parameter values. Now let us change the interarrival rate a bit to $\frac{1}{\lambda} = 45$. The order of the first eight events will hold but the eighth arrival will happen at $t = 90$ due to the property of the generated random numbers. As a consequence, the next event to happen will not be the first departure but the ninth arrival. The next random number will be then assigned to the interarrival time of the tenth customer (update after the ninth arrival) rather than the service time of the fifth customer (update after the first departure), shrinking by a factor of 0.375, whereas the random number following it will be assigned to the service time of the fifth customer, expanded by a factor of 2.67. This exchange can be the beginning of the domino effect that transforms the subsequent development of the system state completely. This is similar to the situation in which the random seed is changed at irregular positions of the queueing process, causing chaotic movements in the development of the system. In this case, the changes in the simulation outcomes that come along with the changes in the parameter values do not only reflect inherent properties of the model but also expanded noise of the randomness. The influence of randomness is unnecessarily expanded because of dislocations of random numbers and the subsequent domino effect occurring irregularly in the course of queueing. Of course, such an artifact has limited effects on the overall pattern of the results of N simulation runs if N is large enough, yet its unpleasant consequences on the numerical optimization should not be underestimated when it comes to parameter estimation later. Due to this artifact, slight changes in the parameter value may result in unpredictable small increases or decreases in the measure of distance between model predictions and

[4]The system state is then $(t, n_A, n_D, n_S) = (100, 8, 0, 8)$. A server is becoming free and the other three are occupied.

empirical observations, leading to a very rough functional surface with dense jumps and discontinuities.

Centralized generation of random numbers resolves the effect of this artifact by regulating the assignment of random numbers to events of different types, so that the assignment is not affected by the concrete development of the system state. It turns out to be able to mitigate the issue of rough functional surface[5] and reduces the number of iterations of the optimization process.

The concrete approach consists of the generation and management of a pool of random numbers before calling the function simulating the queueing process. The simulation requires six types of random numbers: interarrival times, service times, misidentification of the distractors, misidentification of the target stopping criterion and target position. Random numbers of the same type should be generated and sorted separately, such that the assignment of random numbers of different types does not depend on each other. To achieve this, a slot is created for each type. The slots are variables of proper data type such that the random numbers are uniquely ordered. For interarrival times and service times, two $N \times k$ matrices filled with exponentially distributed random numbers with the rate 1 are generated. The entry in the i-th column and the j-th row of the first matrix will be selected in the i-th run and multiplied by $\frac{1}{(k-n_A)\lambda}$ to obtain the interarrival time of the j-th customer. Entries of the second matrix are assigned in the same way for the service times, multiplied by $\frac{1}{\mu}$. Analogously, a third $N \times k$ matrix filled with Bernoulli random numbers with parameter p_1 is generated and assigned in the same way to indicate the misidentification of distractors. For each of the other three types, random numbers subject to the corresponding law are organized in a vector of the length N and the i-th entry is assigned to the i-th simulation run, since only one random number of each of these three types is required per run. The stores are designed with fixed size to ensure that there are enough random numbers to allocate. Some of the random numbers generated ahead may not be used. The unused random numbers will be discarded and not recycled. Even though the centralized generation will have to generate more random numbers in this way, it is much more efficient than a dispersed generation.

[5]Since there are other factors contributing to the characteristics of the functional surface, this measure does not eliminate the roughness. This issue will be explored in detail in Section 6.2.3

The simulation is implemented by the R code presented in Appendix C. Although the modifications have already improved the speed of the simulation to a large extent, it still takes a day or two to estimate all the parameters (including c the number of servers) for a data set under a specific condition (k and target presence must be specified). This is limited by the computational efficiency of R. A compiled language such as C or C++ has clearly speed advantages compared to R. An equivalent implementation of the R code for the simulation in C has been written by Martin Schlather. After all these efforts, the speed has been increased by about 100 times compared to the very first version.

5.2 Sensitivity analysis

Saltelli et al. (2004) defined a sensitivity analysis as "the study of how the uncertainty in the output of a model (numerical or otherwise) can be apportioned to different sources of uncertainty in the model input" (p.45). Sensitivity analysis has become an important tool in modeling practice in various scientific and technical disciplines. It is widely used for a range of purposes, including robustness testing, fault rectification, and model simplification and improvement. In this dissertation, sensitivity focuses on the understanding of the relationships between input (especially the parameters) and output variables in the model. Due to the complex structure of the model, an exploration of the multidimensional input space provides a better understanding of the model behavior to anticipate and resolve issues (e.g. identifiability issues) emerging in the next step — the parameter estimation.

One of the simplest and most common form of sensitivity analysis is the monothetic analysis (also known as the "one-factor-at-a-time method"), i.e., changing one of the input factors at a time and assessing the influence on the output of interest. A local sensitivity assessment is based on examining the local change in the output of interest as the input factor changes by a small amount in the neighborhood of certain points in the input space (hence "local"). Formally, if we denote the output of interest by Y, the input factors by $(X_1, \ldots, X_d)$ and express the model by $Y = f(X_1, \ldots, X_d)$, a sensitivity analysis by local methods involves essentially the calculation of the partial derivative of Y with respect to each

input factor X_i at some fixed point in the input space. A more global sensitivity assessment with respect to any input variable X_i is based on examining the change in Y as X_i varies over its entire domain for fixed values of the remaining input variables.

The drawback of a monothetic analysis is that it only investigates the effects of changes parallel to the X_i axes, one input variable at a time, but does not take the effects of interactions between input variables into account. In other words, its exploration of the input space is limited on the axis directions, so that it cannot detect perturbations in the output variables caused by simultaneous variations of two or more input variables.

For practical reasons, I adopted a monothetic approach of sensitivity analysis. Methods that perform a global sensitivity analysis are complex and computationally costly, thus regarded as beyond the scope of this dissertation. Because of the complexity of the model and the fact that the predictions of RT are based on Monte Carlo simulation, I opt for presenting the results via a visualization approach.

The sensitivity analysis for RT-related parameters focuses on the influence of the parameters $\frac{1}{\lambda}$, $\frac{1}{\mu}$ and c on the ECDF of system response time predicted by the model as the output variable of interest. Because the parameter T_{resn} or T_{resy} is an additive constant to the model prediction, its influence on the ECDF is simply a positive translation by the same units. Thus, the sensitivity of the parameter T_{resn} or T_{resy} does not need further investigation. T_{resn} and T_{resy} were fixed to 0 (i.e., no shift) during the sensitivity analysis of other RT-related parameters. The number of simulation runs was set to $N = 10^5$.

Although the model parameters are the focus of the sensitivity analysis, there are other important input variables, among others the target presence and set size k. These input factors should also be taken into consideration since the sensitivity of the parameters may be different for different values of these variables.

A global assessment of sensitivity instead of a local one, even using monothetic methods, is intractable in this case for two reasons. First, the model has a large number of input variables and is highly complex. Second, the calculation of model predictions for different values of input variables relies on computationally intensive Monte Carlo simulations. Although it is possible to vary a parameter

over its entire domain while keeping the remaining input variables fixed to certain baseline values, a high resolution level of the examination is computationally very costly and difficult to visualize. Therefore, the sensitivity analysis here was performed only for loose grid points in the input space to enable a first glance of the profiles of the sensitivity on a reasonable computational cost.

Specifically, for each of the parameters $\frac{1}{\lambda}$, $\frac{1}{\mu}$ and c, ten equidistant grid points across its parameter domain were selected. For each of the two remaining parameters, four equidistant grid points across the corresponding domains were selected. The RTs predicted by the model were simulated for the ten selected values of the parameter in focus, while the remaining two parameters were fixed to one of the four selected values. In this way, $(10 + 10 + 10) \cdot 4^2 = 480$ data sets were simulated for each condition (set size $\times$ target presence). The ECDFs were calculated from the simulated RTs and plotted in the same plot with different heatmap colors. Because the incorrect RT on a target-absent trial correspond to the model prediction of correct RT on a target-present trial and vice versa, the ECDFs were based on only correct RTs (see Section 4.6 for a detailed discussion on dealing with incorrect RTs). This procedure was repeated, permutating the parameter in focus and for given target presence and set size level.

The range of $\frac{1}{\lambda}$, $\frac{1}{\mu}$ and c were specified as $[10, 100]$, $[100, 800]$ and $1, \ldots, 10$, respectively. For $\frac{1}{\lambda}$, the range was divided in nine (or three, if the parameter is not in focus) intervals of equal length and the endpoints of the intervals were selected. Since the parameter domain of c consists of discrete points, the entire parameter domain was explored when c was the parameter in focus. For the profiling of other parameters, the points $2, 4, 6, 8$ were selected. Because $\frac{1}{\lambda}$ and $\frac{1}{\mu}$ are assumed to be on different scales and $\frac{1}{\mu}$ has a much larger parameter domain, choosing grid points for $\frac{1}{\mu}$ across the entire parameter domain may bias the comparison of the influences of $\frac{1}{\lambda}$ and $\frac{1}{\mu}$. To make the results of the sensitivity analysis comparable, $\frac{1}{\lambda}$ and $\frac{1}{\mu}$ should be varied by the same step length. Hence, the range in which the values of $\frac{1}{\mu}$ were sampled was limited to $[210, 300]$ and the values were selected in the same way as for $\frac{1}{\lambda}$. This range was considered most likely to contain meaningful values of $\frac{1}{\mu}$ because it is most consistent with the empirical findings regarding attentional dwell time.

Model predictions in both target-absent and target-present cases were taken

into account. The set size levels $3, 6, 12, 18$ were investigated in accordance with the data sets of Wolfe et al. (2010), which were the empirical data the model was planned to fit to. For the sake of simplicity and a neat illustration, only the results for $k = 3, 18$ were presented.

5.2.1 Sensitivity of $\frac{1}{\lambda}$

Figure 5.1 and Figure 5.2 illustrate the sensitivity profiles regarding $\frac{1}{\lambda}$ for $k = 3$. Figure 5.3 and Figure 5.4 illustrate the respective profiles for $k = 18$.

The first observation is that $\frac{1}{\lambda}$ appears to have a relatively small influence on the ECDF. It is manifested in the lower and the middle areas of the ECDF in target-absent case and the middle and upper areas in target-present case. The most apparent difference between the plots in the same column is a shift along the x-axis. This means that the pattern of this small effect does not seem to depend on the value of $\frac{1}{\mu}$ (at least the ones selected in the restricted range), if one takes the scale difference of x-axis of the plots into account. For small set size ($k = 3$), the effect (in fact, the entire ECDF) remains the same for $c \geq 4$, which is not surprising because every additional server from the fourth on will be always idle if there are only three customers to serve. For large set size ($k = 18$), the differences between the ECDFs under different values of $\frac{1}{\lambda}$ seem to be more apparent as c becomes larger. This indicates that if the system has more capacity so that customers can be assigned to a server upon arrival (or shortly after arrival), the mean interarrival time plays a larger role in determining the system response time. These observations indicate that the inverse inference on $\frac{1}{\lambda}$ from the observed RT is easier for small k and large c, because its contribution is more distinguishable under these conditions.

The largest concern these plots evoked was that the influence of T_{resn} or T_{resy} could easily confound with this influence attributed to $\frac{1}{\lambda}$. Although $\frac{1}{\lambda}$ transforms the ECDF in a different way (leading to a shift mainly in the middle area) than a simple translation, it has an even form so that shifting by a proper value would cause only minor deviations. In other words, the effect of $\frac{1}{\lambda}$ could be compensated by a shift to achieve a similar change in the distance measure. This is a sign for a correlation between $\frac{1}{\lambda}$ and T_{resn} or T_{resy}, which can cause serious identifiability

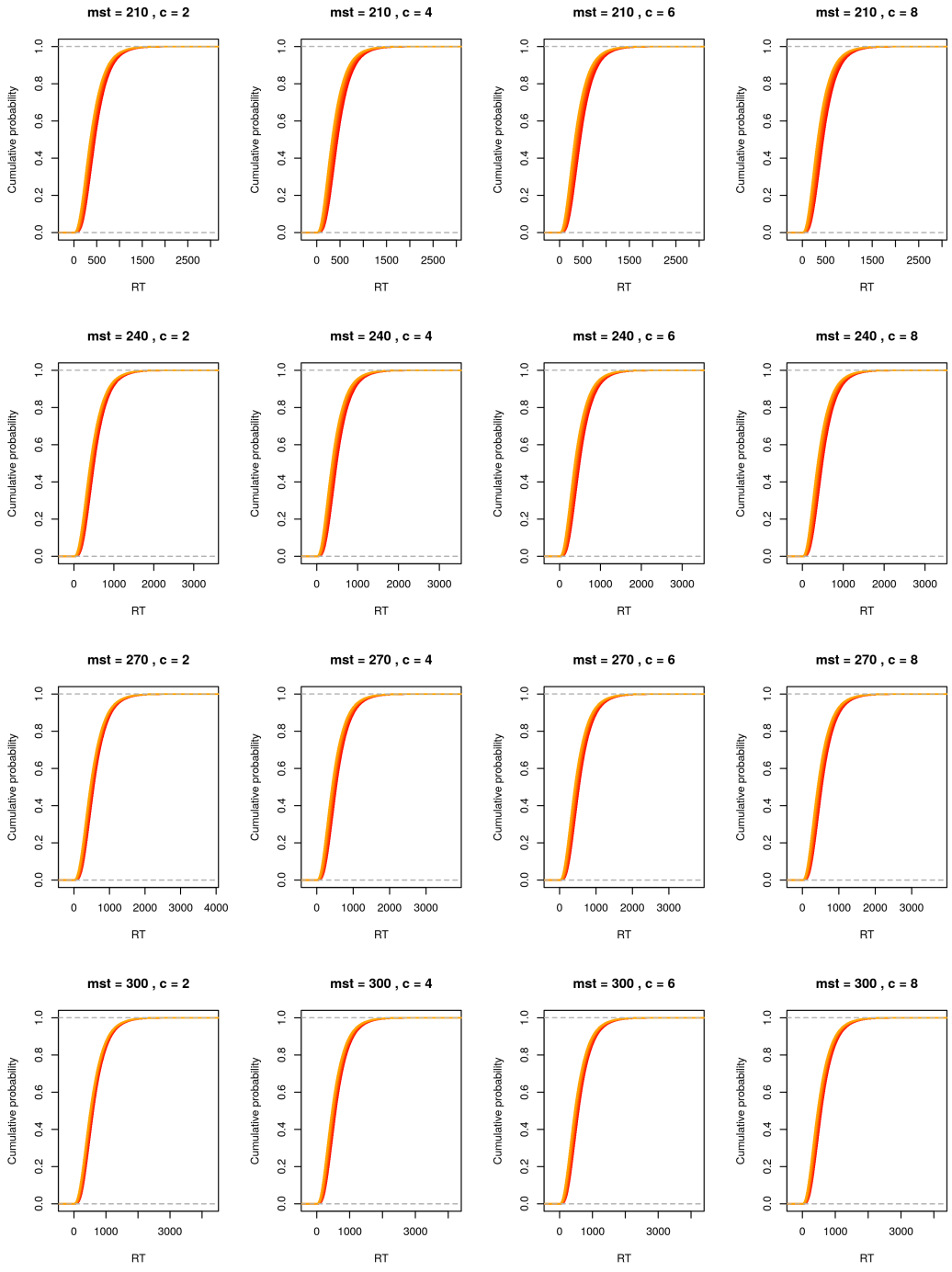

Figure 5.1: Profiles of the ECDFs of model prediction as a function of $\frac{1}{\lambda}$ for target-absent, $k = 3$, $N = 10^5$. "mst" stands for "mean service time".

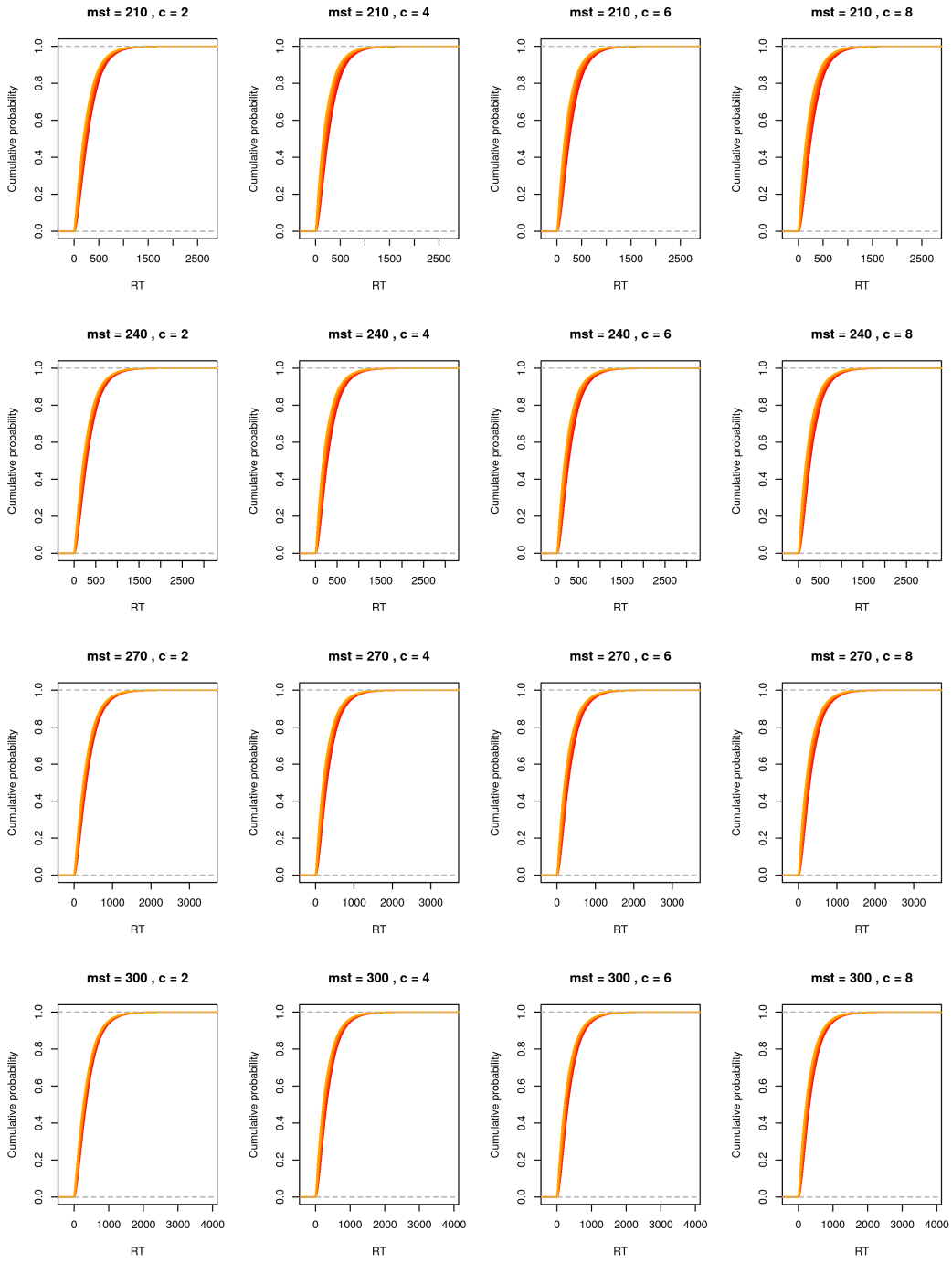

Figure 5.2: Profiles of the ECDFs of model prediction as a function of $\frac{1}{\lambda}$ for target-present, $k = 3$, $N = 10^5$. "mst" stands for "mean service time".

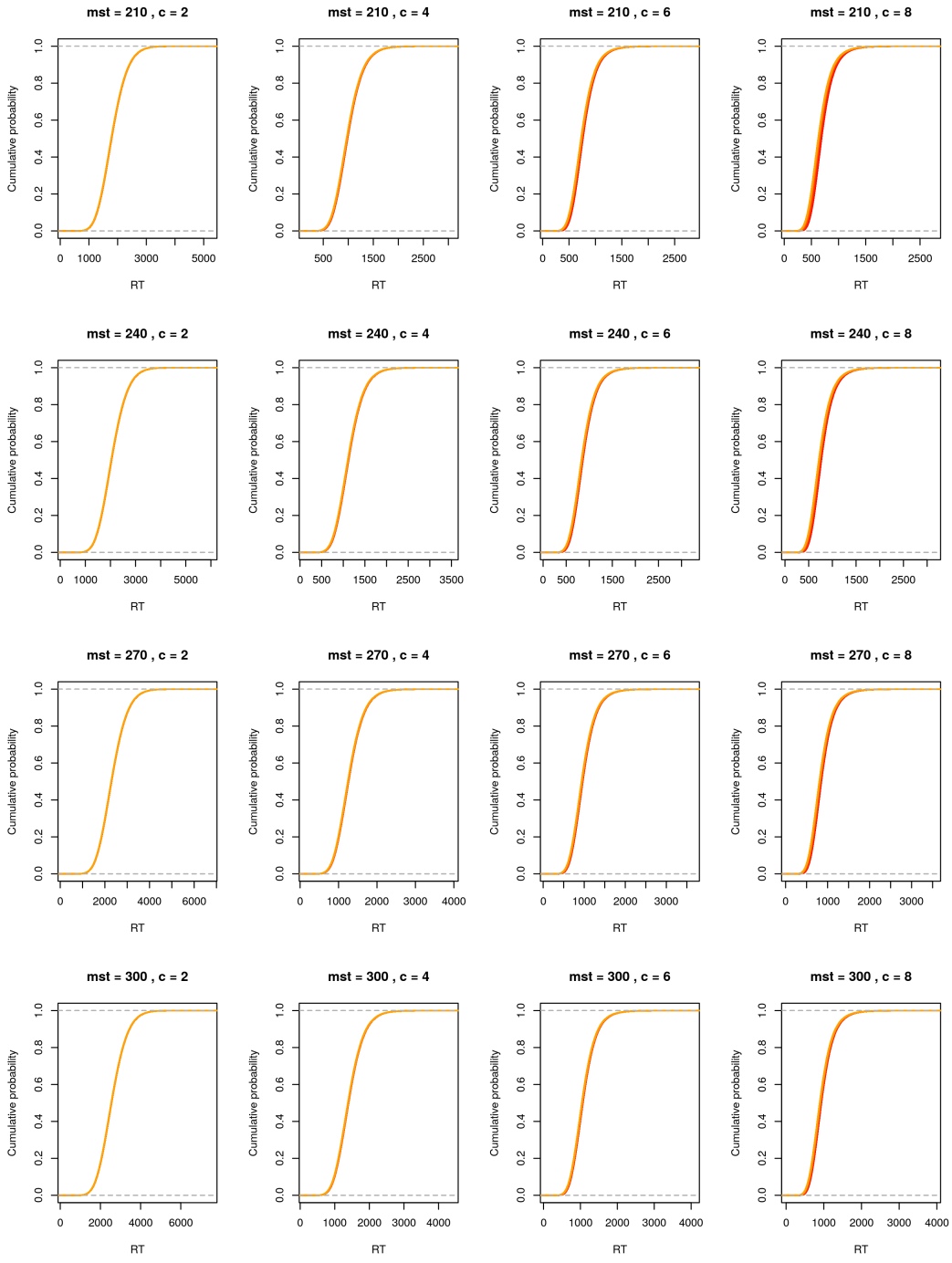

Figure 5.3: Profiles of the ECDFs of model prediction as a function of $\frac{1}{\lambda}$ for target-absent, $k = 18$, $N = 10^5$. "mst" stands for "mean service time".

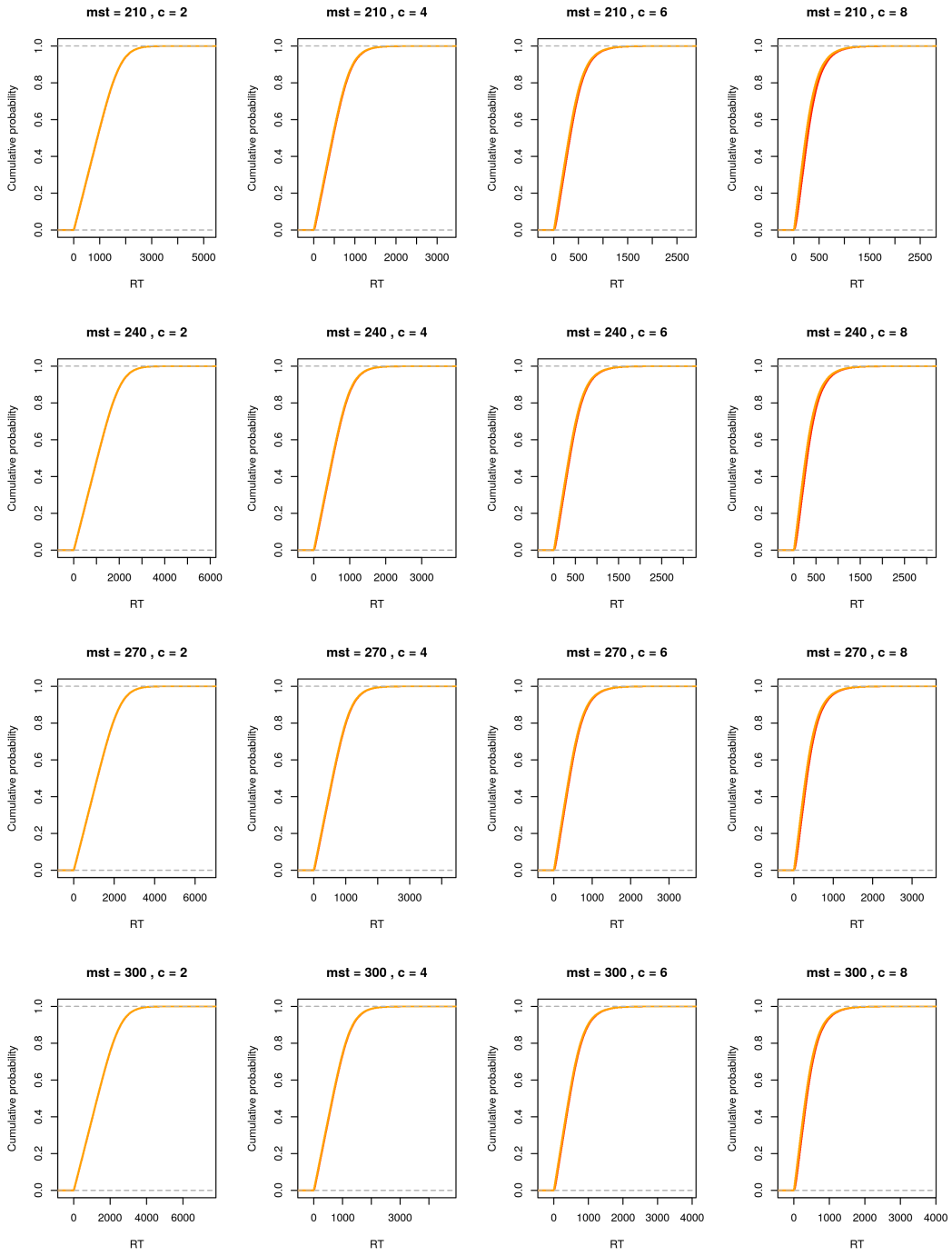

Figure 5.4: Profiles of the ECDFs of model prediction as a function of $\frac{1}{\lambda}$ for target-present, $k = 18$, $N = 10^5$. "mst" stands for "mean service time".

issues. Unfortunately, this concern was confirmed later in the development of parameter estimation method (see Section 6.2.3 for a detailed discussion).

5.2.2 Sensitivity of $\frac{1}{\mu}$

Figure 5.5 and Figure 5.6 illustrate the sensitivity profiles regarding $\frac{1}{\mu}$ for $k = 3$. Figure 5.7 and Figure 5.8 illustrate the respective profiles for $k = 18$.

In comparison with $\frac{1}{\lambda}$, $\frac{1}{\mu}$ appears to exert a much larger influence on the ECDF of RTs predicted by the model. Each unit by which $\frac{1}{\mu}$ varied produced a larger difference in the ECDF than the same unit of $\frac{1}{\lambda}$. By comparing the plots in the same column, it becomes apparent that the influence of $\frac{1}{\lambda}$ was very small relative to the influence of $\frac{1}{\mu}$. The form of the influence of $\frac{1}{\mu}$ was also different. It led to larger shifts in the upper part of the ECDF, causing a considerable change in the curvature. That is, $\frac{1}{\mu}$ changes the form, above all the tail of the RT distribution to a large extent. Again, for small set size ($k = 3$), the entire RT prediction and thus the impact of $\frac{1}{\mu}$ remain the same for $c \geq 4$. For large set size ($k = 18$), the impact of $\frac{1}{\mu}$ depends on c: It gets smaller as c increases. The impression gained by the visualization underestimates the extent of the dependence on c because the plots for smaller c had a larger unit on the x-axis. Apart from that, the absolute influence of $\frac{1}{\mu}$ was clearly larger for $k = 18$ than for $k = 3$. Within the same set size level, it was larger in the target-absent case than in the target-present case in general.

Based on these observations, $\frac{1}{\mu}$ is likely to have a good discriminability. Even though it depends on target presence and on c for large set size, its impact is more apparent than $\frac{1}{\lambda}$. More importantly, it is not easy to compensate such an effect by simple translations. For certain factor combinations, such as small set size and target-present, a shift may be able to produce a similar value in the distance measure. However, if data sets under different conditions are taken into account jointly, it seems promising to infer the value of $\frac{1}{\mu}$ inversely from the RTs. This speculation was also confirm later in the development of parameter estimation method (see Section 6.2.3)

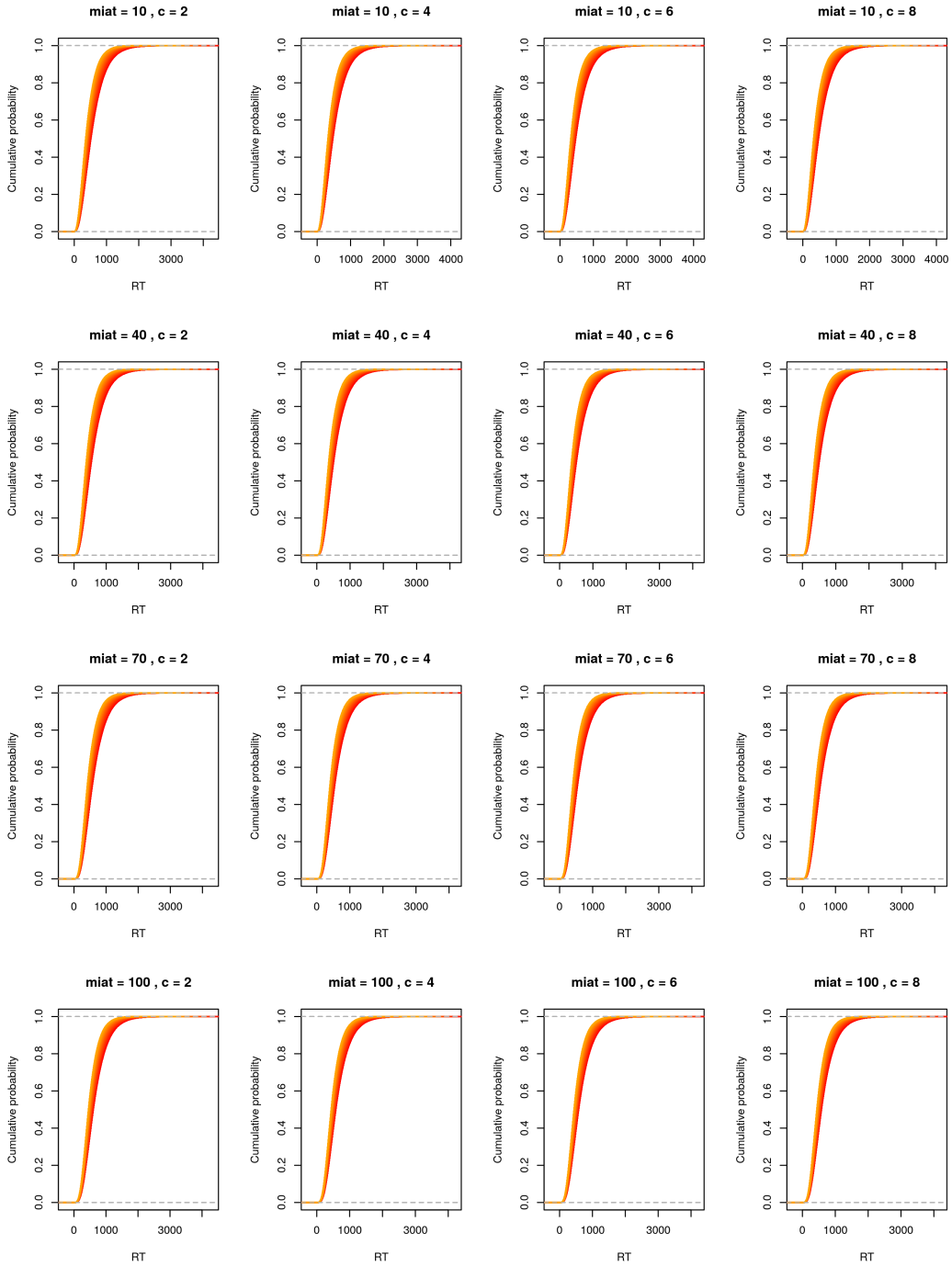

Figure 5.5: Profiles of the ECDFs of model prediction as a function of $\frac{1}{\mu}$ for target-absent, $k = 3$, $N = 10^5$. "miat" stands for "mean interarrival time".

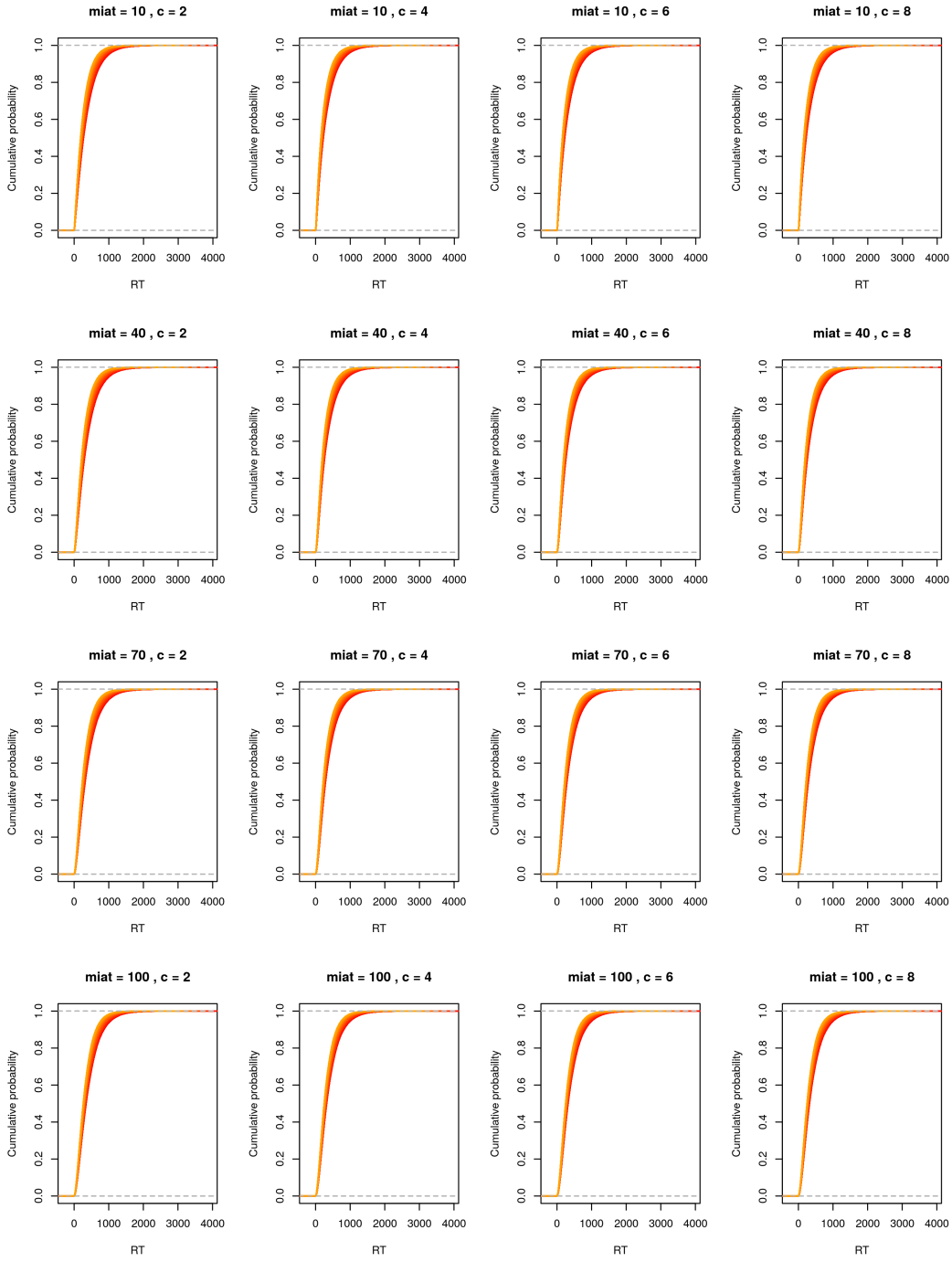

Figure 5.6: Profiles of the ECDFs of model prediction as a function of $\frac{1}{\mu}$ for target-present, $k = 3$, $N = 10^5$. "miat" stands for "mean interarrival time".

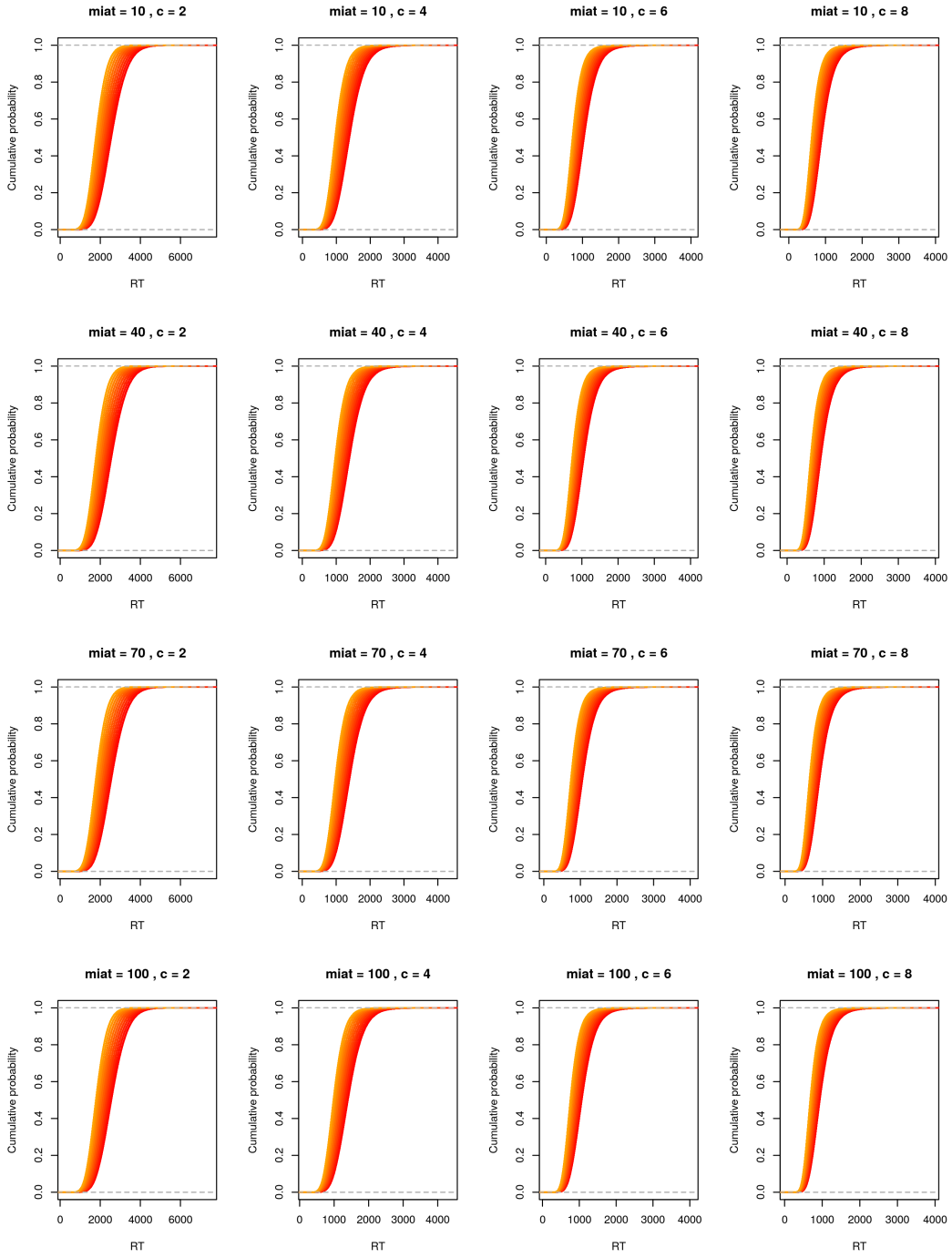

Figure 5.7: Profiles of the ECDFs of model prediction as a function of $\frac{1}{\mu}$ for target-absent, $k = 18$, $N = 10^5$. "miat" stands for "mean interarrival time".

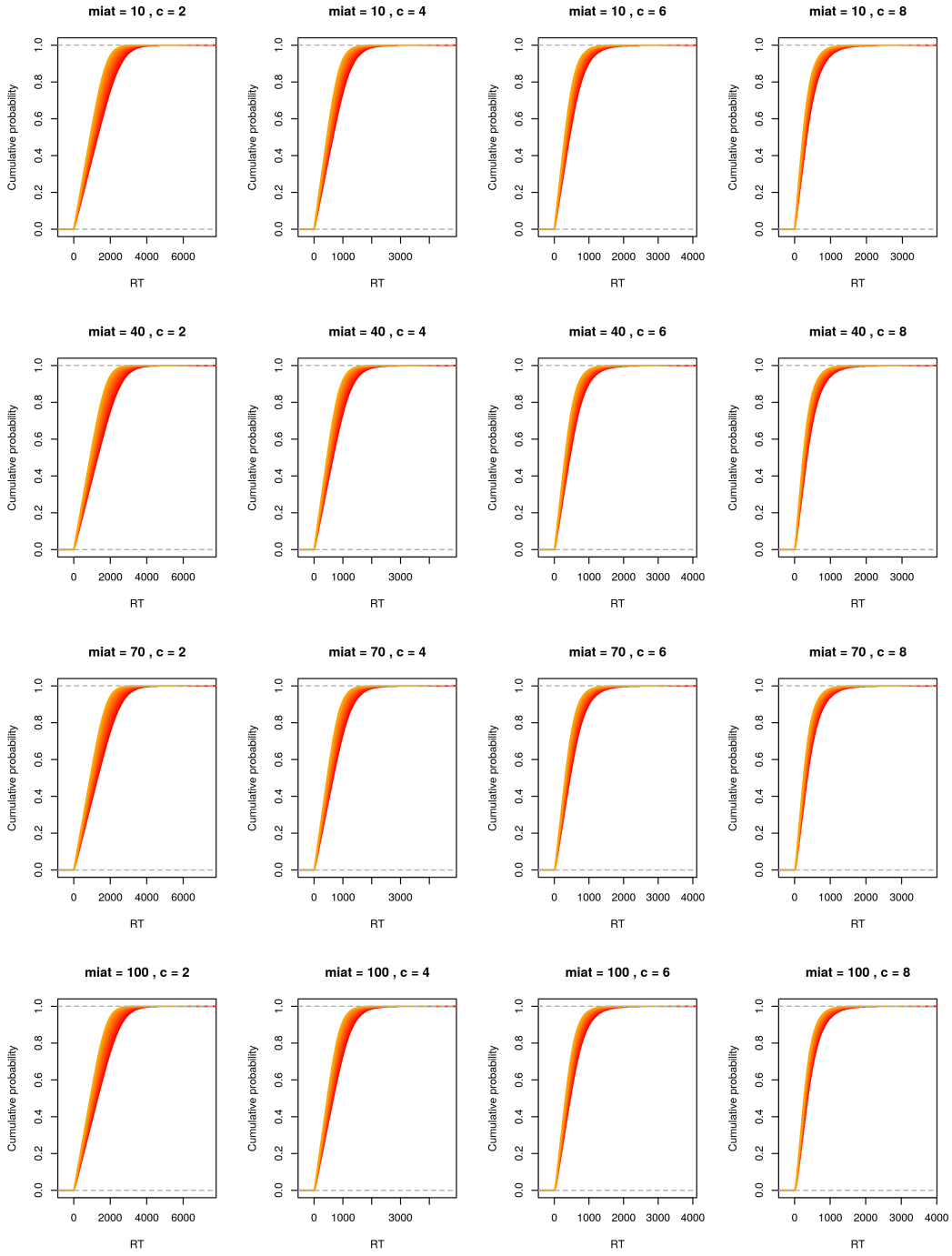

Figure 5.8: Profiles of the ECDFs of model prediction as a function of $\frac{1}{\mu}$ for target-present, $k = 18$, $N = 10^5$. "miat" stands for "mean interarrival time".

5.2.3 Sensitivity of c

Figure 5.9 and Figure 5.10 illustrate the sensitivity profiles regarding c for $k = 3$. Figure 5.11 and Figure 5.12 illustrate the respective profiles for $k = 18$.

The influence of c exhibited a distinctive pattern. There are clear differences among the ECDFs associated with small c but they diminish as c becomes larger. As mentioned above, for $k = 3$ and $c \geq 3$, any server added to the system will not have any effect on the predicted RTs because they will be always idle. It is apparent from Figure 5.9 and 5.10 that after a change in the curvature from $c = 1$ to $c = 2$, the curve of ECDF barely change even from $c = 2$ on under the specified values of remaining parameters. For $k = 18$, the pattern of the influence of c is more apparent and differentiated, yet the differences among the ECDFs for $c \geq 5$ are quite indistinct. The influence of c exhibits some degree of similarity to the influence of $\frac{1}{\mu}$, especially for $k = 3$. $\frac{1}{\lambda}$ did not show any effect that could be detected by the eyes on this pattern, whereas $\frac{1}{\mu}$ mainly led to a shift of the ECDF. Within the same set size level, the change in the curvature caused by c is more apparent in the target-present case than in the target-absent case, although c also leads to a shift in the target-absent case.

These observation indicate that the inverse inference on c from the RTs can be difficult in some circumstances. The largest concern is that $\frac{1}{\mu}$ or $\frac{1}{\mu}$ together with T_{resn} or T_{resy} could confound with c, which leads to identifiability issues. For example, a comparison of Figure 5.5 and 5.9 reveals the possibility of obtaining a similar ECDF by simply increasing $\frac{1}{\mu}$. Similarly, a reduction in $\frac{1}{\mu}$ together with a reduction in T_{resn} or T_{resy} could produce similar changes in Figure 5.11. Nevertheless, the situation does not seem so hopeless as such a concern suggests for two reasons. First, the target-present RTs have a clear advantage in detecting differences attributed to c. In the target-present case, large cs appear to have more distinguishable influence (compare Figure 5.10 and 5.9). For large set size, the influence of c and that of $\frac{1}{\mu}$ are not alike because the resulting (absolute) curvature of the ECDF is quite different. Second, the pattern of change observed in the target-present case cannot be achieved by a shift. Even if a reduction in $\frac{1}{\mu}$ and T_{resn} or T_{resy} could produce similar effect on the ECDF in the target-absent case, the same combination is expect to produce considerable misfit in the target-present case. Thus, combining the informaion from data sets under different

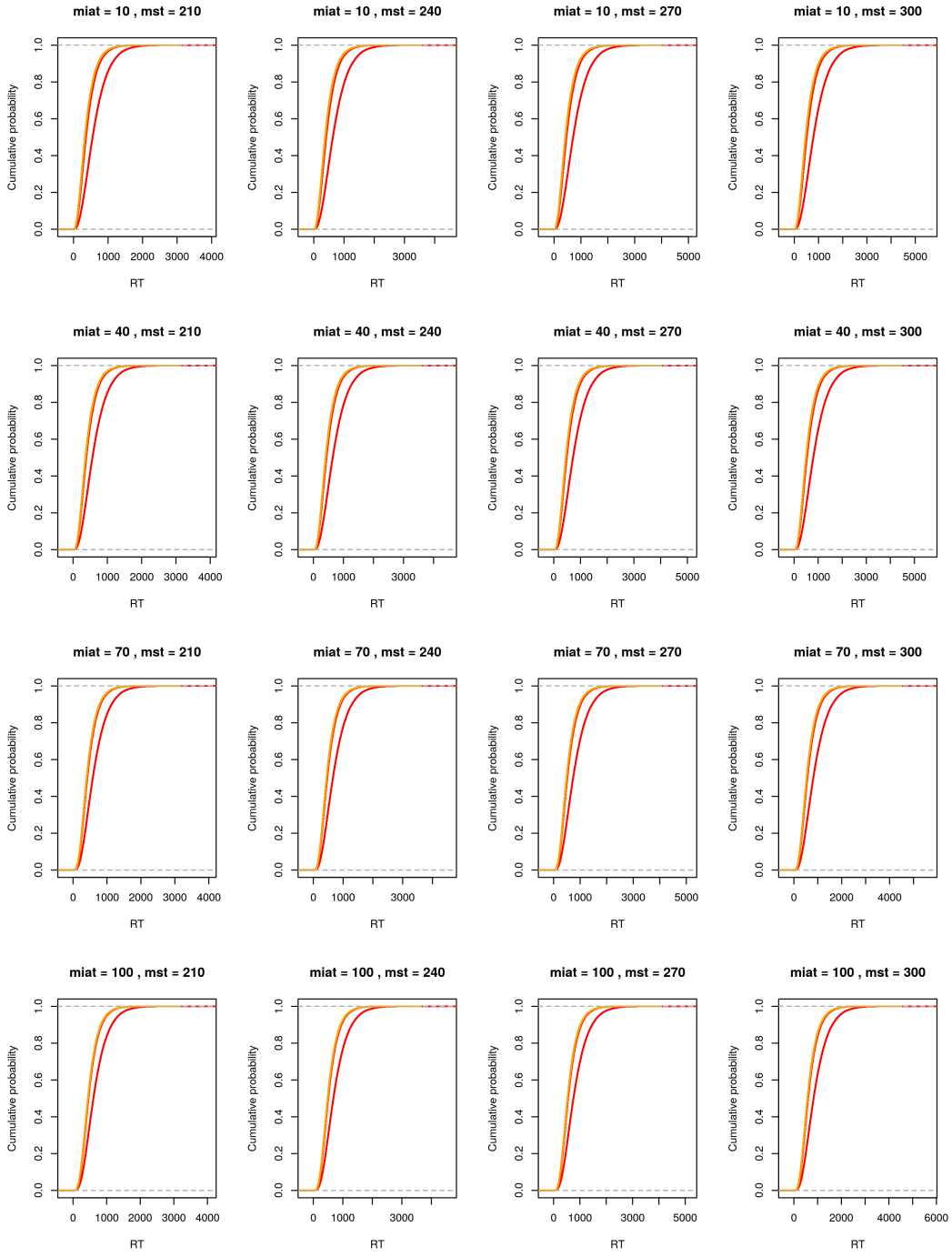

Figure 5.9: Profiles of the ECDFs of model prediction as a function of c for target-absent, $k = 3$, $N = 10^5$. "miat" stands for "mean interarrival time", "mst" for "mean service time".

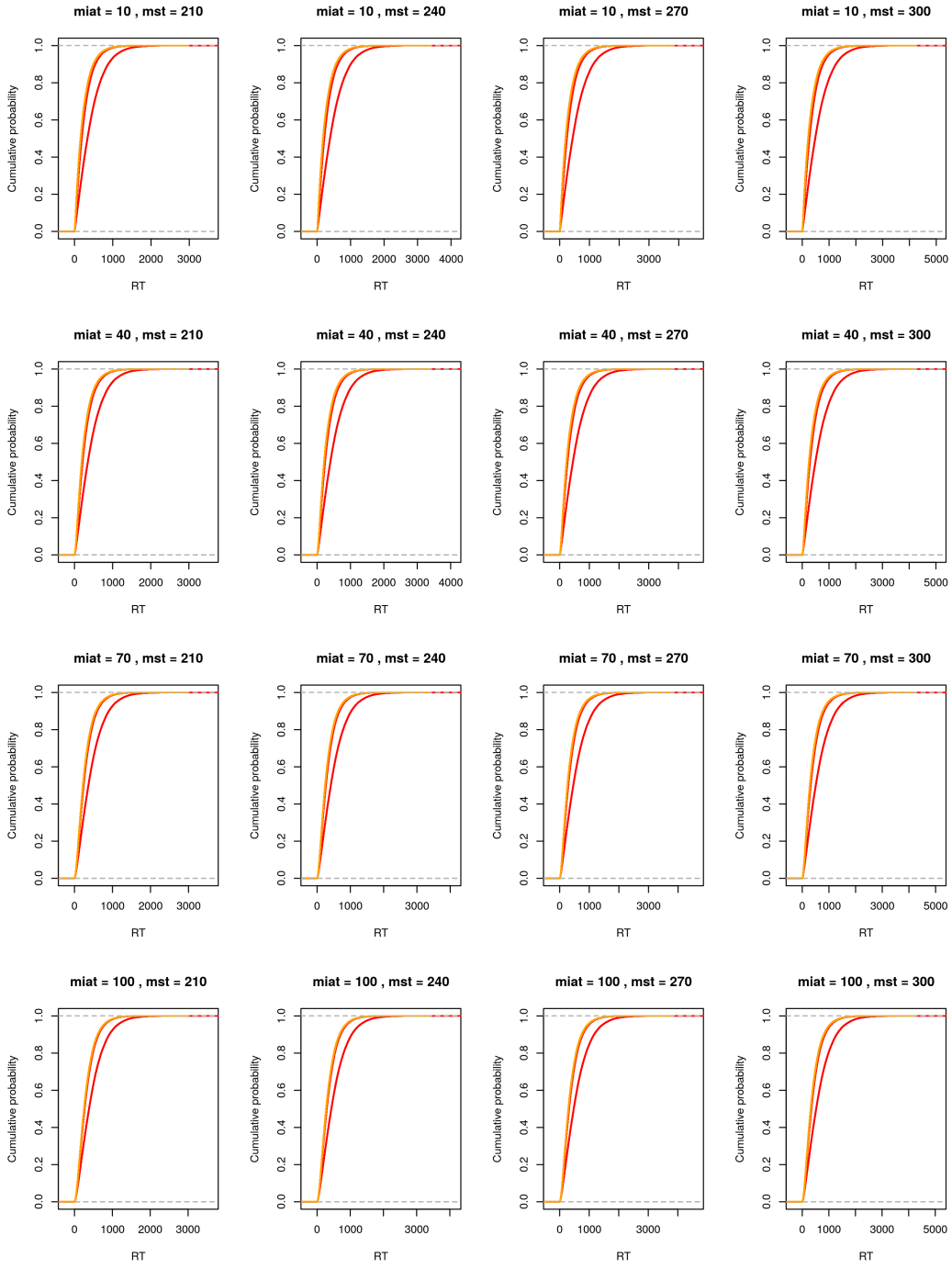

Figure 5.10: Profiles of the ECDFs of model prediction as a function of c for target-present, $k = 3$, $N = 10^5$. "miat" stands for "mean interarrival time", "mst" for "mean service time".

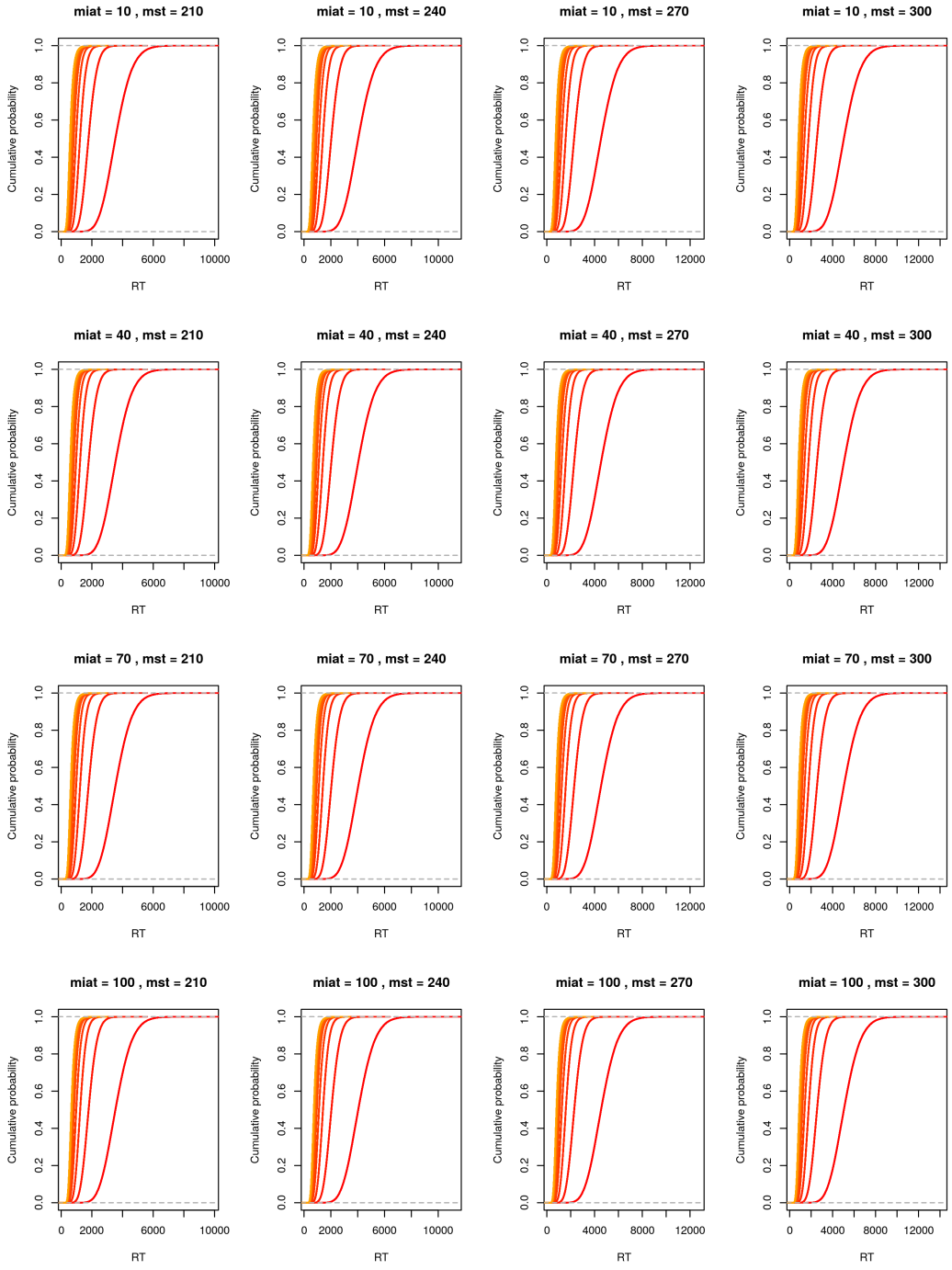

Figure 5.11: Profiles of the ECDFs of model prediction as a function of c for target-absent, $k = 18$, $N = 10^5$. "miat" stands for "mean interarrival time', "mst" for "mean service time".

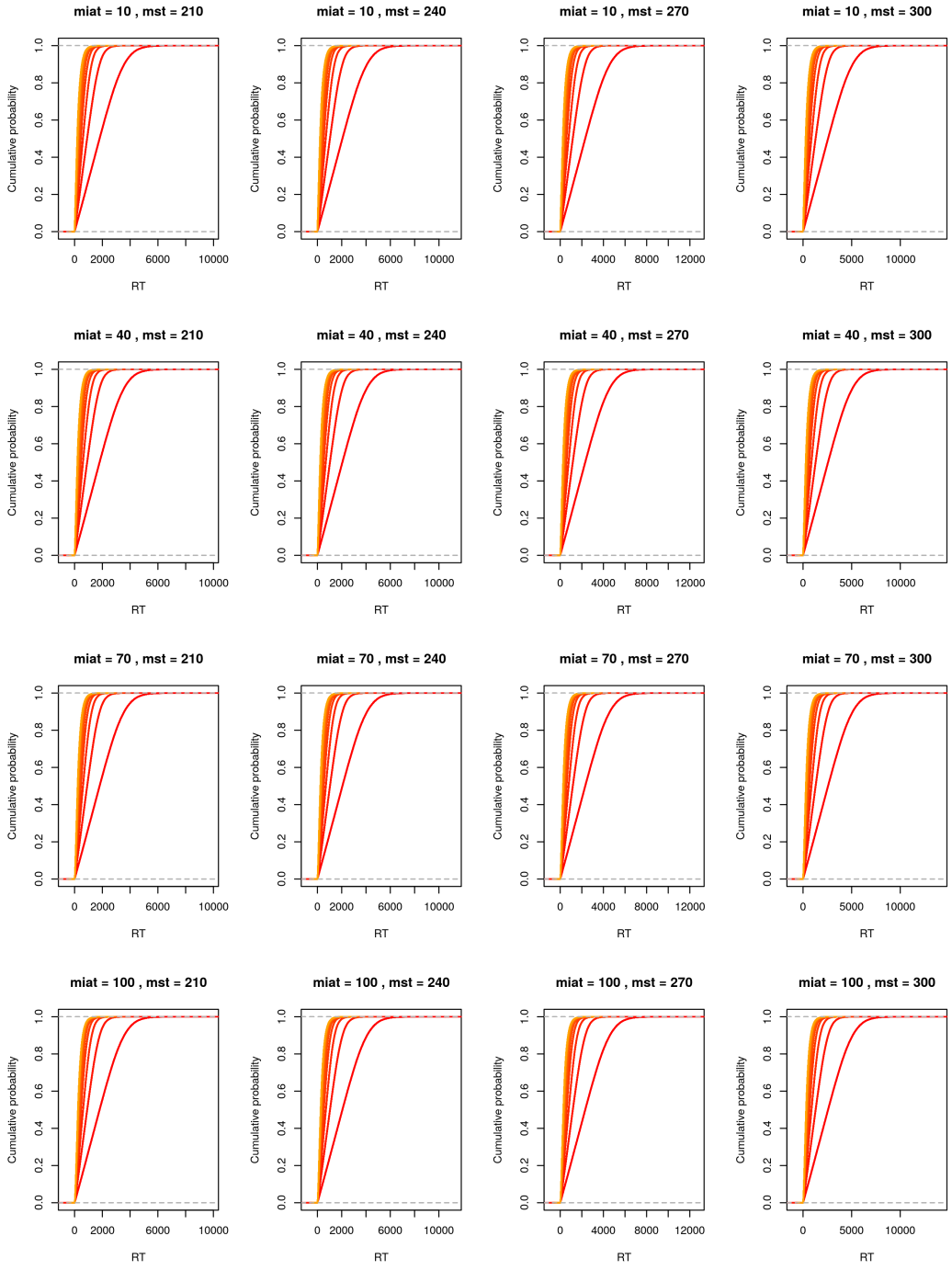

Figure 5.12: Profiles of the ECDFs of model prediction as a function of c for target-present, $k = 18$, $N = 10^5$. "miat" stands for "mean interarrival time", "mst" for "mean service time".

conditions may identify c unambiguously. However, it needs to be born in mind that given the ranges of values as specified in this sensitivity analysis for the other parameters, the model's ability of discriminating $c \geq 5$ is limited.

Chapter 6

Parameter estimation

6.1 Estimating the accuracy-related parameters using MLE

6.1.1 Deriving the MLE for the accuracy-related parameters

Consider first a visual search experiment with a balanced design of m levels of set size and 50% target prevalence. That is, for each set size level, $2w$ trials are realized, whereby the target is present on w trials and absent on the other w trials. Assume that participants are requested to give a response to every trial, i.e., the numbers of "yes" and "no" responses add up to the number of trials. Each "yes" response is viewed as an outcome "1" and each "no" response an outcome "0". For any fixed set size level, the response on one (and each) target-present trial can be considered as a Bernoulli distributed random variable $X_{tp} \sim \text{Bernoulli}(1 - \pi_{md})$ according to Equation (4.16), where

$$\begin{aligned}\pi_{md} =& \mathbb{P}(\text{miss}) \\ =& (1 - a_1 k^{-b})^{k(1-\exp(\alpha+\beta k))-1}((1 - a_1 k^{-b}) \exp(\alpha + \beta k) \\ &+ a_2 k^b (1 - \exp(\alpha + \beta k))).\end{aligned} \tag{6.1}$$

Likewise, according to Equation (4.15), the response on each target-absent trial can be considered as a Bernoulli distributed random variable $X_{ta} \sim \text{Bernoulli}(\pi_{fa})$

with

$$\pi_{fa} = \mathbb{P}(\text{false alarm}) = 1 - (1 - a_1 k^{-b})^{k(1-\exp(\alpha+\beta k))}. \tag{6.2}$$

Assuming the stochastic independence of responses on different trials, the number of "yes" responses on w target-present trials is binomial distributed: $Y_{tp} \sim \mathrm{B}(w, 1 - \pi_{md})$. Similarly, the number of "yes" responses on w target-absent trials is also binomial distributed: $Y_{ta} \sim \mathrm{B}(w, \pi_{fa})$.

For the $2w$ trials with set size k, let $y_{tp,k}$ and $y_{ta,k}$ denote the observed number of "yes" responses on the target-present and target-absent trials, respectively. Then the likelihood of observing the response pattern $(y_{tp,k}, y_{ta,k})$ is given by

$$L(\pi_{md}, \pi_{fa}) = (1 - \pi_{md})^{y_{tp,k}} \pi_{md}^{n-y_{tp,k}} \pi_{fa}^{y_{ta,k}} (1 - \pi_{fa})^{n-y_{ta,k}}$$

Generalizing this to describe the likelihood of observing the response pattern $((y_{tp,k_1}, y_{ta,k_1}), \ldots, (y_{tp,k_m}, y_{ta,k_m}))$ in the entire experiment with set size levels $(k_1, \ldots, k_m)$, we obtain

$$\begin{aligned} &L(\pi_{md,k_1}, \pi_{fa,k_1}, \ldots, \pi_{md,k_m}, \pi_{fa,k_m}) \\ =&C \prod_{i=1}^{m} (1 - \pi_{md,k_i})^{y_{tp,k_i}} \pi_{md,k_i}^{w-y_{tp,k_i}} \pi_{fa,k_i}^{y_{ta,k_i}} (1 - \pi_{fa,k_i})^{w-y_{ta,k_i}}, \end{aligned}$$

whereby C is a constant, and the corresponding log-likelihood

$$\begin{aligned} &l(\pi_{md,k_1}, \pi_{fa,k_1}, \ldots, \pi_{md,k_m}, \pi_{fa,k_m}) \\ =& \ln C + \sum_{i=1}^{m} \big(y_{tp,k_i} \ln(1 - \pi_{md,k_i}) + (w - y_{tp,k_i}) \ln(\pi_{md,k_i}) \\ &+ y_{ta,k_i} \ln(\pi_{fa,k_i}) + (w - y_{ta,k_i}) \ln(1 - \pi_{fa,k_i}) \big). \end{aligned} \tag{6.3}$$

Note that the parameter estimation using MLE does not require a balanced design. The numbers of target-absent and target-present trials with different set size levels are not required to be equal. The only role the numbers of trials play is to infer the numbers of "no" responses, which is apparent in Equation (6.3). Therefore, the estimation method can be applied in the situation where participants are allowed to skip trials (i.e., the sum of the numbers of "yes" and "no" responses is less than the number of trials), which is the case in the study

by Wolfe et al. (2010)[1]. For the general case where the numbers of "yes" and "no" responses on each set size level are given, we simply need to replace the terms $w - y_{ta,k_i}$ and $w - y_{tp,k_i}$ in the log-likelihood function by n_{ta,k_i} and n_{tp,k_i} respectively, using n_{ta,k_i} and n_{tp,k_i} to denote the "no" response on target-absent and target-present trials with set size k_i:

$$\begin{aligned} & l(\pi_{md,k_1}, \pi_{fa,k_1}, \ldots, \pi_{md,k_m}, \pi_{fa,k_m}) \\ &= \ln C + \sum_{i=1}^{m} \big(y_{tp,k_i} \ln\big(1 - \pi_{md,k_i}\big) + n_{tp,k_i} \ln\big(\pi_{md,k_i}\big) \\ &\quad + y_{ta,k_i} \ln\big(\pi_{fa,k_i}\big) + n_{ta,k_i} \ln\big(1 - \pi_{fa,k_i}\big) \big). \end{aligned} \tag{6.4}$$

Since the arguments $(\pi_{md,k_1}, \pi_{fa,k_1}, \ldots, \pi_{md,k_m}, \pi_{fa,k_m})$ of l are functions of the parameters $(\alpha, \beta, a_1, a_2, b)$, substituting Equation (6.1) and (6.2) in Equation (6.4), we obtain a function of $(\alpha, \beta, a_1, a_2, b)$, which is denoted by

$$\widetilde{l}(\alpha, \beta, a_1, a_2, b) = l(\pi_{md,k_1}, \pi_{fa,k_1}, \ldots, \pi_{md,k_m}, \pi_{fa,k_m}). \tag{6.5}$$

The function $\widetilde{l}(\alpha, \beta, a_1, a_2, b)$ is then to maximize. Its derivatives with respect to $\alpha, \beta, a_1, a_2, b$ are given in Appendix D.

6.1.2 Implementation of the MLE approach for accuracy-related parameters

The problem of estimating the accuracy-related parameters is now transformed into an optimization problem with $\widetilde{l}$ in Equation (6.5) as the objective function and $(\alpha, \beta, a_1, a_2, b)$ as the arguments to maximize over. The MLE approach was implemented in `R`, using a multidimensional general purpose optimization routine `optim` of the package stats. For its application, the user needs to specify the initial values of the search, boundaries for the parameters and the objective function. Besides the gradient, optional arguments are the search method and

[1] However, only a slight deviation from a balanced design of set size × target presence is allowed. Recall that the stopping policy relies on the assumptions of 50% target presence and equiprobable set size levels from the perspective of the searcher (see Section 4.3). An apparent deviation from a balanced design regarding these two factors may induce a different stopping behavior.

control arguments, such as scaling values for the objective function, for the gradient and for the parameters (see R manual for details).

The objective function $\widetilde{l}(\alpha, \beta, a_1, a_2, b)$ can be represented straightforwardly. Its gradient was calculated based on the derivatives (Appendix D) and passed to *optim* as well. Since the objective function is multidimensional and differentiable with constraints on all parameters, L-BFGS-B was chosen as the search method.

The most apparent difficulty of implementing MLE for the parameter estimation in the current case is to specify appropriate boundaries for each parameter. The choice of initial values as well as the scaling information in the control arguments also depend on the concrete bounded constraints. On the one hand, there is not always obvious rational foundation for a specific choice of boundaries such that these boundaries have convincing, psychologically meaningful interpretations. On the other hand, setting appropriate boundaries is necessary because it affects the performance of the optimization process to a large extent.

To make sure all variables in the model are bounded in the way the model specifies, a system of inequalities must be satisfied:

- $a_1, a_2, b > 0$
- $-7 < \alpha < 0, \beta > 0, \frac{\alpha}{\beta} < -\max k$
- $a_1(\min k)^{-b} < 1, a_2(\max k)^b < 1$

Then the values of the corresponding boundaries can be obtained by solving this system of inequalities for each parameter. Unfortunately, some of these necessary boundaries are not simply a numeric value but a variable involving another parameter. This also reveals the second issue: parameter correlations. α and β appear to be correlated in the parameter space and so are a_2 and b. Correlations among parameters (also known as collinearity) are undesirable from a numerical optimization point of view. The influence of parameter correlation on the search for optimal parameters will be described in greater details in Section 6.2.3.

To solve these two issues, reparameterization was applied to the optimization problem. Let k_{max} denote the largest set size level realized in an experiment: $k_{max} = \max\{k_1, \ldots, k_m\}$. The parameter β is replaced by

$$\tilde{\beta} = -\frac{k_{max}}{\alpha}\beta$$

and b is replaced by

$$\tilde{b} = -\frac{\ln(k_{max})}{\ln(a_2)} b.$$

In this way, the lower bound for the arguments $(\alpha, \tilde{\beta}, a_1, a_2, \tilde{b})$ is determined as $(-7, 0, 0, 0, 0)$ and the upper bound $(0, 1, 1, 1, 1)$.

The choice of the starting point for the algorithm influences how quick the search converges and in some circumstances (e.g., if the function has multiple local optima) to which local optimum the search converges. In the present case, recovery studies and fitting to empirical data did not show a noticeable effect of the initial values except for $\tilde{\beta}$, which must be treated specially using profiling (see below). Therefore, the mean of the lower and upper bounds is chosen as the starting value for each parameter.

The scaling arguments accept information on the scaling to the objective function and gradient values as well as the scaling values for the parameters. By default, the optimizer treats all variables in the same way, but it performs better when the scales of all variables are comparable. Hence, the scaling arguments are particularly worth trying out when parameters and/or function values have strongly different scale, leading to large differences in the step size and error. In the current case, once the bounds are set, the optimization can be run a few times starting from different points, without specifying any scale values. The iteration history of the optimization can provide some hints on the scale of the variables, which enables the specification of the scaling arguments. For reference, the final settings for the data fitting in this dissertation were -1 for the overall scaling to the function value and $(4.4, 0.45, 0.019, 0.0002, 0.3732)$ for the vector scaling of parameters.

Since there are signs for a slight dependency of the result on the starting values of $\tilde{\beta}$, profiling needs to be applied. The concrete approach is as follows. The parameter domain of $\tilde{\beta}$ is divided into n adjacent, disjoint intervals of equal length, numbered by $1, \ldots, n$. Then n optimization processes are run. For the i-th optimization, the lower bound of $\tilde{\beta}$ takes the left hand endpoint and the upper bound of $\tilde{\beta}$ the right hand endpoint of the i-th interval. The result that yields the largest function value across the n runs is selected as the final result.

Interested readers are referred to Appendix E for more detailed information on the profiling approach.

A key component of the approach is the degree of resolution, i.e., how fine the partition is. The more intervals the parameter domain is partitioned into, the more likely the optimizer finds the global optimum. However, a finer partition also leads to higher computational cost. For the MLE approach implemented here, an optimization run turns out to take only a few seconds, so that this aspect does not play an important role for the decision. In the current case, a partition of 10 intervals provides satisfactory results.

The R code that implements the MLE approach for the estimation of accuracy-related parameters is given in Appendix F.

6.2 Estimating the RT-related Parameters

6.2.1 A minimum distance estimation approach based on the distribution function

To estimate the RT-related parameters, I used the minimum distance estimation (MDE, e.g. Basu et al., 2010; Parr, 1981), minimizing the distance between the CDF of the empirical data and the CDF of the simulated data. As discussed in Section 5.1, the estimation of RT-related parameters relies on Monte Carlo simulation, since no explicit analytical from of CDF or PDF of the model prediction is available. The approaches applicable to such a situation fall into two broad categories: maximizing an approximative/estimated likelihood function (e.g., quasi-maximum likelihood estimate, also known as pseudo-likelihood estimate) or minimizing the closeness between the data and the model (e.g., MDE). Within each category, there are many different ways of realizing the basic idea of the category.

I opt for an MDE approach based on distribution functions for two reasons: feasibility/efficiency consideration and the quality of the estimator. First, this approach is straightforward and computationally efficient. Second, minimum distance estimators based on CDF possess several desirable statistical properties, most importantly robustness and consistency (e.g., Basu et al., 2010; Boos, 1981,

1982; Parr, 1981; Parr & De Wet, 1981; Parr & Schucany, 1980). A detailed argumentation for the choice is given in Appendix G.

The choice of the functional as the distance measure between the distribution functions also plays an important role in determining the performance of the estimation procedure. An appropriately defined distance leads to more efficient computation and estimates with more desirable properties. Distance measures commonly used in CDF-based MDE are the Kolmogorov-Smirnov distance, the Cramér-von Mises distance and some weighted versions of them, such as the Anderson-Darling distance (see e.g. Parr, 1981; Parr & Schucany, 1980, 1982). A further distance measure that is less frequently applied and studied in the context of MDE is the Wasserstein metric. Compared to the Kolmogorov-Smirnov distance and the Cramér-von Mises distance, the Wasserstein metric of order 1 has the advantages of being less sensitive to local deformations, sensitive to translations, numerically more stable and computationally efficient to implement. Therefore, the Wasserstein metric is chosen as the distance measure of the MDE approach. A detailed comparison of the three distance measures is given in Appendix H.

Let F_θ denote parametric model CDF with parameter θ and G_n denote the empirical cumulative distribution function (ECDF) of the empirically observed sample of data with sample size n. According to Vallender (1974), the Wasserstein metric of order 1 is given by

$$W_1(G_n, F_\theta) = \int_{-\infty}^{\infty} |G_n(x) - F_\theta(x)| \, \mathrm{d}x.$$

The `R` code for the calculation of the Wasserstein metric for the queueing model is given in Appendix C.

6.2.2 The objective function

After specifying a distance measure, the next step of the parameter estimation using the MDE approach is to take the distance as the objective function and to find the parameter values that minimize this objective function using numerical optimization.

As discussed in Section 4.6, the RTs of incorrect responses should be taken into account. The question of how to define the overall distance arises. Simply

taking the sum of both distances as the objective function may bias the results considerably. As discussed in Section 4.6, incorrect responses are much rare than correct responses by nature. Thus the pattern of their RTs is usually more distorted by randomness compared to the RTs of correct responses. Consequently, the approximation of the CDF based on the data (regardless of simulated or empirical) is prone to distortions. Whereas a more accurate approximation of the CDF can be obtained by increasing the total number of simulations, there is usually a limit to the number of empirical observations. Suffered from the small sample size, the distance for incorrect RTs is expected to be larger than that for correct RTs, even when the model essentially predicts the two random variable equally well. If we simply take the sum of both distances as the objective function, the distance for incorrect RTs will dominate such that the optimizer searches for parameter values in their favor.

Therefore, a better approach is to weight the distances by the relative frequency of the data and take this weighted sum as the objective function of the optimization. In this way, the more representative and reliable the data, the more influence they are allowed to exert on the optimization. Because the relative frequencies of correct and incorrect responses are positive and adds up to one, the weighted sum is a convex combination of the distances.

This construct is not only a pragmatic solution which realizes a desirable property, but also mathematically meaningful. It can be proven that a convex combination of two metrics is a product metric on the Cartesian product of the two corresponding metric spaces, if and only if both coefficients of the convex combination are non-zero[2] This means that all the mathematical properties of a metric are inherited and statements that are valid for metrics are applicable to this construct as well.

It is reasonable to ask the question of whether taking incorrect RTs into account using this method in fact improves the parameter estimation. Since the distance for incorrect RTs is weighted by their frequency, which is usually quite

[2]The proof is analogous to the proof for the same statement on the unweighted sum of the two metrics, which is a canonical construct. The statement can be generalized to Cartesian product of finitely many metric spaces and any linear combination with strictly positive coefficients. The premise that both coefficients are non-zero means in the application to the current context that both correct and incorrect responses have non-zero frequency in each data set. This is usually fulfilled for sufficiently large sample size.

low, its influence on the weighted sum is limited. It is possible that the influence of incorrect RT is too small to justify the additional computational expense. In the current context, incorporating incorrect RTs in the modeling paid off. In the recovery studies, taking incorrect RTs into account mostly led to smaller deviation of the estimates from the true value than fitting to only the correct RT.

Nevertheless, the benefit of this approach is constrained by the total amount of empirical observations. In the recovery studies, for conditions in which errors appear the rarest, i.e., target-absent with the largest set size, the influence of incorporating incorrect RTs on the estimates for these conditions turns out to be negligible. However, this is only a local phenomenon which only happened when the model was fitted to only a sub-data set (i.e., data set collected under a specific set size × target presence condition), which corresponds to the first step in 6.2.4). When data sets under all conditions are fitted to together with the restriction that all parameters (or all except for the mean service time, see Section 7.2) should be invariant to the experimental conditions, including the incorrect RTs improved the parameter estimation.

6.2.3 Identifiability of the RT model

So far, the technical details necessary for fitting a single (sub-)data set under a specified condition have been elaborated. They constitute the foundation of a conjoint fitting procedure for all sub-data sets from an experiment. To achieve the goal of obtaining reliable and accurate parameter estimates that explain data under all the conditions an experiment realizes, there are two questions to answer:

- Is the performance of this foundation, i.e., the fitting of sub-data sets adequate?
- How to combine the information extracted from each sub-data set to form an accurate overall picture of the parameters?

To answer the first question, the identifiability of the RT model needs to be investigated. Since no analytical representation of the model prediction is available, the identifiability cannot be explored with mathematical rigor.

However, it is possible to explore this issue by numerical methods. In fact, several identifiability issues were observed, including dependence on the initial values, convergence issues and deviations from the true parameter values in recovery studies. There are several factors behind these identifiability issues, such as ill-conditioning, collinearity and overparameterization. The techniques adopted to deal with these problems are described, including profiling with regard to $\frac{1}{\lambda}$, allowing the residual time parameter to depend on the type of the response, and an iterative procedure fixing a subset of the parameters and its complementary subset alternately. As an answer to the second question, I elaborate these techniques in Section 6.2.4, integrate them as a unified procedure that fits data sets under different conditions conjointly.

6.2.3.1 Identifiability issues

A property of the model is the different constraints on possible parameter values. The model in its original form (without distinction between T_{resn} and T_{resy}) has four parameters for RT modeling: the mean interarrival time $\frac{1}{\lambda}$, the mean service time $\frac{1}{\mu}$, the residual time T_{res} and the number of servers c. Among these, c can only take values from natural numbers and must be restricted (for plausibility reasons). Since there is so far no empirical finding that allows a direct inference on how large c could be at the most, I consider 10 as a proper number for the upper bound of this construct[3] This means that the estimation for c can be achieved by profiling, which separates it from the estimation for other parameters. In other words, c must be fixed in the course of an optimization procedure which optimizes over all other parameters. The optimization procedure must be run for all natural numbers that are considered potentially suitable value of c. After this procedure has been repeated for all c, the value of c that yields the best fit will be selected. According to this procedure, the parameters that are estimated from the sub-data set under each specified condition (set size × target presence) are $\frac{1}{\lambda}$, $\frac{1}{\mu}$ and T_{res}.

In general, fitting the model to simulated sub-data sets under a specified set size × target presence condition showed unsatisfactory performance of the

[3]If the ranges of $\frac{1}{\lambda}$ and $\frac{1}{\mu}$ specified here are correct, the model will predict a search slope inconsistent with the empirically observed slopes for $c \geq 10$.

parameter estimation. It resulted in estimates that deviated from the true values to a considerable extent. The deviation depends on the parameter and the experimental condition. The largest relative deviation was observed for $\frac{1}{\lambda}$. The smallest relative deviation was observed for $\frac{1}{\mu}$. Target-present sub-data sets yielded generally better fits than target-absent ones. The fit decreases as set size increases. In more detailed investigations, several adverse phenomena were observed.

First, for both the simulated data in the recovery studies and the empirical data collected by Wolfe and colleagues (E. M. Palmer et al., 2011; Wolfe et al., 2010), a dependence of the result on the initial values was observed. More specifically, the starting value of $\frac{1}{\lambda}$ seem to has a universal influence. Not only the estimate for $\frac{1}{\lambda}$ itself, but also those for other parameters, above all T_{res}, changed when different starting values of $\frac{1}{\lambda}$ were given to `optim`[4]. The initial values of other parameter seemed to have an influence as well, but the change in the initial values had to be very large to induce a change in the estimates and induced the change was usually small. Especially the initial value of $\frac{1}{\mu}$ exhibited negligible effect on the estimates for all parameters. This indicates an identifiability issue of $\frac{1}{\lambda}$. It is in line with the observation in the sensitivity analysis (Section 5.2) that changing $\frac{1}{\lambda}$ has weak influence on the distance in comparison with other parameters. In contrast, the estimate of $\frac{1}{\mu}$ was affected to a fairly limited extent. In the recovery studies, neither changes in the starting value of any parameter nor a biased estimate of other parameters were able to distort the estimate of $\frac{1}{\lambda}$ to a considerable extent. Nevertheless, even $\frac{1}{\mu}$ became unstable for the empirical data sets with large set size levels, especially in the target-absent case.

Second, for some empirical data sets, at least one of the parameters was always estimated at one of its bounds, although another value could have obviously yielded a better fit. Apparently, the optimization algorithm failed to explore areas away from the bounds for these parameters. This phenomenon occurred above all to $\frac{1}{\lambda}$. The iteration history of the optimization process shows that the

[4]The dependence on the initial values was observed for all empirical sub-data sets except for those with large set size levels and target absence. However, this is due to the fact that the fit for these sub-data sets were very bad in general and the parameters were always estimated at the bounds.

search was restricted to a small neighborhood of the bounds or the starting value, although it was not imposed any restriction on. This is an abnormal behavior.

A further investigation using profiling on $\frac{1}{\lambda}$ reveals the complication of these two issues (see Section 6.1.2 for details about the profiling method). The objective function appears to have quite rough functional surface along $\frac{1}{\lambda}$, such that many local minima that are hard to distinguish gather intensively.

Furthermore, profiling on $\frac{1}{\lambda}$ indicates a near linear dependence (also known as collinearity) between $\frac{1}{\lambda}$ and T_{res}. The influence of T_{res} appears to be able to partially compensate the influence of $\frac{1}{\lambda}$. For instance, reducing $\frac{1}{\lambda}$ (within certain extent) and increasing T_{res} at the same time can yield similar discrepancy. Collinearity among parameters is usually obstructive for finding the global minimum. The ideal condition for the optimization algorithm to work effectively is one in which the influence of the parameters is independent of each other (geometrically orthogonal) in the sense that the contour-lines of the functional surface are disjoint convex sets in the parameter space, for example, having the form of circular ripples. The search process can approach a local minimum easily by moving along the directions parallel to the axes of the parameter space (i.e., by changing the value of only one of the parameters). However, if a collinearity exists, the contour-lines of the functional surface may be non-convex, but rather twisted in the parameter space.

Collinearity is difficult to deal with in general. A common approach is to reparameterize the model to transform the parameter space into a more favorable one. The goal is to transform the parameter space such that the contour-lines of the functional surface become less twisted and more regular. However, there is hardly any general rule for how to find the transformation that works the best for a specific problem. Sometimes the transformation works out and solves the collinearity problem, but sometimes it makes the problem worse by twisting the contour-lines in a wrong direction. Unfortunately, the transformations I tried out could not solve the current problem.

An additional phenomenon is the exchange of the rolls of $\frac{1}{\lambda}$ and $\frac{1}{\mu}$. However, its occurrence was restricted to the case $c \geq k$ and when the parameter domains of $\frac{1}{\lambda}$ and $\frac{1}{\mu}$ overlap. In the area where they overlap, exchanging the value of $\frac{1}{\lambda}$ and $\frac{1}{\mu}$ yielded indistinguishable fit. This phenomenon disappeared after the

ranges of $\frac{1}{\lambda}$ and $\frac{1}{\mu}$ were adjusted to be disjoint. The adjustment is justified for the current problem. A detailed discussion on setting the domains of $\frac{1}{\lambda}$ is given in next subsection.

These adverse phenomena result in convergence issues of the optimization process. Sometimes the search failed to converge to a minimum within the given iteration steps. Even if it converged, the results deviated from the true parameter values considerably. Deviations were observed even if the true values were given as the initial values or as the vector scaling of parameters.

In summary, the identifiability of the model can be characterized as "the fragile $\frac{1}{\lambda}$, the robust $\frac{1}{\mu}$ and the interfering T_{res}." The identifiability issues concern clearly $\frac{1}{\lambda}$ and T_{res}. Since the mean interarrival time $\frac{1}{\lambda}$ is of larger theoretical interest than the residual time T_{res}, the following discussion focuses on the factors causing identifiability issues regarding $\frac{1}{\lambda}$.

6.2.3.2 Factors causing identifiability issues

The identifiability issues (especially regarding $\frac{1}{\lambda}$) appear to be a product of several factors. First, the model has the property that during the course of queueing, the effective arrival rate is proportional to the number of customers that have not been in the system (see Section 3.4.2). This means that the *effective* arrival rate varies by nature depending on the state of the system. As mentioned in Section 3.4, the pattern of system response time (under either stopping criterion) is determined jointly by the effective arrival rate and the system service capacity. Even if $\frac{1}{\lambda}$ changes to a considerable extent relatively (measured by percentage), the change in the pattern of system response time may be able to reflect only a limited part of it[5]. Note that not all arrivals contribute information to the model prediction because the queue may have been stopped before some of the arrivals can happen. The probabilities of the system being in each state, determine whether a certain arrival will happen at all before the queueing stops according

[5]Consider for example the following case. For a set size of ten, if the true mean interarrival time $\frac{1}{\lambda}$ is 50, then the effective mean interarrival time for the $1, 2, 3, ..., 10$ arrival (i.e., at the moment when the number of customers outside the system becomes $10, 9, 8, ..., 1$) will be $5, \frac{50}{9}, \frac{25}{4}, ..., 50$. On a scale of 1 to 100, these do not differ much from $8, \frac{80}{9}, 10, ..., 80$, which are the effective mean interarrival time under the assumption of a $\frac{1}{\lambda}$ of 80.

to the termination rule. These probabilities depend in turn on the relative ratio of the effective interarrival time to the service capacity[6].

The second reason is that the service time $\frac{1}{\mu}$(average time required for recognition) is usually much longer than $\frac{1}{\lambda}$ (average time required for perception of single features). The speed of the processing (estimated by the slope of RT on set size) provides hints on the scale of the mean interarrival time[7]. It allows inference on a rough upper bound for $\frac{1}{\lambda}$. Assuming a set size of 10, if $\frac{1}{\lambda}$ was much larger than the slope, say, larger than 100 ms, then finishing the first step of processing alone for the $1, 2, ..., 10$ item would have taken $10, 20, \ldots, 100$ ms on average according to the model. Even if there was no further processing step, the time required for all ten items to arrive would be 550 ms on average, yielding a slope of 55 ms/item. Then we would not have been able to observe slopes mostly falling in the range of 20 to 50 ms.

On the other hand, the results of the studies on attentional dwell time provides the scale for the mean service time. From the study of Theeuwes et al. (2004) (see Section 2.5.2), it can be inferred that the time required to relocate attention from a location to another amounts to about 250 ms (in their experimental setting). Considering that the stimuli used in their experiment were very simple (arrows and probe dots), it is plausible to view this amount of time as a rough lower bound to the mean service time[8]. Taken both sources of information into consideration, it appears that the $\frac{1}{\lambda}$ should be about three to ten times as large as $\frac{1}{\lambda}$. This implies that for any queueing system with less than 10 servers (which is a plausible assumption), the queue is likely to be congested. It follows that the pattern of the system response time, regardless of target-absent or target-present trial, will be determined mainly by $\frac{1}{\mu}$ and c. The information of $\frac{1}{\lambda}$ reflected in the system response time will be easily concealed because $\frac{1}{\mu}$ is dominant. In the target-present case, the information of $\frac{1}{\lambda}$ (contribution of $\frac{1}{\lambda}$) in the system

[6]In the example above, if the 10-th arrival does happen (e.g., in case of target-absent trial and relatively congested queue), 50 and 80 are easier to differentiate numerically.

[7]According to the queueing model, the slope of RT on set size is determined jointly by the effective arrival rate, the service capacity (i.e., $\frac{1}{\mu}$ and c) and the stopping criterion. Therefore, it does not reflect $\frac{1}{\lambda}$ directly. However, the slope allows inference on the range of $\frac{1}{\lambda}$.

[8]Strictly speaking, this measure must have included the mean arrival time of the stimulus as well. Even though, the mean service time should not be lower than 150 ms according to this finding.

response time may be able to be revealed to a larger extent. It includes cases where the queue stops early enough (e.g., after the first or second service), when the queue has not yet become congested. Nevertheless, since these cases involve only the first a few arrivals, the effective interarrival time will be numerically more difficult to differentiate for the first reason.

A third factor that contributes to the identifiability issue regarding $\frac{1}{\lambda}$ is overparameterization. Unlike the first two factors, this is not an aspect of the model's nature, but rather a modeling issue. The model appears to be overparameterized by possessing a shift parameter representing the residual time. There are signs indicating that this is related to the collinearity between $\frac{1}{\lambda}$ and T_{res} described above. A further investigation shows that this issue is resolved once the number of parameter is reduced in a proper way. If the T_{res} is fixed to the true value (or a value close to the true value), the estimates of both $\frac{1}{\lambda}$ and $\frac{1}{\mu}$ become accurate and the estimation process becomes stable and behaves nicely (search occurs within the entire range). The dependence on starting value is also weakened. This means that T_{res} is probably straining the model to be difficult to identify. If the true value of T_{res} (or reasonably close ones) is given to the model, the identifiability issues diminish.

6.2.3.3 Solving identifiability issues

The overparameterization issue has more profound meaning. The first aspect concerns the dependence of the residual time on response type. It turns out that if the residual time is in fact different for "yes" and "no" response, assuming a common residual time for both types of responses will lead to a more severe identifiability issue. In a recovery study, data was simulated assuming different T_{res} for "yes" and "no" responses. The results show that fitting a model with one single common T_{res} for both response types to these simulated data yields bad model fit and severely distorted estimates. Hence, allowing T_{res} to depend on the type of response is a reasonable modeling choice. Doing this means adding a parameter to the model. Note that this is different from relaxing the constraint that a parameter should be constant across all experimental conditions, because the response type is not an experimental factor. Both "yes" and "no" responses can appear in every experimental condition. Thus two parameters

for the residual time are always required when fitting the model to every single sub-data set.

Now, let us return to the notation of T_{resn} representing the residual time of "no" responses and T_{resy} that of "yes" responses. From a modeling point of view, differentiating the residual time for "yes" and "no" responses includes the case where both residual times are the same and is hence applicable regardless of whether the assumption of different residual times is in fact valid. However there is a dilemma. This approach induces an additional parameter, which may make the overparameterization issue more severe. This phenomena was observed in a recovery study. The estimation became more unstable and the behavior more chaotic. Nevertheless, the recovery study shows that once T_{resn} and T_{resy} were fixed to the true values (or reasonably close ones), all the issues associated with overparameterization were resolved.

The second aspect related to overparameterization concerns condition-dependent parameters. For some parameters, it is theoretically meaningful to impose the restriction of independence of certain experimental factors on them. For example, the mean interarrival time $\frac{1}{\lambda}$ should neither depend on target presence nor on set size, because it is supposed to characterize the early visual processing stage which operates on lower perception processing level and not easy to influence deliberately. It is plausible that $\frac{1}{\lambda}$ can be influenced by certain aspects of the stimuli, such as the contrast or the number of feature dimensions. Although the amount of information to process in total increases with set size, this should not affect $\frac{1}{\lambda}$ because it describes the time required to complete the processing of the features of a stimulus. Similarly, there is hardly a theoretical reason why the residual times should depend on set size. From another perspective, certain experimental factors should not influence any of the parameters directly. Target presence is one of such factors. The observer is not aware of the actual status of a trial regarding the target presence until a decision is made based on the search experience on a trial, which depends on these parameters[9]. By assuming a dependence of the residual time on the response type, T_{resn} and T_{resy} should correlate with target presence. But this

[9]There may be inter-trial variability due to adjustment of the parameters depending on the search experience on the previous trial. However, this is not a subject of focus in the current project and does no belong to the phenomena the model aiming to explain.

cannot be interpreted as a direct influence of target presence on T_{resn} and T_{resy}. Apart from these situations where an independence is obvious, a dependence of a parameter on an experimental factor should be considered. Specifying condition-dependent parameters means relaxing the constraint on the parameter and allowing an individual parameter for each level of the factor. This increases the total number of parameters that need to be estimated from the entire data set. For example, for four set size levels, if $\frac{1}{\mu}$ is specified as set size dependent, there will be seven parameters to estimate instead of four.

Among the three known factors described in Section 6.2.3.2 that contribute to the difficulty of estimating the parameters stably and accurately, especially for $\frac{1}{\lambda}$, the first one is a property of the model that matches the theoretical notion of early parallel processing stage and the second is given by the nature of the modeling problem. They cannot and should not be changed. The third factor, however, is at least partially a technical problem and can be dealt with using proper techniques.

As mentioned above, once the (approximate) true values of T_{resn} and T_{resy} are given to the model as a fixed argument, the identifiability issues diminish and the optimization process behaves desirably. Therefore, it appears to be a good strategy to first find out a good approximation of T_{resn} and T_{resy} and then fix them on these values while estimating $\frac{1}{\lambda}$. In other words, the logic is to estimate T_{resn} and T_{resy} as accurately as possible in order to approach a reliable estimate of $\frac{1}{\lambda}$.

Things start becoming circular when one scrutinizes the question of how to obtain a good approximation of T_{resn} and T_{resy}. Without additional source of information, the best thing one could do in this situation is to estimate them based on the model, keeping at least one of the other parameters fixed. Otherwise, the model will be overparameterized and the optimization unlikely to result in good estimates of T_{resn} and T_{resy}. Since $\frac{1}{\mu}$ turns out to be the most robust parameter, choosing $\frac{1}{\mu}$ to fix is natural.

However, fixing $\frac{1}{\mu}$ alone is not always sufficient to overcome the overparameterization. Due to the dependence between $\frac{1}{\lambda}$ and T_{resn} and T_{resy}, the dependence of the estimates on the starting value still exists, even after $\frac{1}{\lambda}$ is fixed. Therefore, both $\frac{1}{\lambda}$ and $\frac{1}{\mu}$ should be fixed when estimating T_{resn} and T_{resy}. Because the quality

of the resulting estimation depends on the quality of the fixed arguments, a single round is probably subject to biases. Hence, iterating the estimation of (T_{resn}, T_{resy}) and $(\frac{1}{\lambda}, \frac{1}{\mu})$ alternately while holding the other pair fixed is a better strategy.

As mentioned above, no reparameterizaion that improves the optimization process was found. To cope with the identifiability issues of $\frac{1}{\lambda}$, the technique of profiling was applied instead. The profiling technique has been applied to estimating the parameter $\tilde{\beta}$ in the accuracy model (see Section 6.1.2).

In summary, several identifiability issues were observed and diagnosed. Numerical techniques were applied as countermeasures to diminish these issues. However, the application of these techniques has limitations. First, the identifiability issues are partially conditioned by the nature of the model and the nature of the modeling problem. Second, the sample size of the sub-data sets is apparently not sufficient for such a complex model. When applied to data with a small sample size, a precise estimation technique may even lead to misleading results in this case due to overfitting. Nevertheless, in the recovery studies, overfitting was observed only when fitting sub-data sets separately. If we consider the sub-data sets as different information sources which each reflect mainly an incomplete or one-sided aspect of the underlying mechanism, combining the information extracted from these sources may provide a more comprehensive and accurate picture of the underlying mechanism. The observation that the parameter estimates from sub-data sets of the same experiment did not fall apart from each other far away is also in line with this assumption. This can be the key to resolving the identifiability issues if an effective procedure is applied. In other words, all sub-data sets from an experiment should be fit conjointly with the restriction that all the sub-data sets share the same parameter values. Ideally, each sub-data set makes a unique, complementary contribution under this constraint. In next section, a hierarchical fitting procedure is proposed to achieve this goal.

6.2.4 A hierarchical parameter estimation procedure unifying information across experimental conditions

The goal of the procedure is to combine information from each sub-data sets collected under different experimental conditions to form an overall picture of the parameter values underlying the data in the same experiment.

In light of the identifiability issues and numerical difficulties observed when fitting individual sub-data sets, simply taking all sub-data sets together and fitting the model to them all at once is problematic. Although the parameters are constrained to be invariant across sub-data sets, the complexity of the objective function increases due to different model configurations for different sub-data sets. In this case, a change in a parameter could induce changes in the objective function in more different ways, depending on the configuration of other parameters. The ill-conditioning problem will become more severe and more difficult to deal with. Consequently, more numerical difficulties are to expect, which increases the difficulty of resolving the identifiability issues.

Therefore, a procedure with a hierarchical structure is designed to reduce the complexity. A flow chart of the procedure is illustrated in Figure 6.1. The procedure is composed of two layers. The function of the first layer is to obtain local heuristic values which are sufficiently good parameter estimates for individual sub-data sets. In this layer, the model is fitted to each sub-data sets separately without any global constraints. The function of the second layer is then to integrate these individual heuristic values and utilize them as input to find parameter values that can explain all sub-data sets globally. In this layer, the model is fitted to all sub-data sets simultaneously with the constraint that the parameters must be invariant to the experimental conditions. To take all sub-data sets into account, the objective function is defined as the weighted sum of the distances of all sub-data sets (details explained below). In order to dealing with numerical difficulties, either layer is further composed of two steps. In every step, the techniques that suit the purpose and the characteristics of the step are selected and implemented to improve the quality of its (intermediate) results. Throughout the procedure, information from different sub-data sets are integrated and refined step by step such that inferences from each individual sub-data set are adapted in light of the information gained from other sub-data

sets gradually. The parameter estimation "evolves" at each step to become better in accounting for all sub-data sets globally. In the following, each step will be explained in details.

The first layer is characterized by necessary use of arbitrary initial values, small size of the data and low complexity due to high homogeneity of the data. Therefore, all the numerical difficulties and identifiability issues described in the last section are expected to occur but to a smaller degree. All the numerical techniques described in the last section are directly applicable except for one. Estimating a subset of the parameters while holding the complementary set fixed requires approximately accurate values for the fixed parameters to work effectively. Otherwise, the results can be biased severely. At the very beginning, no prior information about the optimal value of any parameter is available such that the range of a good value to fix on can be narrowed down. This means that a step from arbitrary starting values to a set of preliminary estimates that is close to the optimal values is required. A natural idea is first estimating all parameters from the sub- data sets with arbitrary initial values to obtain values to fix on and then iterating the estimation with the results from the last iteration as the value to fix on or new initial values. I followed this idea but improved its precision by a modification. If we estimate all of the four parameters $(\frac{1}{\lambda}, \frac{1}{\mu}, T_{resn}, T_{resy})$ with arbitrary initial values at the beginning, the results is likely to suffer from overparameterization, as discussed in the last section. Because the parameter estimation at the first step is based on each of the sub-data sets separately, one way to reduce the number of parameters is to include only the RTs of correct responses. Excluding the RTs of incorrect responses means that the data remaining in each sub-data set are all associated to only one of the response options. Then the parameters to estimate are reduced to $(\frac{1}{\lambda}, \frac{1}{\mu}, T_{resn})$ or $(\frac{1}{\lambda}, \frac{1}{\mu}, T_{resy})$, depending on the condition under which the sub-data set was collected. This is a trade-off between coping with overparameterization and making full use of the data. Given the small sample size of a sub-data set, the quality of the information carried by RTs of incorrect responses in each sub-data set is more limited due to their low frequency. Because the estimation of T_{resy} for target-absent data sets as well as that of T_{resn} for target-present data sets rely completely on the incorrect RT in these data sets, their estimates cannot be very accurate, even if there was

no numerical difficulties. It appears to be a good strategy at the very beginning to restrict the information extracted from the data to the most reliable to achieve the best ratio of information utility.

Adopting this expedient strategy in the first step of the first layer (step 1) does not mean forgoing the use of incorrect RTs. The second step of the first layer (step 2) takes RTs of incorrect responses into account. Because the "no" responses are the majority on target-absent trials and so as "yes" responses on target-present trials, T_{resn} estimated from (the correct RTs of) the target-absent data and T_{resy} estimated from (the correct RTs of) the target-present data are likely to be accurate and reliable. In other words, the requirement for the technique of fixing a subset of the parameters is met after step 1 since it is able to provide good values for T_{resn} and T_{resy} to fix on. More specifically, in step 2, RTs of both correct and incorrect responses are included in the analysis and either T_{resn} or T_{resy} is held fixed during the estimation of the other three parameters. For each sub-data set under target-absent condition, T_{resy} is fixed on its estimate $\widehat{T_{resy}}$ in step 1 based on the target-present sub-data set with the same set size level. For each sub-data set under target-present condition, T_{resn} is fixed on its estimate $\widehat{T_{resn}}$ in step 1 based on the target-absent sub-data set with the same set size level. The starting values of the other three parameters in step 2 take the values of the estimates in step 1 based on the same sub-data set. In this way, the problem of overparameterization is mitigated. Although a sub set of the parameters is fixed, this process is not iterated because an iteration yields a bad cost-benefit ratio. Iterating this process is computational very costly due to the application of profiling (described below) and it does not appear to change the results to a large extend. Since the objective of layer 1 is to obtain satisfactory heuristic values of the parameters, a high precision in this step is desirable but not required. Thus the high computational expense is not justified. Moreover, the value to which the parameter is fixed results completely determined by another sub-data set. Since the information integration across sub-data sets will occur more effectively in layer 2, an iteration here is not necessary.

To cope with the identifiability issues regarding $\frac{1}{\lambda}$, profiling on $\frac{1}{\lambda}$ is applied throughout the two steps in the first layer. In both steps, the parameter domain of $\frac{1}{\lambda}$ is divided into ten disjoint, adjacent intervals of equal length. An optimization

process is run for each of these intervals with the endpoints of the interval as lower and upper bounds and its midpoint as the starting value of $\frac{1}{\lambda}$. The settings regarding the other parameters remain the same across these ten optimization runs. The estimate of $\frac{1}{\lambda}$ in step 1 obtained from the optimization process with restriction on an interval is passed to the optimization process in step 2 with restriction on the same interval as the new initial value.

The second layer is characterized by use of heuristic anchors, large size of the data and high complexity due to high heterogeneity of the data. Therefore, the use of techniques and strategies focuses on integrating different information sources and reducing complexity. In this layer, all data points are included regardless of the condition and correctness of the response. In order to take all sub-data sets into account simultaneously, the objective function is defined as a weighted sum of the distances of all sub-data sets. The weight of each sub-data set is determined by "the best individual fit ever obtained" (details described below). Following the countermeasure to overparameterization described in the last section, an iterative procedure was adopted in which $(\frac{1}{\lambda}, \frac{1}{\mu})$ and (T_{resn}, T_{resy}) are fixed in turns. Profiling is not applied to the estimation in layer 2 for three reasons. First, a value considered as good starting value or value to fix on for $\frac{1}{\lambda}$ is determined from the results of layer 1. By surviving the selection, this value is supposed to indicate the area of the optimal value. Second, fixing $(\frac{1}{\lambda}, \frac{1}{\mu})$ and (T_{resn}, T_{resy}) alternately copes with the problem of collinearity between $\frac{1}{\lambda}$ and (T_{resn}, T_{resy}). This is not given in layer 1 so that profiling is necessary there. Third, profiling within an iterative procedure is computationally costly.

Before coming into the second layer, some preparatory work is required. The intermediate results of layer 1 need to be selected and integrated. A baseline of the fit for each sub-data set also needs to be determined for the selection and for constructing the objective function in layer 2. After step 2, we obtain for each sub-data set ten groups of estimates resulting from the optimization processes on each of the ten intervals regarding $\frac{1}{\lambda}$. According to the principle of the profiling technique, the group yielding the best fit among the ten groups should be selected. However, simply selecting the group with the smallest distance for the corresponding sub-data set does not achieve the goal of incorporating sub-data sets under different conditions. A group may result in the smallest distance

because of some data-specific aspects (overfitting), which makes it locally optimal from the perspective of the entire data set. The groups yielding the smallest distance for different sub-data set are not necessarily associated with the same interval. If results associated with different intervals are selected, integrating them becomes problematic because taking the average is not justified[10]. Because the ultimate goal of the estimation procedure is to find the optimal parameters that account for all the sub-data sets at the same time, the information from different sub-data sets must be integrated in a way such that the heuristic values for the next step are good for all sub-data sets at the same time. That is, the selection should take all sub-data sets into consideration simultaneously. Therefore, the selection criterion should be the best overall fit across data sets instead of the best individual fit of single data sets.

For this purpose, an appropriate definition of the overall fit is required. It is also required for layer 2 where two or more data sets are fitted to together. Recall that a similar problem occurred when constructing the objective function accounting for correct and incorrect RTs simultaneously (see Section 4.6). Simply taking the sum of the distances of the sub-data sets is problematic here as well, since the best possible fit for a sub-data set appears to depend on the experimental condition. Target presence is observed to have a significant influence of the best possible fit regardless of set size. Target-absent sub-data sets have worse fits than target-present sub-data sets in general (i.e., larger distance). This influence seems to be intrinsic because it was observed for both simulated and empirical data. Although the distance is generally larger for empirical data, the difference between the distances of target-absent and target-present data is similar as for simulated data. Furthermore, the fit of both simulated and empirical data decreases (i.e., the distance increases) for increasing set size regardless of target presence. However, the increase in the distance is much larger for empirical data. That is, even though the simulated data sets with all set size levels can be fitted to

[10]Suppose that the locally optimal group of estimates are associated with the interval at the left end for one sub-data set and the with the interval at the right end for another sub-data set. The average will be located in the middle. But taking the average in this case completely loses the information of the local optimality of both groups because no inference on the performance of the averaged value is possible. In the worst case, the values in the middle area yield the worst fit within the domain for both sub-data sets. This approach is not helpful. Thus, choosing individually optimal estimates is inconsistent with the goal of layer 1.

quite well, this was only possible for empirical data sets with small set size levels. Apparently, there are some mechanism playing a role for visual search among large number of stimuli that the model is not able to account for (a detailed discussion on this issue see Section 8). Regardless of the theoretical explanation, it appears to be a general phenomenon that the extent to which a sub-data set can be explained by the model is quite different. In other words, when a sub-data set is fitted to individually, the smallest possible distance can be on a scale of ten times as for another sub-data set. This means that they bring different "qualifications" in the optimization process when fitted to together. If we simply take the sum of the distances arising from each data set as the objective function, the data sets with larger distance will dominate the optimization process and the estimation may be biased in favor of these data sets. Therefore, these distances should be weighted such that they are expressed on a common scale before being added up.

A magnitude considered as appropriate weight that fulfills this purpose is the reciprocal of the minimum distance among the ten optimization processes associated to a sub-data set step 2. For each sub-data set, the minimum distance can be seen as a reference value representing the best fit that has ever been achieved for this sub-data set by individual model-fitting without constraints related to other sub-data sets. Weighting the distances associated with the other intervals by its reciprocal is equal to dividing these distances by the minimum distance itself. The values resulting from the weighting are in fact ratios equal or greater than one which can be interpreted as "how much worse is the fit compared to the best case." The larger the value, the worse the relative fit. In this way, both the selection of the optimal estimates in layer 1 and the objective function in layer 2 takes the diverse "qualifications" of individual sub-data sets into consideration.

The optimal heuristic anchor for layer 2 is then determined in the following way. First, the ten distances associated with the same sub-data set are divided by the minimum among them. Then the weighted distances associated with the same interval are added up across all sub-data sets. The estimates associated with the interval with the least weighted sum are selected. Now for each sub-data set, one group of estimates is selected. To integrate them into an optimal heuristic

anchor containing one value for each parameter, the selected estimates for the same parameter are averaged across sub-data sets. Averaging here is justified because these values are performing well for all sub-data sets according to the selection criterion.

In step 3, all the sub-data sets are fitted jointly with the constraint that the parameters must be the same across data sets. As there are two subsets of parameters to be fixed on alternately, there are two possibilities for the order. Among the heuristic values obtained from step 2, $\frac{1}{\mu}$ should be most accurate because it is most robust. Since $\frac{1}{\lambda}$ has relatively small influence and it is constrained to an interval, its influence will not be too large even if the value fixed on deviates from the true value. Thus, it is a better strategy to fix $(\frac{1}{\lambda}, \frac{1}{\mu})$ first. It is more likely to obtain accurate estimates of T_{resn} and T_{resy} in this way. The starting values of T_{resn} and T_{resy} in this step are then the average of estimates from target-absent and target-present data in step 2.

After obtaining estimates of T_{resn} and T_{resy} from fitting all sub-data sets together, these estimates can now serve as input in step 4. (T_{resn}, T_{resy}) exchanges the role with $(\frac{1}{\lambda}, \frac{1}{\mu})$. They are then fixed to the estimated values from step 3 and the objective function turns to a function of $(\frac{1}{\lambda}, \frac{1}{\mu})$, in regard to which the function is minimized. Their starting values take the values to which they are fixed in step 3.

The estimates resulting from step 4 are passed back to step 3 of the next iteration as new inputs. Step 3 and 4 iterate alternately until the estimates stabilize or the maximum number of iterations is reached. The stabilization criterion is defined as an absolute difference between the old and new value within 0.5 for all parameters. 0.5 is chosen because a precision on the level of 1 millisecond is not meaningfully interpretable. The maximum number of iterations is set at 30.

As mentioned in Section 3.3.4, the parameter c (number of servers) should be a natural number restricted to ten. Thus the approach of profiling is applied to estimate c. In other words, all other parameters are estimated assuming a fixed c and the estimation has to be done for each c in the parameter domain. Hence, the procedure described here has to be repeated for $c = 1, \ldots, 10$.

The `R` code that implements this procedure is given in Appendix I.

The final estimates are then determined by comparing the distances across

different c. The criterion is the unweighted sum of distances of target-absent and target-present sub-data sets. The weighted sum is not appropriate as criterion when comparing fits across data sets because the weights are determined by the individual minimum distance in step 2, which are not shared by other sub-data sets. The unweighted sum, however, is not perfect. Although it works well for simulated data such that the true value of c is always selected, this is not necessarily the case for empirical data because of additional noise or unexplained mechanisms. Due to the larger heterogeneity among the empirical sub-data sets, there is certain ambiguity when using the unweighted sum as the selection criterion. This issue will be discussed in detail in Section 7.3.

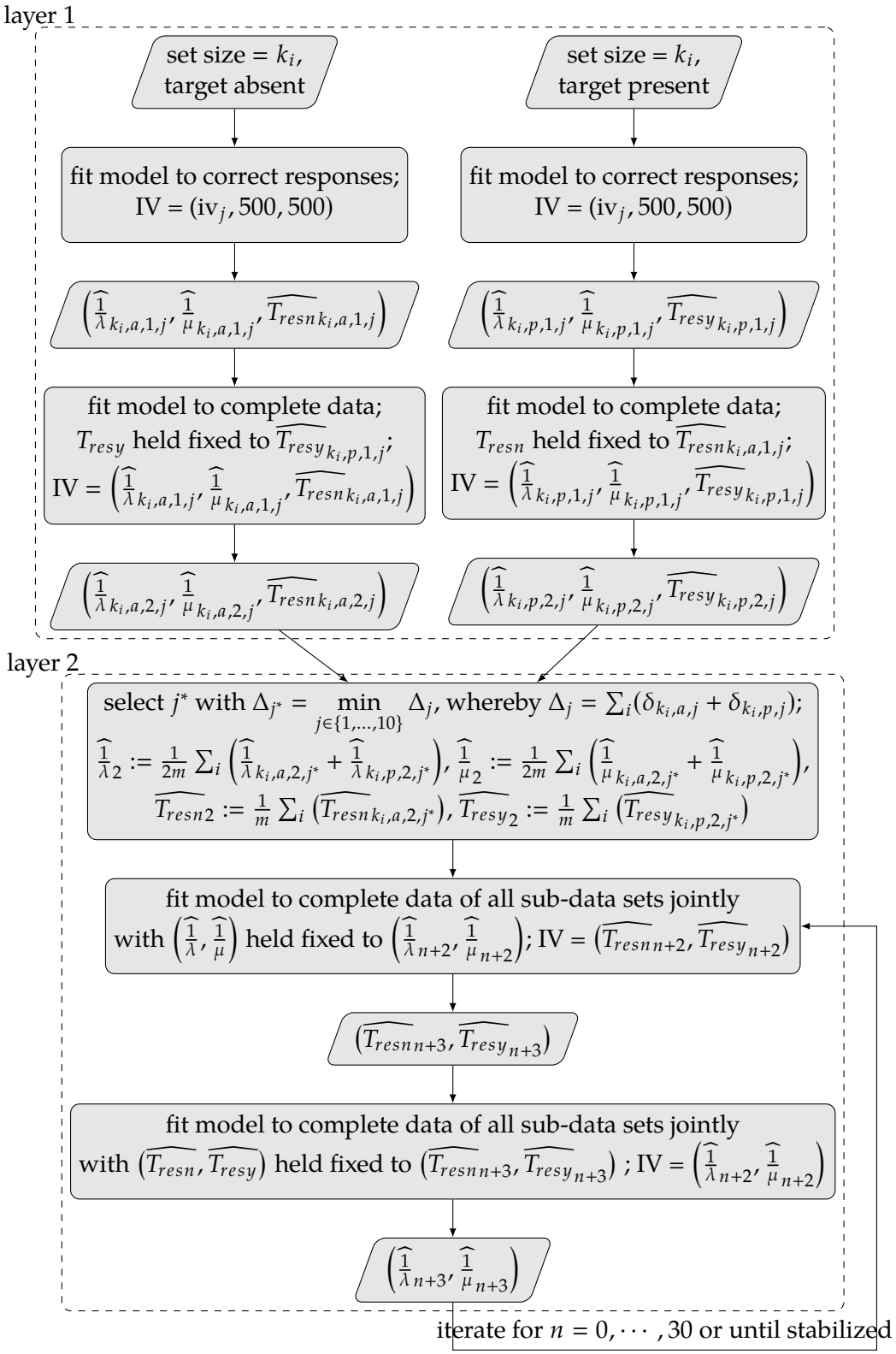

Figure 6.1: An illustration of the hierarchical parameter estimation procedure as a flow chart.

Chapter 7

Model fitting

Testing a model with empirical data is an essential step in the process of scientific method. The goal of this chapter is to investigate the ability of the queueing model of visual search to explain behavioral data observed from human observers.

A body of data collected and published by Wolfe and colleagues (E. M. Palmer et al., 2011; Wolfe et al., 2010) is used to test the model for three reasons. First, their study contain a large number of trials from each subject doing each task type. As discussed in Section 6.2, statistical inference on the RT-related parameters of the queueing model of visual search is only feasible when the model is fitted to the entire RT distribution. To obtain reliable parameter estimates, data sets collected under different conditions (set size $\times$ target presence) ought to be fitted first individually and then together. This means that a large sample size in each cell of the design is required to obtain meaningful characterization of the RT distributions. For each task type, Wolfe and colleagues collected approximately 4000 trials from each of the nine or ten subjects, so that in each set size $\times$ target presence condition, a total of about 4500 to 5000 trials are available. To my knowledge, this is the largest visual search data published in the form of raw data. Second, the design and conduction of their experiments did not involve extraneous elements. Set size and target presence are the only independent variables manipulated and simple figures are used as stimuli. Third, the three types of visual search tasks conducted in their study are considered representative for their degree of difficulty. Feature search, conjunction search and spatial configuration search (will be described below) have been of theoretical interest

in the visual search literature and important for the study of mean RT of visual search. They are typically perceived by human observers as differently difficult (easy, medium and difficult, respectively) and the associated RT × set size slopes usually fall in different ranges. As the primary goal of this subproject is to examine the validity of the model in explaining behavior in a typical visual search situation, we consider this data as appropriate for model validation and expect informative inferences from the results.

7.1 Brief description of the methods of Wolfe et al. (2010)

To facilitate the understanding of the subsequent analysis and discussions in the current context, the most relevant aspects of the study in which the data was collected are summarized here. A detailed, comprehensive description of the stimuli and methods can be found in Wolfe et al. (2010) and the website on which the data is published.

The data was collected from twenty-eight participants aged 18 to 55 with normal or corrected-to-normal visual acuity and color perception. Three types of search tasks were conducted. In the feature search task (nine participants), participants searched for a red vertical rectangle among green vertical rectangles of the same size and form. In the conjunction search task (ten participants), the same red vertical rectangle was defined as the target as well but the display included red horizontal rectangles as additional distractors. In the spatial configuration search (nine participants), participants looked for a white digital "2" among white digital "5"s. The target and the distractors possess exactly the same features (both consist of three horizontal and two vertical lines) but different configuration due to the different positions of the vertical lines. Stimuli in these three search tasks were all presented on a black background of a squared-shaped display area, which was divided into a 5×5 invisible lattice pattern with cells of the same size. In a search display, each cell contained one or no stimulus, which was placed at a random location within the cell. The number of stimuli present in the cells added up to the set size chosen for the current trial. An example of the search display is illustrated in Figure 2.1 in Section 2.1.3, Chapter 2.

For each task type, the experiment had a factorial within-subject design of two factors: set size (3, 6, 12 or 18) and target presence (absent or present). On each trial, both the set size of the display and the presence of the target were chosen randomly, with a probability of 0.5 for either of target presence and absence and a probability of 0.25 for each of the four possible set sizes. As a consequence, some of the cells of the design had slightly more trials than others. Following 30 practice trials at the beginning of each block, each participant completed 12 blocks of 300 and one block of 400 experimental trials, resulting in a total of 4000 experimental trials[1]. Both factors were intermixed in each block.

Throughout the experiment, a white fixation cross was present in the center of the screen, on which participants were instructed to keep their eyes focused. The beginning of each trial was indicated by an auditory signal. After 500 ms, the search display appeared and remained visible and unchanged until the participant responded by pressing either of the keys indicating target-present or target-absent. Participants were instructed to respond as quickly and accurately as possible. After the response, a feedback was shown for 500 ms, informing the participant whether the response was correct.

7.2 Results

The model was fitted to the conjunction search data and the spatial configuration search data. The feature search data was forgone because no recognition stage is considered required in this task type (see Section 2.2.1.1). That is, in the notion of the queueing model, the completion of the feature search task does not require a service stage. Thus the model is not applicable to describing the decision mechanism in feature search.

For the analysis, trials with RT < 200 ms were excluded because it is not plausible that such short RTs reflect a decision made based on the search experience. This criterion was also used by E. M. Palmer et al. (2011) and Wolfe et al. (2010). In their analysis, trials with long RTs were excluded (RT > 4000 ms

[1]The concrete number of data points collected from a participant may deviate from 4000 because it was possible to skip a trial by pressing the space bar. Practice trials were discarded from the data sets.

for conjunction search and RT > 8000 ms for spatial configuration search) as well. This is not done in our analysis because the queueing model predicts RTs without an upper limit and the participants were not forced to response within a certain amount of time. A further difference to E. M. Palmer et al. (2011) and Wolfe et al. (2010) is that the error trials were included in our analysis. The model was fitted to aggregated data across all participants to ensure a large sample size.

The estimation of accuracy-related parameters using MLE converged successfully for both conjunction search and spatial configuration search data. The application of the profiling technique resulted in a maximum unambiguously for both data sets. Table 7.1 shows the ML estimates of the accuracy-related parameters for both task types. For conjunction search task, it yielded a log-likelihood of -4274.83 and for spatial configuration task -4431.19. The error rates predicted by the model were calculated based on these ML estimates according to Equations (4.15) and 4.16. Figure 7.1 shows a comparison of the predicted and the observed error rates for the conjunction search task. Figure 7.2 shows a comparison of the predicted and the observed error rates for the spatial configuration search task.

Table 7.1: ML estimates of accuracy-related parameters

	$\hat{\alpha}$	$\hat{\beta}$	$\hat{a_1}$	$\hat{a_2}$	$\hat{b}$
Conjunction search	−4.013132	.053088	.026247	.000010	1.280048
Spatial configuration	−4.414780	.114495	.0181728	.000012	1.108340

Note. The numbers are rounded to the sixth decimal since the predictions are very sensitive to rounding (especially for $\hat{a_1}$ and $\hat{a_2}$).

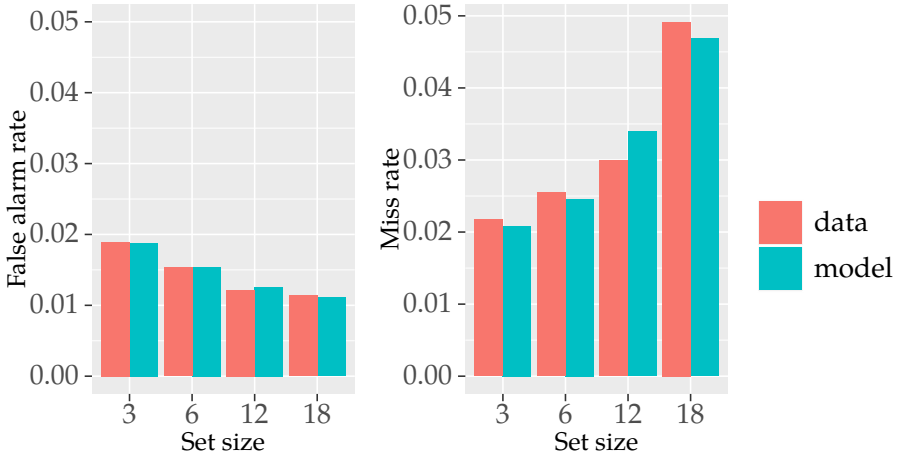

Figure 7.1: Comparison of the error rates in the conjunction search data of Wolfe et al. (2010) and the model predictions.

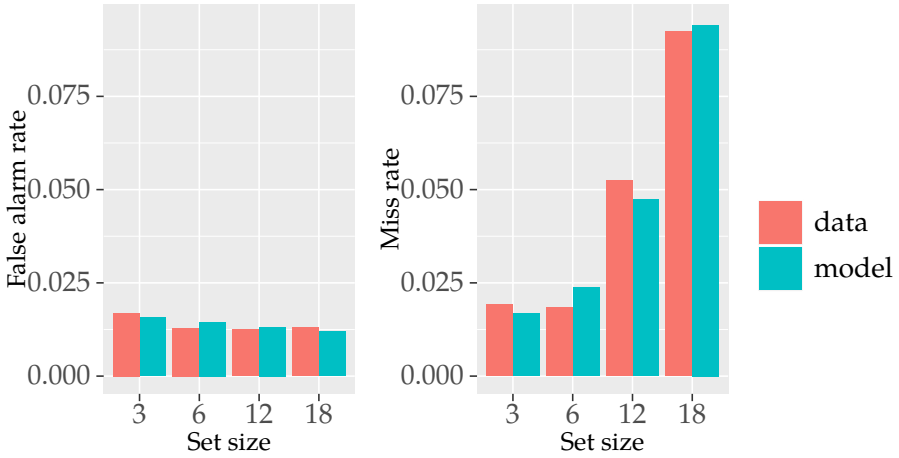

Figure 7.2: Comparison of the error rates in the spatial configuration search data of Wolfe et al. (2010) and the model predictions.

The estimation of RT-related parameters was successful despite some convergence issues occurred occasionally in layer 1 of the estimation procedure. The iteration of layer 2 converged successfully for both task types.

For the conjunction search data, it yielded a global good model fit in general even with the most strict constraint that all five parameters remain invariant across all experimental conditions. Although relaxing this constraint for the

parameter $\frac{1}{\mu}$ (allowing dependence on set size) did reduce the total distance to some extent, the estimates of $\frac{1}{\mu}$ for different set sizes did not fall far away from each other in general (absolute difference within 72), especially for those cs with better fits (absolute difference within 53). Hence the more parsimonious model with the same parameters across sub-data sets is preferred. The results presented here refer to this model. As c increases, the distance sum first decreases then increases for both target-absent and target-present sub-data sets. The best overall fit was observed for $c = 5$, consisting of a sum of the distances of 406.53 for target-absent sub-data sets and 143.90 for target-present sub-data sets. For both target-absent and target-present sub-data sets, the sum was the minimum among $c = 1, \ldots, 10$. The distances of each sub-data sets depending on c are presented in Table 7.2.

The MDE estimates of the RT-related parameters depending on c are displayed in Table 7.3. In Figure 7.3 and 7.4, the histograms of the RTs predicted by the model for $c = 5$ are compared with the histograms of the empirical RTs of conjunction search, separated by target presence, set size level and correctness of the response.

Table 7.2: Distances for the conjunction search data

	target absent				target present				
set size	3	6	12	18	sum	3	6	12	18	sum
$c = 1$	69	143	521	900	1633	57	103	302	531	994
$c = 2$	52	46	150	272	519	35	31	69	139	274
$c = 3$	38	45	127	228	437	34	26	36	59	154
$c = 4$	40	37	113	222	412	34	26	32	54	146
$c = 5$	49	39	102	217	407	39	28	30	47	144
$c = 6$	54	43	95	214	407	44	30	35	41	149
$c = 7$	66	55	80	207	408	58	36	30	37	162
$c = 8$	71	61	74	207	413	67	42	31	46	186
$c = 9$	73	64	71	211	419	70	44	34	58	206
$c = 10$	74	66	70	216	425	72	46	36	68	221

Note. The numbers have been rounded to the nearest integer to enable a lucid overview.

Table 7.3: MDE estimates of RT-related parameters for the conjunction search data

	$\widehat{\frac{1}{\lambda}}$	$\widehat{\frac{1}{\mu}}$	$\widehat{T_{resn}}$	$\widehat{T_{resy}}$
$c = 1$	100	100	129	234
$c = 2$	100	100	200	294
$c = 3$	86	104	237	326
$c = 4$	53	132	241	334
$c = 5$	23	151	245	353
$c = 6$	10	163	244	363
$c = 7$	10	181	213	363
$c = 8$	10	189	203	365
$c = 9$	10	193	199	365
$c = 10$	10	195	197	365

Note. The numbers have been rounded to the nearest integer to enable a lucid overview.

For the spatial configuration search data, it did not yield a globally good fit with the most strict constraint that all five parameters remain invariant across all experimental conditions. In general, good fits were only observed locally for some of the sub-data sets and they always came along with obvious misfits for other sub-data sets. In particular the sub-data sets with smallest set size and those with the largest could not be fitted well at the same time. The factor set size appeared to play an important role. For small c (1-3), sub-data sets with small set size levels (3 and 6) could be fitted well, but sub-data sets with large set size levels (12 and 18) could not. For larger cs (4-6), the reversed pattern was observed. For the largest cs, only the sub-data sets with set size 6 could be fitted well.

In light of these observations, the constraint on the parameter $\frac{1}{\mu}$ was relaxed such that it was allowed to vary depending on the set size level. After this relaxation, a good global fit was achieved. The distances of each sub-data sets depending on c are presented in 7.4. The selection of the optimal c, however, is ambiguous. As c increases, the distance sum for target-present sub-data sets first decreases and then increases, but the distance sum for target-absent sub-data sets decreases monotonically. The least total sum (588.49) was observed for $c = 7$,

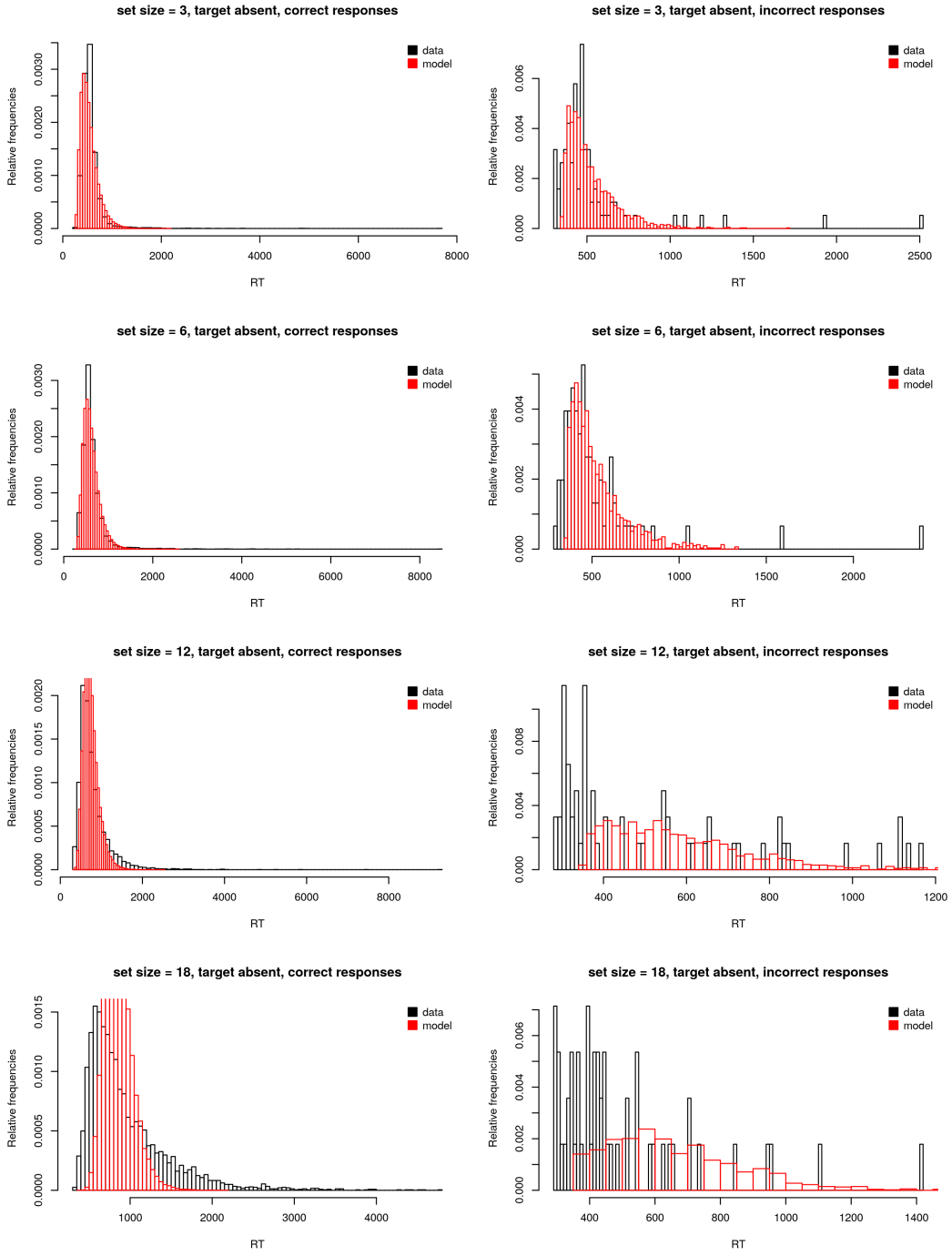

Figure 7.3: Comparison of the RT distribution on target-absent trials in the conjunction search data of Wolfe et al. (2010) and the model predictions for $c = 5$.

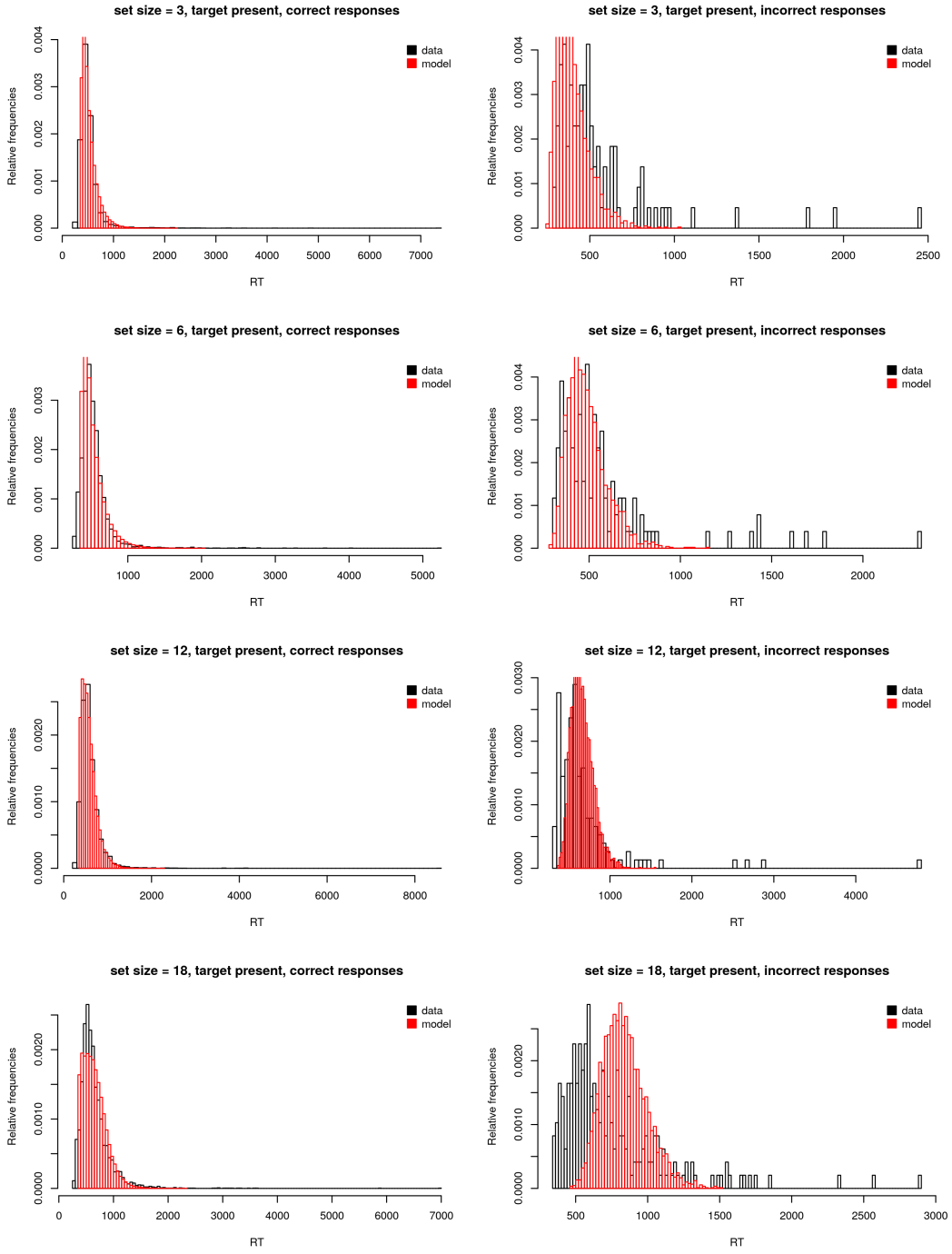

Figure 7.4: Comparison of the RT distribution on target-present trials in the conjunction search data of Wolfe et al. (2010) and the model predictions for $c = 5$.

consisting of a sum of the distances of 401.17 for target-absent sub-data sets and 187.32 for target-present sub-data sets. However, the total sum for $c = 6$ (595.02) and $c = 8$ (589.94) were very close. As will be discussed in the next section, I consider $c = 5$ the optimal value based on several observations, arguing that the reduction of the total sum for $c = 6, 7, 8$ are likely to be achieved by overfitting the target-absent sub-data sets with set size of 12 and 18. It yielded a sum of the distances of 453.50 for target-absent sub-data sets and 157.95 for target-present sub-data sets (611.45 in total).

The MDE estimates of the RT-related parameters depending on c are displayed in Table 7.5. In Figure 7.5 and 7.6, the histograms of the RTs predicted by the model for $c = 5$ are compared with the histograms of the empirical RTs of conjunction search, separated by target presence, set size level and correctness of the response.

Table 7.4: Distances for the spatial configuration search data

	target absent				target present				
set size	3	6	12	18	sum	3	6	12	18	sum
$c = 1$	19	90	187	314	609	32	40	67	105	244
$c = 2$	16	70	177	323	585	27	27	61	102	217
$c = 3$	16	46	147	329	537	27	23	47	94	191
$c = 4$	16	39	115	324	494	27	31	27	79	165
$c = 5$	17	37	86	313	453	31	35	30	63	157
$c = 6$	16	37	72	304	430	32	35	50	48	165
$c = 7$	16	37	62	287	401	35	37	66	50	187
$c = 8$	16	38	53	260	366	33	36	82	72	223
$c = 9$	16	37	49	245	347	33	36	92	92	252
$c = 10$	16	38	47	235	336	32	36	96	109	273

Note. The numbers have been rounded to the nearest integer to enable a lucid overview.

Table 7.5: MDE estimates of RT-related parameters for the spatial configuration search data

	$\widehat{\frac{1}{\lambda}}$	$\widehat{\frac{1}{\mu_3}}$	$\widehat{\frac{1}{\mu_6}}$	$\widehat{\frac{1}{\mu_{12}}}$	$\widehat{\frac{1}{\mu_{18}}}$	$\widehat{T_{resn}}$	$\widehat{T_{resy}}$
$c = 1$	56	153	141	123	110	317	358
$c = 2$	65	223	252	238	217	307	374
$c = 3$	70	235	325	339	319	301	386
$c = 4$	78	233	352	422	415	291	386
$c = 5$	81	235	361	493	501	285	388
$c = 6$	69	234	361	529	569	302	403
$c = 7$	53	236	362	552	631	319	419
$c = 8$	59	234	361	569	700	316	414
$c = 9$	59	235	361	578	749	314	412
$c = 10$	62	234	361	582	786	312	410

Note. The numbers have been rounded to the nearest integer to enable a lucid overview.

7.3 Discussion

7.3.1 Accuracy-related findings

The results indicate that model is able to predict the error rates in both task types very well. The essential patterns of both false alarm rate and miss detection rate are captured by the model. The deviation of the model predictions from the empirical data was especially small for false alarm rates. For miss rates, the largest discrepancies were observed for set size of 12 and 18 for conjunction search and set size of 6 and 12 for spatial configuration search. The direction of the deviations, however, did not seem to be systematic. Nevertheless, this does not exclude the possibility that there were other mechanisms playing a role which are not captured by the model because the number of observable deviations is too small.

The estimate of β represents the steepness of the reflected logistic curve. It is interpreted as how polarized the confidence criterion is for different set size levels. Since $\hat{\beta}$ is larger for spatial configuration search than conjunction search,

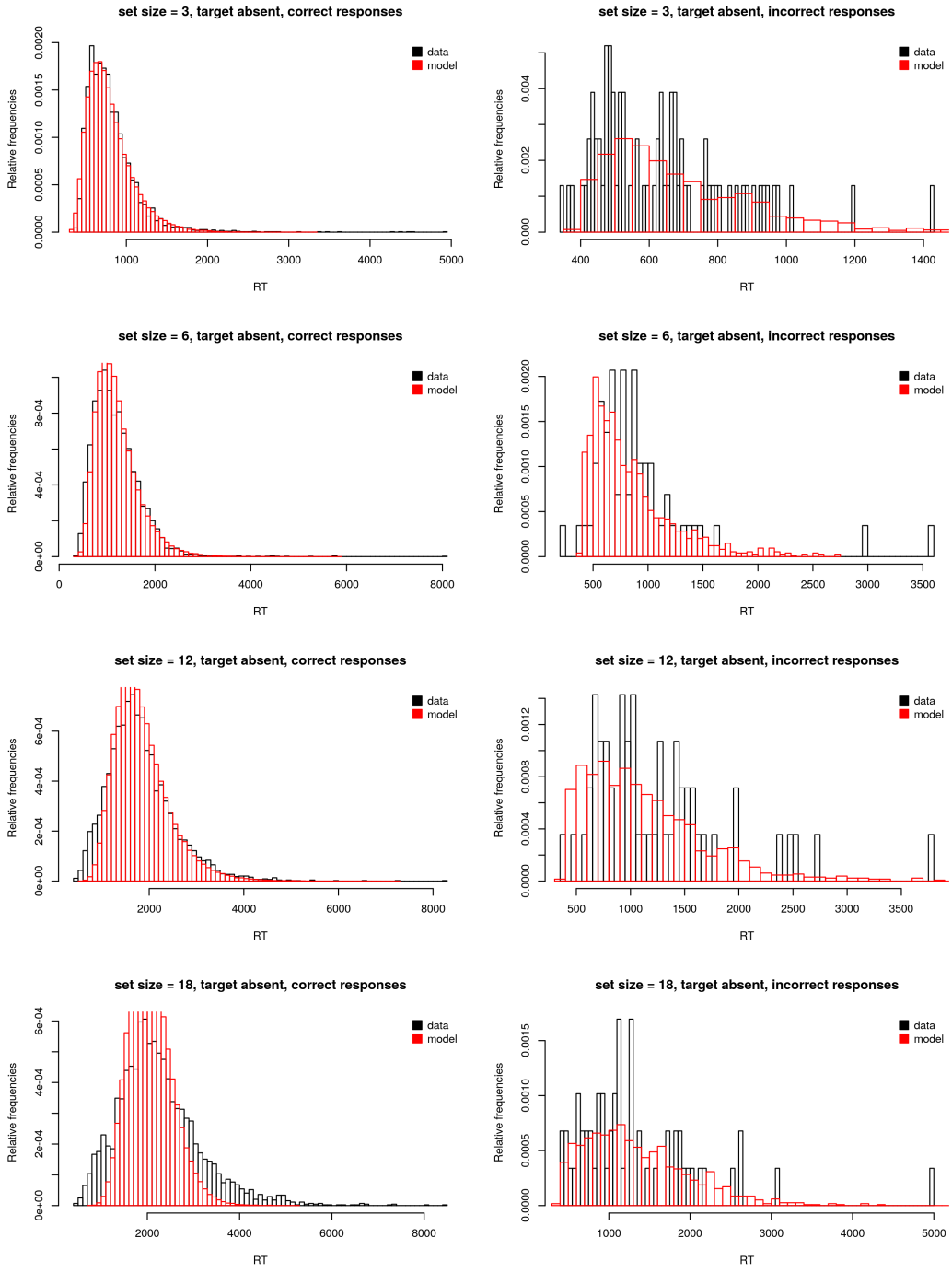

Figure 7.5: Comparison of the RT distribution on target-absent trials in the spacial configuration search data of Wolfe et al. (2010) and the model predictions for $c = 5$.

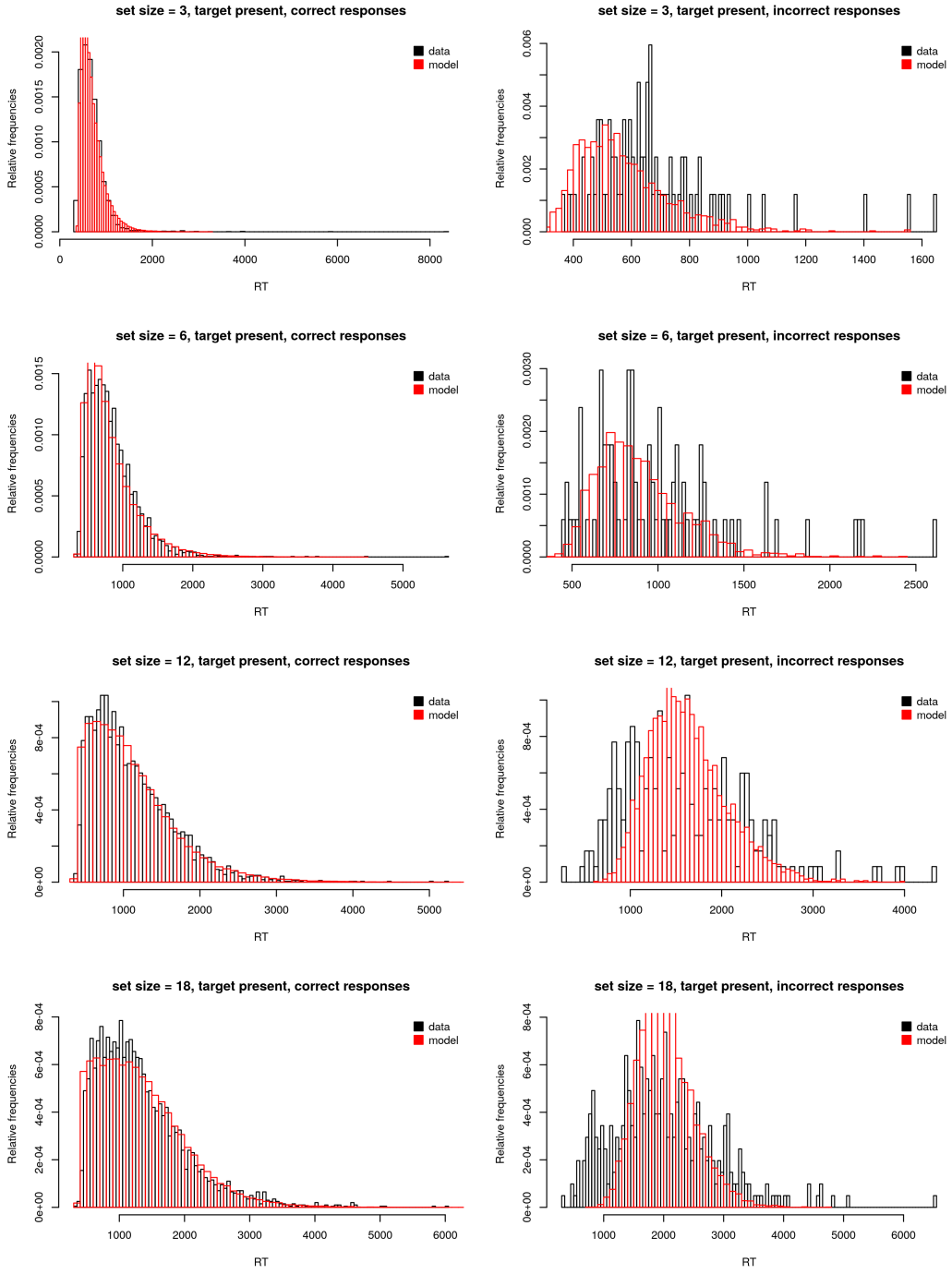

Figure 7.6: Comparison of the RT distribution on target-present trials in the conjunction search data of Wolfe et al. (2010) and the model predictions for $c = 5$.

the reflected logistic curve is steeper, which means that the confidence criterion is more polarized for spatial configuration search. In other words, the participants were more efficiency-oriented when performing the spatial configuration task. They were more willing to perform thorough search (i.e., setting a more strict confidence criterion) to achieve a high accuracy for smaller set sizes and to perform incomplete search (i.e., setting a laxer confidence criterion) to reduce search effort.

This is in line with the other characteristic of the termination criterion reflected in $-\frac{\alpha}{\beta}$, which represents the inflection point of the reflected logistic curve. The inflection point is interpreted as the set size level from which the observer would give up search on equal sized or larger trials and respond always "no" instead. The estimate of this magnitude is about 75 for conjunction search and 39 for spatial configuration task. This means that if the experiment had included larger set size levels and the participants had followed the same strategy, they would have given up search at a much smaller set size level when completing spatial configuration task than conjunction search task. The interpretations of $\hat{\beta}$ and $-\frac{\hat{\alpha}}{\hat{\beta}}$ in these two experiments are consistent with the subjective experience of the participants that spatial configuration task is more difficult.

The estimate of b represents the absolute value of the slope of the log-log plot of p_1 and p_2 on set size. It is interpreted as the extent to which the discrimination threshold is adapted depending on the set size level. A larger b indicates a faster (more sensitive) adaptation of the discrimination threshold to set size. Since $\hat{b}$ was larger for conjunction search task than for spatial configuration task, the probability of misidentifying a distractor for the target decreases faster for conjunction search task. Likewise, the probability of misidentifying the target for a distractor increases faster for conjunction search task.

The estimates of a_1 and a_2 represent the intercepts of of the log-log plot of p_1 and p_2 on set size, respectively. They reveal the position of the discrimination threshold if there was only one object to search ($p_1 = a_1$ and $p_2 = a_2$ for $k = 1$). According to the results, the probability of misidentifying a distractor for the target is lower for spatial configuration task (more strict threshold) than conjunction search. In contrast, the probability of misidentifying the target for a distractor is higher for spatial configuration task. This is in line with the assumption that p_1

and p_2 contrast each other.

7.3.2 RT-related findings

Since no measure of goodness of fit has been developed yet, a general evaluation of the model fit is accomplished qualitatively by comparing the pattern of the distances of individual sub-data sets with the pattern observed for the simulated data in the recovery studies. Observations from fitting the empirical data that deviate from the patterns obtained from the recovery studies indicate discrepancies that cannot be explained by random errors. Recall that two major observations regarding the distance in the recovery studies were: a) holding the set size constant, target-absent sub-data sets had a larger distance than target-present sub-data sets; b) for both target-absent and target-present sub-data sets, the distance increased with increasing set size. These phenomena can be explained by the amount of randomness accumulated during a queueing process as random events take place. The more events (arrivals and departures) take place before the queue stops, the larger variance in the system response time caused by randomness. In target-absent case, the system has to finish the service for more customers than in target-present case. The number of customers the system has to serve increases with set size. This is why the phenomena described are to expect. Furthermore, if the number of servers c has exceeded the set size k, the fit cannot be improved by adding more servers because the additional servers will always remain idle. The fit can remain constant or decrease due to restrictions on the parameters.

For conjunction search, the distance was larger for target-absent sub-data sets than target-present sub-data sets to an ordinary extent. It increased with increasing set size with the exception of $k = 3$. Both target-absent and target-present sub-data sets with set size 3 exhibited a larger distance than expected. A detailed investigation shows that the additional misfit results mainly from a larger (positive) skewness and a larger variance predicted by the model as well as the larger discrepancy for incorrect RTs.

Moreover, the distances of the target-absent sub-data sets with set sizes 12 and 18 as well as the target-present sub-data set with set size 18 were larger than expected. It is apparent from Figure 7.3 that the (positive) skewness and the

variance the model predicted for correct RTs were too small compared to the skewness and variance of the empirical data. There is a considerable proportion of particularly quick correct rejections on the one hand and particularly slow correct rejections on the other hand the model was not able to account for. This means that the model overestimated the efficiency of correct rejections on some trials and underestimated it on some other trials, when set size is large. The opposite was observed for the target-present sub-data set with set size 18. Figure 7.4 shows that the major discrepancy for correct RTs is at the peak of the distribution. The empirical RTs of hits were more concentrated around the mode than the model predicted. This means that the model underestimated the proportion of "typical" efficiency of finding the target when set size is large.

Considerable discrepancies were observed locally for incorrect RTs. Empirical false alarms and misses for large set sizes (12 and 18) were much faster, whereas empirical misses for set size 3 were much slower than the model predicted.

The parameter estimates appear to reflect the characteristics of the conjunction search task. For $c = 5$, the mean interarrival time $\frac{1}{\lambda}$ was estimated at 23.37 ms and the mean service time $\frac{1}{\mu}$ at 151.43 ms. The short mean interarrival time indicates an efficient processing of stimulus features, which were the color (red or green) and the orientation (horizontal or vertical). The short mean service time indicates an efficient integration of stimulus features. The residual time for "no" response was estimated at 245.40 ms and for "yes" response at 352.90 ms. The response "target-absent" had a clear advance of around 100 ms. If this is not due to habitual reasons such as handiness, it probably reflects a stronger readiness for the "no" response. This explanation is consistent with the notion that the observer starts with equal prior probabilities of target absence and target presence and judges according to the posterior probability of target absence based on the search experience. Because of the optimal information foraging strategy, the "no" response is viewed as the default option and revision only takes place when the system identifies a target. Hence, a readiness for the "no" option is reasonable under this condition.

A noticeable observation from 7.3 is that $\frac{1}{\lambda}$ was estimated at the upper bound for $c = 1, 2$ and at the lower bound for $c = 6, 7, 8, 9, 10$. It implies above all that the parameter domain of $\frac{1}{\lambda}$ has restricted the model fit. As the number of servers

increases, the service capacity of the system increases and the old arrival rate may become too low to match it. Thus it is possible to achieve a lower total sum of distances if $\frac{1}{\lambda}$ is allowed to take values smaller than 10. This casts doubts on the optimality of the value 5 for c. After all, the total sum of distances for $c = 5$ had only a slight advantage over that for $c = 6$. This reveals a limitation of the model. Although the model has retain identifiability by fitting all sub-data sets simultaneously in recovery studies, the noise or unexplained variance in empirical data may interfere the identifiability regarding $\frac{1}{\lambda}$ and c. Consequently, the total sum of distances may not be the optimal criterion for comparisons across c.

Before discussing the findings for spatial configuration task, I first explain why I consider $c = 5$ the optimal value before going into details of the estimates. The first reason is related to the implication that the total sum of distances may be not accurate as a criterion for comparing results of empirical data across c. Because the constraint on $\frac{1}{\mu}$ is relaxed such that a dependence on set size is allowed for spatial configuration search data, the extent to which the total sum of distances reflects the overall fit can be further restricted. A dependence of $\frac{1}{\mu}$ on set size allows more space for fitting the individual characteristics of sub-data sets, but at the same time increases the risk of biases caused by local overfitting. The reduction of total sum of distances for $c = 6$ comes from the target-absent sub-data sets with set size 12 and 18 and the target-present sub-data set with set size 18. Similar to conjunction search data, these sub-data sets exhibited a higher increase of distance compared to simulated data, especially the target-absent sub-data set with set size 18. Second, it is unnatural that the target-present sub-data set with set size 12 had a larger distance than the one with set size 18 for $c = 6$. This contradicts the observation for simulated data. Although the phenomenon that sub-data sets with smaller set size have a larger distance than sub-data sets with larger set size has also been observed for the target-present conjunction search sub-data sets with set size 3 and 6, it appears general in that case because it did not depend on c. In the case of spatial configuration search, this did not occur for $c \leq 5$. Apparently, the local fit of the target-present sub-data set with set size 12 was traded off inappropriately for $c \geq 6$. Third, $c = 5$ was the smallest number at which the estimates of $\frac{1}{\mu}$ increased monotonically

with increasing set size. As Table 7.5 shows, $\hat{\frac{1}{\mu}}$ decreases monotonically with increasing set size for $c = 1$. As c increases, this relation turns to first increasing and then decreasing. $\hat{\frac{1}{\mu}}$ increases until set size 6 for $c = 2$ and until set size 12 for $c = 3$ and $c = 4$. This phenomenon was also observed for conjunction search data, when the constraint on $\frac{1}{\mu}$ was relaxed. It indicates a limitation of the fit by c. When c is small, the search efficiency reflected in the empirical data for large set sizes was obviously too high for the given service capacity to explain. With the number of servers fixed to a low value restricting the service capacity, a fit could only be achieved by unusually short service times. In other words, the sub-data sets could not be explained without a trade-off between c and $\frac{1}{\mu}$ for $c \leq 5$. These clues together indicate that $c = 5$ is most likely to be the optimal value.

In general, spatial configuration search data yielded a better fit than conjunction search except for the sub-data sets with set size 18. This is not very surprising given the fact that the constraint on $\frac{1}{\mu}$ was relaxed. The good fit is apparent from the highly matched plots in 7.5 and 7.6. Especially the target-absent sub-data set with set size 3 had an extraordinarily good fit. The individual distances exhibit patterns in accordance with the simulated data.

Similar to conjunction search data, the largest discrepancies regarding correct RTs were also observed for the sub-data sets with set size 12 and 18. There were particularly quick correct rejections the model fail to predict for set size 12 and 18. The sub-data set with set size 18 also had a considerable proportion of particularly slow correct rejections that was not predicted by the model. For the hits, the model underestimated the peak of the distribution.

The major discrepancies regarding incorrect RTs were different to conjunction data. Despite some deviation, the model was able to capture the patterns of the empirical false alarm RTs essentially. Empirical misses for set size 3 were clearly slower than the model predicted. For set size 12 and 18, there were both particularly fast and particularly slow misses that the model was not able to predict.

The parameter estimates are consistent with the characteristics of the spatial configuration search. For $c = 5$, the mean interarrival time $\frac{1}{\lambda}$ was estimated at 81.48 ms. It is much larger than $\hat{\frac{1}{\lambda}}$ for conjunction search and indicates a lower efficiency of early processing of the features. The dimensions of the features

were comparable to those for conjunction search. The feature color was simpler since all stimuli were white. The other dimension is more complex, containing horizontal and vertical lines. In the conjunction search task, every stimulus can only have either orientation (vertical or horizontal). In spatial configuration task, every stimulus has both horizontal and vertical components in exactly the same amount (three horizontal and two vertical). Five lines are not only superior in the amount than a bar, processing them also requires encoding the spatial information so that the components of the same stimulus can be identified as belonging together. Nevertheless, it is unclear to what extent the amount or the additional spatial information affects the efficiency of early processing of features.

The mean service time $\frac{1}{\mu}$ was estimated at 234.91, 360.83, 493.24 and 500.74 for $k = 3, 6, 12, 18$. Regardless of set size, these times are clearly longer than the estimated mean service time for conjunction search. This means that integrating the features of a spatial configuration stimulus requires much longer time. The difficulty of the spatial configuration task arises in both the early processing stage and the recognition stage. The recognition of the stimuli relies on precise encoding of the coordinate information of all the components and correct location of their relative positions, whereas such detailed processing are not necessary for the recognition of a conjunction search stimulus.

The observation that the estimated mean service time increased with increasing set size indicates that a longer time is required for recognizing each single stimulus as there are more stimuli in display. I connect this observation to the adaptation of discrimination threshold and explain the increased necessary time for recognition of spatial configuration stimuli as the additional costs for a more strict threshold for large set sizes. The first reason is that the increase of estimated mean service time concerns a dependence of the processing of each single stimulus in the recognition stage on the set size. In this model, there are several magnitudes explicitly specified as set size dependent. Whereas the termination rule describes a dependence of the termination criterion on set size on a trial, the threshold adaptation describes a dependence of the discrimination threshold on set size during the identification of individual stimuli. According to the model, the probability of misidentifying a distractor for the target p_1

decreases with increasing set size. This means that the discrimination threshold gets more strict when there are more stimuli in the display. It seems plausible that the integration must last longer to ensure that the higher precision level can be achieved. The second reason is that the change in the estimated mean service time accords with that in p_1. The increase of $\widehat{\frac{1}{\mu}}$ is not linear. It gets slower as set size increases: ca. 130 ms from set size 3 to 6, ca. 130 ms from set size 6 to 12 and ca. 8 ms from set size 12 to 18[2]. Similarly, the model also predicts p_1 to increase with a decreasing rate. This is apparent from the derivative of p_1 with respect to k.

Why did the estimated mean service time increase for the spatial configuration task but not for the conjunction task? First of all, relaxing the constraint on $\frac{1}{\mu}$ did yield a better fit for all sub-data sets. The estimated mean service time under the relaxed constraint also showed an increasing pattern. Yet the increase was fairly small so that the estimated mean service times for different set size levels can be considered approximately the same. The model assuming set size invariant parameters was preferred according to Occam's razor. Assume that the model with set size dependent $\frac{1}{\mu}$ does describe the reality better and $\frac{1}{\mu}$ does increase with increasing set size for the conjunction search task, how can one explain the different extent of increase for conjunction search and spatial configuration task? A larger increase for the spatial configuration search task may result from a larger marginal cost of the threshold adaptation, which then results from the more complex processing of the recognition stage. As discussed above, the recognition of a spatial configuration stimulus requires assembling the components correctly according to their spatial relations. It is insofar plausible that longer additional time is required when striving for the same increase of precision level of identification for spatial configuration. Even though p_1 increases faster for conjunction search according to the estimates of b, the corresponding marginal cost of the threshold adaptation can be so slow that it results in smaller additional processing time altogether.

[2]Note that the set size did not increase always with a step of 3. Set size 9 had not been realized in the experiment.

Chapter 8

General discussion

Mathematics is not about numbers, equations, computations, or algorithms: it is about understanding.
— *William Paul Thurston*

8.1 Summary of main results of the project

From a theoretical perspective, the major achievement of this dissertation project is a novel mathematical model of visual search that accounts for RT and accuracy data simultaneously on a distributional level. The queueing model of visual search successfully implements the conceptualization of Moore and Wolfe (2001) of a hybrid model which is both serial and parallel. In the notion of the queueing model of visual search, there is a bottle neck in the visual processing where visual stimuli enter one after another but then are processed further in parallel. The model possesses the properties of incorporating a long processing time of a few hundred milliseconds for each single item and at the same time being capable of predicting a rapid processing rate of under a hundred milliseconds per item. In this way, the queueing model of visual search demonstrated the possibility of accommodating seemingly conflicting empirical findings within a unified framework.

The queueing model of visual search is well embedded in the existing body of theories on visual attention. It is based on the same notion of FIT and GS regarding the basic architecture of visual processing. It differentiates between the early visual processing stage characterized by massive, parallel processing of stimulus features and the later recognition stage characterized by integration of features and formation of an inherent representation of the object. The model assumption on arrivals with an effective interarrival rate corresponds to the concept of parallel early processing stage. The model assumption on services at parallel servers corresponds to an elaborative recognition stage. Relying on these theoretical roots, the innovation of the model lies in the specification of the assignment of feature representations to the recognition stage when processing resources is not immediately available by queueing and parallel servers.

The queueing model of visual search has also shown its explanatory power for empirical data. Fitting the model to empirical data yielded very good fits for both accuracy and RT data. The model is capable of predicting error rates as well as RT distributions that are very close to empirical data under any experimental condition. This means that the queueing model does not only possess conceptual foundation that are intuitive and convincing but also empirical supports. It is able to explain accuracy and RT data simultaneously to a large degree of precision on a sophisticated level. Its explanatory power shows that limited-capacity parallel models are not the only model class in line with rapid search rate and long processing time. Under a hybrid construction, it is possible for models involving a mandatory serial processing stage to accommodate previous findings and make very fine, differentiated predictions that strikingly accord well with empirical data under strict restrictions.

A remarkable property of the model is its ability to account for aggregated data. As sufficiently large data sets from an individual observer are intractable, the model was fitted to data aggregated across several human observers. Although many numerical difficulties appeared during the preliminary model assessment, fitting the model to aggregated empirical data successfully led to interpretable results and good model fits. Complex as the model is, it also seems to be robust to the interference of individual differences of the participants.

A further achievement of this dissertation project is a comprehensive, holistic

view of modeling visual search, which is reflected in four aspects. First, the queueing model of visual search takes both RT and accuracy data into account. It is not a simple combination of two models which account for RT and accuracy data independently of each other. The RT model relies on the prediction of a termination criterion using the accuracy model. This connects both parts of the model in a meaningful way. The explanation of the increasing estimated mean service time also relies on a model component accounting of the accuracy model and the corresponding observations. Second, the model makes differentiated predictions depending on set size and target presence and combines information from sub-data sets under different experimental conditions to make statistical inference on the parameters. It is the simultaneous consideration of all sub-data sets that helped overcome the numerical difficulties and identifiability issues occurred in the recovery studies. Third, the queueing model of visual search takes RTs of both correct and incorrect trials into account. Instead of simply excluding the data of incorrect trials from the analysis, the model recognizes the importance of incorrect RTs and makes full use of information using the accuracy model specified. Fourth, the queueing model of visual search does not only account for summary statistics of the RT data, such as mean and variance, but also the entire distribution of RTs.

From a methodological perspective, this dissertation project provides above all a detailed demonstration of a scientific process of developing a novel mathematical model as a tool to explain psychological phenomena. The development of the model entails reasoning based on empirically observed patterns, theoretical construction, building and examining the mathematical foundations, as well as technical implementation and rectification. The insights and experience gained from the model development are not limited to the this specific model but transferable to modeling a broader class of psychological phenomena using models from other families. It explored and solved a practical problem of applying a model without an analytical representation. It demonstrated the utility of a Monte Carlo simulation based approach and the MDE approach. Along the way, a scheme of strategies and techniques for overcoming difficulties in numerical estimation has been developed.

8.2 Broader implications

8.2.1 Learning from errors in visual search

Taking error rates and error RTs into account addresses an epistemic issue. In previous studies, error trials were always excluded from the analysis based on the fact that the data collected contain only a very small proportion of errors. The low average error rate, however, is related to the fact that the visual search tasks chosen in standard visual search experiments are mostly very easy ones. The tasks have been designed to be simple mainly for the purpose of neat experimentation (see Section 2.3). Easy tasks are often accompanied by low error rates, which are regarded as desirable since the researcher can easily exclude all error trials from the analysis without misgivings. The data will be clean for standard evaluation methods that do not deal with errors. The sample size will be at the same time large because the large part of the trials can be preserved, improving the ratio of analyzable data to the expense on data collection. In other words, it would be still tempting to construct the search task in a way that keeps the error rate low even if the demand of neat experimentation had not existed.

On the other hand, the downside of restricting the study of visual search to easy tasks is also obvious — we would not be able to reveal the secrets behind behavior in more demanding, error-prone visual search situations. Visual search in everyday life is performed above all in naturalistic scenes, where observers are confronted with more complex tasks and are error-prone. Is it the same mechanism that is in operation as when performing easy tasks in the laboratory? Or the human mind processes complex visual search tasks by engaging different or additional mechanisms? To approach the answers to these questions, we must investigate visual search behavior in complex or difficult tasks. We may need to analyze data containing frustratingly large proportion of error trials. Excluding all error trials from the analysis in such situations is not justified anymore.

8.2.2 Thinking outside the black box

An observation of the research field on visual attention is that theories or theoretical constructs usually cluster around a certain phenomenon detected

by using a certain paradigm of either class described in Section 2.1.2. For a specific phenomenon, there may be several theories developed in the attempt to explain it, but theories explicitly explaining phenomena across experimental paradigms from both classes are rare. The models reviewed in Section 2.2 aim to explain empirical findings of visual search paradigm, above all search efficiency. Empirical findings regarding attentional dwell time and attentional blink have been used merely as a primitive check. There are rarely attempts of detailed explanation of these experimental findings by models of visual search, i.e., how does the model predict the patterns of interference. Vice versa, there are many theories aiming to explain attentional blink, but they do not concern explaining findings of visual search.

Although it is the time course of attention that both classes of paradigms aim to study, overarching theories are rare. The limitation of explanatory power across paradigms was sometimes attributed to different mechanisms underlying different phenomena. There is certainly a possibility that the observations made using visual search paradigms and attentional dwell time paradigm (and the related paradigms such as RSVP) are resulted from different mechanisms. But there is so far no conclusive evidence that can rule out this possibility or its complement. If we accept the notion that there is one single mechanism operating visual attention, any theory of visual attention would be obligated to provide explanations for observations produced by vast paradigms. The situation would be comparable to the parable of a group of blind men trying to conceptualize what an elephant is like by touching a part of its body. The blind men ended up arguing against each other because each claimed a different description of the elephant based on his own experience. By applying all kinds of paradigms, we have collected a great amount of experimental data. What if we had gained fragment pictures of the black box, yet the pieces of a puzzle were still dispersed? What if the lack of conclusive theoretical explanation of each single phenomenon was due to an isolated view of one of the facets of the same entity? Undoubtedly, the only way to find convergence in such a situation is to consider different paradigms within a theoretical framework. It is time to break the barriers between research methods that restrict theoretical accounts on single phenomena.

Such a development has already emerged. As an example, to my knowl-

edge, TVA (in the newest version "theory of temporal visual attention" (TTVA) Bundesen, 1990; Petersen et al., 2012) is the only theory that has made explicit predictions on both the results of visual search paradigm and those of attentional dwell time paradigm by Duncan et al. (1994). In its original form as published in 1990, TVA concerns solely explaining findings of visual search with static, simultaneous presentation of stimuli. It was until 2012 that TVA was generalized to TTVA, incorporating theoretical elements explaining how states of attention change over time to accommodate findings of dynamic, asynchronous presentation of stimuli. The TVA was extended by introducing multiple races upon every appearance of a new stimulus and multiple calculations of attentional weights every time a new race is initiated. The result of every calculation of attentional weights is a redistribution of the attentional resources. TTVA explains the interference of T1 with T2 in the attentional dwell time paradigm by a lock time of the neurons retaining the representation of T1 in VSTM. Due to its head start in stimulus onset, T1 has an advantage in the first race and further in locking resources in the limited processing capacity of visual system. This advantage gets lost after the lock time, when resources are released, which would explain the recovery of the identification performance of T2 as SOA becomes longer. Petersen et al. (2012) assumed the lock time to be exponentially distributed.

8.2.3 The old good binding problem in a new model

The key of explaining the findings regarding attentional dwell time and of related paradigms within the framework of the queueing model of visual search is to reconsider the binding problem (e.g., Treisman & Gelade, 1980). The study of the binding problem has been focused on processing static pictures, where temporal information plays a subordinate role in the recognition of visual objects. However, in paradigms using dynamic pictures, identification would not be possible without encoding temporal information.

In our imagination, objects enter the visual system as a whole. But this appears not to be the case. The signals streaming into our visual system are nothing other than a mess. How do we know that some materials belong to the same objects? How do we group them out from the stream of impressions? We surely need spatiotemporal information. We utilize spatiotemporal discontinuities as signals

to create new object tokens. Spatioemporal discontinuities indicate that there is a high probability that signals separated by them belong to the same object. Although attention is likely to operate along the master map representing spatial information, it is unthinkable that it works without consideration of temporal information.

The interference observed in the attentional dwell time paradigm and related paradigms may be caused by a confusion in the spatiotemporal information the features carry. Segments of the representation of the objects stream along different channels of feature map, converging to the temporal pool of feature activations. There, attention runs along the master map, picking out feature representations with the same spatiotemporal coordinate. In static pictures, the temporal information a piece of representation carries is completely determined by the subjective time perception of the observer. There is a clear, unique reference point — onset of all stimuli. In dynamic pictures, temporal information is not easy to determined. The appearance of the mask or of the distractors (or T2) is not initiated by the observer. The observer must first perceive the instance of the change and then mark this instance as a further reference point to the internal clock. The position of this instance on the internal time coordinate do not necessarily match the objective time. Two events in very close temporal proximity cannot be differentiated subjectively. The SOA of mask or T2 implemented in the experiments was usually larger than 50 to 100 ms, such that they can be perceived as individual pictures. The probability of separate them as individual items should be high, otherwise identification will drop to zero. However, the rate may be still too fast for the observer to add accurate temporal labels to the incoming segments of representations. As a result, the temporal information may not be encoded accurately enough for attention to always pick out the right feature representations from different channels. There may be a probability that one or some feature representation of the mask or of T1 is labeled falsely and taken incorrectly as parts of T2 so that the fragments considered belonging to T2 cannot be assembled appropriately to form a correct representation of T2.

This can also explain why the mask reduced the identification accuracy of T1 in the experiments of Moore et al. (1996). The mask affects backwardly in time, because sometimes the assembly of T1 parts is still ongoing upon the onset of

the mask and if the arrival of mask parts is happen to be fast enough. The new feature representations are crashing in, the allocator may get confused and think that these parts also belong to T1 so it assign them to the same server where the T1 parts are still being assembled. Then the assembly will fail because the parts do not fit together.

This notion is also inline with the finding that the standard RSVP reduces the identification performance of both T1 and T2. All these distractors in between leave confusing spaciotemporal information in the visual system.

There are some questions seem to not agree with this notion, though. First, why was the performance not affected by SOA when subjects were instructed to ignore T1 or T2? Second, Theeuwes et al. (2004) used static picture, how can this notion explain the finding?

If something is not attended, even if their feature representations have entered the system, they will not be assigned to any server and their temporal information is not viewed as relevant. Or they may still be assigned, but not carefully, so that the spatiotemporal information is not taken as reference point. The temporal information of T2 is selected as reference point and thus become more accurate. If one does not have to react to the occurrence of the proceeding event, one can estimate the occurrence of the event one is supposed to attend to.Besides, T2 identification was not completely independent of SOA even under the "ignoring T1" condition. The performance declined for small SOA.

As to the Theeuwes et al. (2004) paradigm, their finding is not conflicting with this notion. This paradigm does not measure interference. The character of their paradigm is that processing is forced to be strictly serial. Observer can only move their attention to the second focus location correctly after they have successfully recognized the first target (arrow). Assume that their estimation is accurate, the 250 ms can be interpreted as the time required for the observer to *complete* the processing (identification) of the first target. This is completely in line with the queueing model.

When the picture is static, temporal information can be coded correctly along the temporal coordinates, so that the assignment of feature representations to the server can work correctly. Errors can arise, but extremely rare. In that case, we observe illusory conjunction.

In sum, the queueing model of visual search seems promising in accommodating empirical findings using paradigms of the other class, if the time dimension is taken into account for feature integration. Nevertheless, a quantitative examination of this claim is required. There is much work to do in this direction of future research.

8.3 Limitations, open questions and outlook

8.3.1 Inhibition of return and termination rule

As discussed in Section 3.4.2, the current model assumes a perfect inhibition of return. However, this was an expedient choice for the sake of simplicity. There is empirical evidence showing that the inhibition of return is not sufficient in fact (see Section 4.2.1). In accordance with that, there was a considerable proportion of slow responses for target-absent sub-data sets that the model could not predict. I consider inadequate inhibition of return a source for these particularly slow responses.

However, inadequate inhibition of return cannot be the only source. It implies merely that already searched items are possible to be sampled again. If the observer follows the termination rule strictly, the search on a target-absent trial will still terminates in the time as the model predicts. If the longer RTs are not due to extended processing time of individual items, then it must arise from searching a extended number of items. A possibility is that the counter for searched items does not work adequately when the number of searched items exceed certain value. That is, as more items have been searched, the counter reflects this number less accurately. Although the observer try to follow the termination rule, he or she may fail to do so perfectly because a critical element does not work accurately. The observer may think that he or she has stopped exactly according to the termination rule, but in fact the number of searched item may deviate strongly from that the termination rule prescribes. This could explain the major discrepancies for large set size levels. There were both particularly quick and particularly slow correct rejections that could not be explained by the model.

Obviously, to result in particularly slow correct rejections, inadequate inhibition of returns is necessary. If inhibition of returns is perfect, the system will stop

at the latest when all items in the display have been searched. The slow correct rejections in the empirical data were much slower than that.

Therefore, a question that needs investigation is how to model inadequate inhibition of return and the counter for searched items. The answer to this question affects the predictions on both the accuracy and RT. As discussed in Section 3.4.2, the model assumption of perfect inhibition of return is implemented as effective interarrival rates depending on the number of departures. If the model retains the original specification as a Markovian queueing model, it implements the assumptions of no inhibition of return.

8.3.2 Guidance

Another theoretical issue related to the discrepancies observed for RTs is guidance. The model does not assume any guidance for the target in the sense that the target enters the system at a random position with equal probability. However, for large set size, there seems to be very efficient search that the model did not predict. The model underestimated the frequency of fast hits systematically for large set size levels. This can be a sign for the presence of certain guidance. Incorporating guidance in the model will have consequences on the prediction of RTs.

8.3.3 Explaining empirical findings using variants of visual search paradigm

If the model reveals general mechanisms of visual search, it should be able to explain empirical findings using variants of visual search other than the standard visual search as well given appropriate adaptation.

Within the standard visual search task, feature search remains unexplained. Since the model assumes a mandatory recognition stage and feature search does not require processing in this stage for a correct response, the model is not applicable to feature search data. However, if it can specify a relation of the timing of the decision to the arrivals, predictions on RTs for feature search are possible. An assumption could be for instance that a system response is made after the completion of processing features.

An interesting variant is visual search with redundant or multiple targets. In the studies with redundant targets, participants were aware of the fact that some of the trials would contain more than one target. Taking this knowledge into account, the posterior probability is different. Consequently, the structure of the problem will be different due to the different reward rate. However, the participants were not told about the proportions of each type of trials. This means that the reward rate cannot be (approximately) calculated. If observers wants to maximize the reward rate, they need to estimate the proportions first, which forces them to perform exhaustive search for a while and to "keep statistics". But then it is uncertain whether the effort for the determination of termination rule would pay off. The gain on search efficiency by applying a termination rule may not justify the lost by collection information to inference on the optimal termination rule. The problem with redundant targets is much more complex than it appears.

8.3.4 Further methodological developments

Although the model has been fitted to empirical data successfully, there was ambiguities when comparing models and interpreting the results. So far, the analysis of the results has been based on the values of distances and visualization of comparisons. Some important questions were not able to be answered clearly. For example, how well does the sum of discrepancies perform as a measure of fit? Can it differentiate model variants clearly? Did the restriction on $\frac{1}{\mu}$ reduce the fit significantly? For which values of c was the fit of which sub-data sets significantly better than others? A measure of goodness of fit could help making decisions regarding this kind of questions. Without a thorough study using for example bootstrapping methods, it is difficult to answer these questions.

Test statistics derived from the distance measures presented in Appendix H (or their variants) have been studied in the literature. For the Wasserstein metric of order 1, which has been applied in the parameter estimation here, it is also possible to develop proper measures of goodness of fit. However, this is a challenging task and goes beyond the scope of this dissertation.

Another useful methodological concept is a confidence interval for the model parameters. It helps answering questions such as how reliable is the estimate

of $\frac{1}{\lambda}$, are there significant difference between the residual times in different task types. Developing methods of a confidence interval for the queueing model also requires extensive work.

8.3.5 Possible extensions

As a final remark for this dissertation, I see many possibilities of applying the major results of this dissertation to various contexts of cognitive science. Since queueing models in general reflects a certain order in the resource allocation when there is demand conflicts, it is conceivable to apply them in explaining other cognitive processing that involves resource allocation. Queueing models can be applied in the study of complex systems whose processing nature display serial and parallel aspects, probably with recursive information processing, e.g., memory search and decision making between multiple options.

The general principles of foraging theory have many possibilities of application as well. When it comes to searching with efforts and uncertain outcomes (e.g., collecting information), the optimization of search efficiency may be able to explain the decision making of when to stop the search process.

References

Barndorff-Nielsen, O. E., & Cox, D. R. (1994). *Inference and asymptotics*. London, England, Chapman & Hall.

Basu, A., Shioya, H., & Park, C. (2010). *Statistical inference: The minimum distance approach*. Boca Raton, FL, CRC Press.

Bhat, U. N. (2015). *An introduction to queueing theory: modeling and analysis in applications* (2nd ed.). Basel, Swiss, Birkhäuser. https://doi.org/10.1007/978-0-8176-8421-1

Boos, D. D. (1981). Minimum distance estimators for location and goodness of fit. *Journal of the American Statistical Association*, *76*(375), 663–670.

Boos, D. D. (1982). Minimum Anderson-Darling estimation. *Communications in Statistics — Theory and Methods*, *11*(24), 2747–2774. https://doi.org/10.1080/03610928208828420

Breitmeyer, B., Ogmen, H., & Öğmen, H. (2006). *Visual masking: Time slices through conscious and unconscious vision*. London, England, Oxford University Press.

Bricolo, E., Gianesini, T., Fanini, A., Bundesen, C., & Chelazzi, L. (2002). Serial attention mechanisms in visual search: A direct behavioral demonstration. *Journal of Cognitive Neuroscience*, *14*(7), 980–993. https://doi.org/10.1162/089892902320474454

Broadbent, D. E., & Broadbent, M. H. P. (1987). From detection to identification: Response to multiple targets in rapid serial visual presentation. *Perception and Psychophysics*, *42*(2), 105–113. https://doi.org/10.3758/BF03210498

Brown, S. D., & Heathcote, A. (2008). The simplest complete model of choice response time: Linear ballistic accumulation. *Cognitive Psychology*, *57*(3), 153–178. https://doi.org/10.1016/j.cogpsych.2007.12.002

Bundesen, C. (1990). A theory of visual attention. *Psychological Review*, *97*(4), 523–547. https://doi.org/10.1037/0033-295X.97.4.523

Bundesen, C., & Habekost, T. (2004). Attention (K. Lamberts & R. Goldstone, Eds.; 1st ed.). In K. Lamberts & R. Goldstone (Eds.), *Handbook of cognition* (1st ed.). London, England, SAGE Publications.

Cain, M. S., Vul, E., Clark, K., & Mitroff, S. R. (2012). A bayesian optimal foraging model of human visual search. *Psychological Science*, *23*(9), 1047–1054. https://doi.org/10.1177/0956797612440460

Cameron, E. L., Eckstein, M., Tai, J., & Carrasco, M. (2004). Signal detection theory applied to three visual search tasks — identification, yes/no detection and localization. *Spatial Vision*, *17*(4), 295–325. https://doi.org/10.1163/1568568041920212

Carlson, J. M., & Doyle, J. (1999). Highly optimized tolerance: A mechanism for power laws in designed systems. *Physical Review E*, *60*(2), 1412.

Charnov, E. (1976). Optimal foraging theory: the marginal value theorem. *Theoretical Population Biology*, *9*, 129–136. https://doi.org/10.1016/0040-5809(76)90040-X

Chun, M. M., & Potter, M. C. (1995). A two-stage model for multiple target detection in rapid serial visual presentation. *Journal of Experimental Psychology*, *21*(1), 109–127.

Chun, M. M., & Wolfe, J. M. (1996). Just say no: How are visual searches terminated when there is no target present? *Cognitive Psychology*, *30*(1), 39–78. https://doi.org/10.1006/cogp.1996.0002

Cousineau, D., & Shiffrin, R. M. (2004). Termination of a visual search with large display size effects. *Spatial Vision*, *17*(4-5), 327–352. https://doi.org/10.1163/1568568041920104

Dehaene, S. (2011). *The number sense: How the mind creates mathematics*. New York, NY, Oxford University Press.

Dodd, M. D., Castel, A. D., & Pratt, J. (2003). Inhibition of return with rapid serial shifts of attention: Implications for memory and visual search. *Perception & Psychophysics*, *65*(7), 1126–1135.

Donkin, C., & Shiffrin, R. M. (2011). Visual search as a combination of automatic and attentive processes. *Proceedings of the Annual Meeting of the Cognitive Science Society, USA*, *33*, 2830–2835.

Duncan, J. (1987). Attention and reading: Wholes and parts in shape recognition — A tutorial review (M. Coltheart, Ed.). In M. Coltheart (Ed.), *Attention and Performance XII: The Psychology of Reading*, Hove, England, Erlbaum.

Duncan, J., & Humphreys, G. W. (1989). Visual search and stimulus similarity. *Psychological Review*, *96*(3), 433–458. https://doi.org/10.1037/0033-295X.96.3.433

Duncan, J., & Humphreys, G. W. (1992). Beyond the search surface: Visual search and attentional engagement. *Journal of Experimental Psychology: Human Perception and Performance*, *18*(2), 578–588. https://doi.org/10.1037/0096-1523.18.2.578

Duncan, J., Ward, R., & Shapiro, K. L. (1994). Direct measurement of attentional dwell time in human vision. *Nature*, *369*(6478), 313–315. https://doi.org/10.1038/369313a0

Egeth, H. E., & Dagenbach, D. (1991). Parallel versus serial processing in visual search: Further evidence from subadditive effects of visual quality. *Journal of Experimental Psychology: Human Perception and Performance*, *17*(2), 551–560. https://doi.org/10.1037/0096-1523.17.2.551

Egeth, H. E., Virzi, R. A., & Garbart, H. (1984). Searching for conjunctively defined targets. *Journal of Experimental Psychology: Human Perception and Performance*, *10*(1), 32–39. https://doi.org/10.1037/0096-1523.10.1.32

Ehinger, K. A., & Wolfe, J. M. (2016). When is it time to move to the next map? Optimal foraging in guided visual search. *Attention, Perception, and Psychophysics*, *78*(7), 2135–2151. https://doi.org/10.3758/s13414-016-1128-1

Eriksen, C. W. (1966). Independence of successive inputs and uncorrelated error in visual form perception. *Journal of Experimental Psychology*, *72*(1), 26–35. https://doi.org/10.1037/h0023316

Eriksen, C. W., & Spencer, T. (1969). Rate of information processing in visual perception: Some results and methodological considerations. *Journal of Experimental Psychology*, *79*(2, Pt. 2), 1–16.

Fisher, D. L. (1982). Limited-channel models of automatic detection: Capacity and scanning in visual search. *Psychological Review*, *89*(6), 662–692.

Fisher, D. L. (1984). Central capacity limits in consistent mapping, visual search tasks: Four channels or more? *Cognitive Psychology*, *16*(4), 449–484. https://doi.org/10.1016/0010-0285(84)90017-3

Friedman-Hill, S., & Wolfe, J. M. (1995). Second-order parallel processing: Visual search for the odd item in a subset. *Journal of Experimental Psychology: Human Perception and Performance*, *21*(3), 531–551.

Gilchrist, I. D., & Harvey, M. (2006). Evidence for a systematic component within scan paths in visual search. *Visual Cognition*, *14*(4-8), 704–715.

Godwin, H. J., Menneer, T., Cave, K. R., Thaibsyah, M., & Donnelly, N. (2015). The effects of increasing target prevalence on information processing during visual search. *Psychonomic Bulletin and Review*, *22*(2), 469–475. https://doi.org/10.3758/s13423-014-0686-2

Green, D. M., & Swets, J. A. (1966). *Signal detection theory and psychophysics.* New York, NY, Wiley.

Gross, D., Shortle, J. F., Thompson, J. M., & Harris, C. M. (2008). *Fundamentals of queueing theory* (4th ed.). Hoboken, NJ, Wiley.

Heathcote, A., Brown, S. D., & Mewhort, D. J. K. (2002). Quantile maximum likelihood estimation of response time distributions. *Psychonomic Bulletin and Review*, *9*(2), 1–31. https://doi.org/10.3758/BF03196299

Horowitz, T. S., & Wolfe, J. M. (1998). Visual search has no memory. *Nature*, *394*(6693), 575–577.

Horowitz, T. S., & Wolfe, J. M. (2003). Memory for rejected distractors in visual search? *Visual Cognition*, *10*(3), 257–298.

Horowitz, T. S., Wolfe, J. M., Alvarez, G. A., Cohen, M. A., & Kuzmova, Y. I. (2009). The speed of free will. *The Quarterly Journal of Experimental Psychology*, *62*(11), 2262–2288. https://doi.org/10.1080/17470210902732155

Hout, M. C., Walenchok, S. C., Goldinger, S. D., & Wolfe, J. M. (2015). Failures of perception in the low-prevalence effect: Evidence from active and passive visual search. *Journal of Experimental Psychology: Human Perception and Performance*, *41*(4), 977–994. https://doi.org/10.1037/xhp0000053

Ishibashi, K., Kita, S., & Wolfe, J. M. (2012). The effects of local prevalence and explicit expectations on search termination times. *Attention, Perception, and Psychophysics*, *74*(1), 115–123. https://doi.org/10.3758/s13414-011-0225-4

Klein, R. M. (1988). Inhibitory tagging system facilitates visual search. *Nature*, *334*(6181), 430–431.

Klein, R. M., & MacInnes, W. (1999). Inhibition of return is a foraging fascilitator in visual search. *Psychological Science*, *10*(4), 346–352.

Kleiss, J. A., & Lane, D. M. (1986). Locus and persistence of capacity limitations in visual information processing. *Journal of Experimental Psychology: Human Perception and Performance*, *12*(2), 200–210. https://doi.org/10.1037/0096-1523.12.2.200

Krueger, L. E. (1984). Perceived numerosity: A comparison of magnitude production, magnitude estimation, and discrimination judgments. *Perception and Psychophysics*, *35*(6), 536–542. https://doi.org/10.3758/BF03205949

Kyllingsbæk, S. (2006). Modeling visual attention. *Behavior Research Methods*, *38*(1), 123–133. https://doi.org/10.3758/BF03192757

Liu, Y. (1993). *Queueing networks as models of human performance and human-computer interaction* (tech. rep. No. 93-32). University of Michigan. Ann Arbor, MI. Retrieved from https://deepblue.lib.umich.edu/bitstream/handle/2027.42/6352/bap9100.0001.001.pdf?sequence=5{\&}isAllowed=y.

Liu, Y. (1996). Queueing network modeling of elementary mental processes. *Psychological Review*, *103*(1), 116–136.

Liu, Y. (2013). Queuing and network models (J. D. Lee & A. Kirlik, Eds.). In J. D. Lee & A. Kirlik (Eds.), *The oxford handbook of cognitive engineering*. New York, NY, Oxford University Press.

Mandelbrot, B. (1953). An informational theory of the statistical structure of language. *Communication Theory*, *84*, 486–502.

McNamara, J. (1982). Optimal patch use in a stochastic environment. *Theoretical Population Biology*, *21*(2), 269–288. https://doi.org/10.1016/0040-5809(82)90018-1

Metcalfe, D., & Barlow, K. (1992). A model for exploring the optimal trade-off between field processing and transport. *American Anthropologist*, *94*(2), 340–356. https://doi.org/10.1525/aa.1992.94.2.02a00040

Miller, J. (1993). A queue-series model for reaction time, with discrete stage and continuous flow models as special cases. *Psychological Review*, *100*(4), 702–715.

Moore, C. M., Egeth, H. E., Berglan, L. R., & Luck, S. J. (1996). Are attentional dwell times inconsistent with serial visual search? *Psychonomic Bulletin and Review*, *3*(3), 360–365. https://doi.org/10.3758/BF03210761

Moore, C. M., & Wolfe, J. M. (2001). Getting beyond the serial/parallel debate in visual search: A hybrid approach (K. L. Shapiro, Ed.). In K. L. Shapiro (Ed.), *The limits of attention: Temporal constraints on human information processing*. New York, NY, Oxford University Press.

Moran, R., Zehetleitner, M., Müller, H. J., & Usher, M. (2013). Competitive guided search: Meeting the challenge of benchmark RT distributions. *Journal of Vision*, *13*(8), 1–31. https://doi.org/10.1167/13.8.24

Mordkoff, J. T., Yantis, S., & Egeth, H. E. (1990). Detecting conjunctions of color and form in parallel. *Perception and Psychophysics*, *48*(2), 157–168. https://doi.org/10.3758/BF03207083

Murphy, S. A., & van der Vaart, A. W. . (2000). On profile likelihood. *Journal of the American Statistical Association*, *95*(450), 449–465.

Nakayama, K., & Silverman, G. H. (1986). Serial and parallel processing of visual feature conjunctions. *Nature*, *320*(6059), 264–265. https://doi.org/10.1038/320264a0

Newman, M. E. J. (2005). Power laws, pareto distributions and zipf's law. *Contemporary Physics*, *46*(5), 323–351.

Palmer, E. M., Horowitz, T. S., Torralba, A., & Wolfe, J. M. (2011). What are the shapes of response time distributions in visual search? *Journal of Experimental Psychology: Human Perception and Performance*, *37*(1), 58–71. https://doi.org/10.1037/a0020747

Palmer, J. (1995). Attention in visual search: Distinguishing four causes of a set-size effect. *Current Directions in Psychological Science*, *4*(4), 118–123. https://doi.org/10.1111/1467-8721.ep10772534

Palmer, J., Verghese, P., & Pavel, M. (2000). The psychophysics of visual search. *Vision Research*, *40*(10-12), 1227–1268. https://doi.org/10.1016/S0042-6989(99)00244-8

Parr, W. C. (1981). Minimum distance estimation:a bibliography. *Communications in Statistics – Theory and Methods*, *10*(12), 1205–1224. https://doi.org/10.1080/03610928108828104

Parr, W. C., & De Wet, T. (1981). On minimum Cramér-von Mises-norm parameter estimation. *Communications in Statistics - Theory and Methods*, *10*(12), 1149–1166. https://doi.org/10.1080/03610928108828100

Parr, W. C., & Schucany, W. R. (1980). Minimum distance and robust estimation. *Journal of the American Statistical Association*, *75*(371), 616–624. https://doi.org/10.1080/01621459.1980.10477522

Parr, W. C., & Schucany, W. R. (1982). Minimum distance estimation and components of goodness-of-fit statistics. *Journal of the Royal Statistical Society. Series B (Methodological)*, *44*(2), 178–189. https://doi.org/10.2307/2985181

Pashler, H. (1987). Detecting conjunctions of color and form: reassessing the serial search hypothesis. *Perception and Psychophysics*, *41*(3), 191–201. https://doi.org/10.3758/BF03208218

Pashler, H., & Badgio, P. C. (1985). Visual attention and stimulus identification. *Journal of Experimental Psychology: Human Perception and Performance*, *11*(2), 105–121.

Patefield, W. M. (1977). On the maximized likelihood function. *The Indian Journal of Statistics, Series B (1960-2002)*, *39*(1), 92–96.

Peltier, C., & Becker, M. W. (2016). Decision processes in visual search as a function of target prevalence. *Journal of Experimental Psychology: Human Perception and Performance*, *42*(9), 1466–1476. https://doi.org/10.1037/xhp0000248

Petersen, A., Kyllingsbæk, S., & Bundesen, C. (2012). Measuring and modeling attentional dwell time. *Psychonomic Bulletin and Review*, *19*(6), 1029–1046. https://doi.org/10.3758/s13423-012-0286-y

Potter, M. C., & Levy, E. I. (1969). Recognition memory for a rapid sequence of pictures. *Journal of Experimental Psychology*, *81*(1), 10–15.

Ramdas, A., Trillos, N. G., & Cuturi, M. (2017). On Wasserstein two-sample testing and related families of nonparametric tests. *Entropy*, *19*(2), 1–15. https://doi.org/10.3390/e19020047

Raymond, J. E., Shapiro, K. L., & Arnell, K. M. (1992). Temporary suppression of visual procesing in a RSVP task: an attention blink? *Journal of Experimental Psychology*, *18*(3), 849–860.

Reeves, A., & Sperling, G. (1986). Attention gating in short term visual memory. *Psychological Review*, *93*(2), 180–206.

Rich, A. N., Kunar, M. A., Van Wert, M. J., Hidalgo-Sotelo, B., Horowitz, T. S., & Wolfe, J. M. (2008). Why do we miss rare targets? Exploring the boundaries of the low prevalence effect. *Journal of Vision*, *8*(15), 1–17. https://doi.org/10.1167/8.15.15

Saltelli, A., Tarantola, S., Campolongo, F., & Ratto, M. (2004). *Sensitivity analysis in practice: A guide to assessing scientific models*. Hoboken, NJ, Wiley.

Shapiro, K. L., Raymond, J. E., & Arnell, K. M. (1994). Attention to visual pattern information produces the attentional blink in rapid serial visual presentation. *Journal of Experimental Psychology: Human Perception and Performance*, *20*(2), 357–371.

Shiffrin, R. M., & Gardner, G. T. (1972). Visual processing capacity and attentional control. *Journal of Experimental Psychology*, *93*(1), 72–82. https://doi.org/10.1037/h0032453

Sommerfeld, M. (2017). *Wasserstein distance on finite spaces : Statistical inference and algorithms* (Doctoral dissertation, Georg-August-Universität Göttingen, Göttingen, Germany). Retrieved from https://ediss.uni-goettingen.de/bitstream/handle/11858/00-1735-0000-0023-3FA1-C/DissertationSommerfeldRev.pdf?sequence=1.

Sperling, G. (1963). A Model for Visual Memory Tasks. *Human Factors*, *5*(1), 19–31. https://doi.org/10.1177/001872086300500103

Stephens, D. W., & Krebs, J. R. (1986). *Foraging theory*. Princeton, NJ, Princeton University Press.

Sternberg, S. (1967, April). *Scanning a persisting visual image versus a memorized list.* Paper presented at the Annual meeting of the Eastern Psychological Association, Boston.

Sternberg, S. (1966). High-speed scanning in human memory. *Science*, *153*(3736), 652–654. https://doi.org/10.1126/science.153.3736.652

Stewart, W. J. (2009). *Probability, Markov chains, queues, and simulation: The mathematical basis of performance modeling*. Princeton, NJ, Princeton University Press.

Sung, K. (2008). Serial and parallel attentive visual searches: Evidence from cumulative distribution functions of response times. *Journal of Experimental Psychology: Human perception and performance*, *34*(6), 1372–1388. https://doi.org/10.1037/a0011852

Theeuwes, J., Godijn, R., & Pratt, J. (2004). A new estimation of the duration of attentional dwell time. *Psychonomic Bulletin and Review*, *11*(1), 60–64. https://doi.org/10.3758/BF03206461

Thornton, T. L., & Gilden, D. L. (2007). Parallel and serial processes in visual search. *Psychological Review*, *114*(1), 71–103. https://doi.org/10.1037/0033-295X.114.1.71

Townsend, J. T. (1971). A note on the identifiability of parallel and serial processes. *Perception and Psychophysics*, *10*(3), 161–163.

Townsend, J. T. (1990). Serial vs. parallel processing: Sometimes they look like Tweedledum and Tweedledee but they can (and should) be distinguished. *Psychological Science*, *1*(1), 46–54.

Townsend, J. T., & Ashby, F. G. (1983). *Stochastic Modeling of Elementary Psychological Processes*. Cambridge, England, Cambridge University Press.

Townsend, J. T., & Nozawa, G. (1995). Spatio-temporal properties of elementary perception: An investigation of parallel, serial, and coactive theories. *Journal of Mathematical Psychology*, *39*, 321–359. https://doi.org/10.1006/jmps.1995.1033

Townsend, J. T., & Roos, R. N. (1973). Search reaction time for single targets in multiletter stimuli with brief visual displays. *Memory and Cognition*, *1*(3), 319–332.

Townsend, J. T., & Wenger, M. J. (2004). The serial-parallel dilemma: A case study in a linkage of theory and method. *Psychonomic Bulletin and Review*, *11*(3), 391–418.

Treisman, A. M. (1988). Features and objects: The fourteenth Bartlett memorial lecture. *The Quarterly Journal of Experimental Psychology Section A*, *40*(2), 201–237. https://doi.org/10.1080/02724988843000104

Treisman, A. M., & Gelade, G. (1980). A feature-integration theory of attention. *Cognitive Psychology*, *12*(1), 97–136. https://doi.org/10.1016/0010-0285(80)90005-5

Treisman, A. M., & Gormican, S. (1988). Feature analysis in early vision: Evidence from search asymmetries. *Psychological Review*, *95*(1), 15–48. https://doi.org/10.1037/0033-295X.95.1.15

Treisman, A. M., Sykes, M., & Gelade, G. (1977). Selective attention and stimulus integration (S. Dornič, Ed.). In S. Dornič (Ed.), *Attention and Performance VI: Proceedings of the Sixth International Symposium on Attention and Performance*, Hillsdale, NJ, Erlbaum.

Vallender, S. (1974). Calculation of the Wasserstein distance between probability distributions on the line. *Theory of Probability and Its Applications*, *18*(4), 784–786. https://doi.org/10.1137/1118101

Verghese, P. (2001). Visual search and attention: A signal detection theory approach. *Neuron*, *31*(4), 523–535. https://doi.org/10.1016/S0896-6273(01)00392-0

Ward, R., Duncan, J., & Shapiro, K. (1997). Effects of similarity, difficulty, and nontarget presentation on the time course of visual attention. *Perception and Psychophysics*, *59*(4), 593–600. https://doi.org/10.3758/BF03211867

Ward, R., & McClelland, J. L. (1989). Conjunctive search for one and two identical targets. *Journal of Experimental Psychology: Human Perception and Performance*, *15*(4), 664–672. https://doi.org/10.1037/0096-1523.15.4.664

Wolfe, J. M. (1994a). Guided search 2.0: A revised model of visual search. *Psychnomic Bulletin and Review*, *1*(2), 202–238. https://doi.org/10.3758/BF03200774

Wolfe, J. M. (1994b). Visual search in continuous, naturalistic stimuli. *Vision Research*, *34*(9), 1187–1195.

Wolfe, J. M. (1998a). Visual search (H. E. Pashler, Ed.). In H. E. Pashler (Ed.), *Attention*. Hove, England: Psychology Press.

Wolfe, J. M. (1998b). What can 1 Million search trials tell us about visual search? *Psychological Science*, *9*(1), 33–39.

Wolfe, J. M. (2003). Moving towards solutions to some enduring controversies in visual search. *Trends in Cognitive Sciences*, *7*(2), 70–76. https://doi.org/10.1016/S1364-6613(02)00024-4

Wolfe, J. M. (2007). Guided search 4.0: Current Progress with a model of visual search (W. D. Gray, Ed.). In W. D. Gray (Ed.), *Integrated models of cognitive systems*. New York, NY, US, Oxford University Press. https://doi.org/10.1007/978-94-011-5698-1_30

Wolfe, J. M. (2012). When do I quit? The search termination problem in visual search. *Nebraska Symposium on Motivation*, 183–208. https://doi.org/10.1007/978-1-4614-4794-8-8

Wolfe, J. M. (2018). Visual search (J. T. Wixted & J. Serences, Eds.; 4th ed.). In J. T. Wixted & J. Serences (Eds.), *Steven's handbook of experimental psychology and cognitive neuroscience: Vol 2. sensation, perception, and attention* (4th ed.). Hoboken, NJ, Wiley.

Wolfe, J. M., Cain, M. S., & Alaoui-Soce, A. (2018). Hybrid value foraging: How the value of targets shapes human foraging behavior. *Attention, Perception, and Psychophysics*, *80*(3), 609–621. https://doi.org/10.3758/s13414-017-1471-x

Wolfe, J. M., Cave, K. R., & Franzel, S. L. (1989). Guided search: an alternative to the feature integration model for visual search. *Journal of Experimental Psychology: Human Perception and Performance*, *15*(3), 419–433. https://doi.org/2527952

Wolfe, J. M., Palmer, E. M., & Horowitz, T. S. (2010). Reaction time distributions constrain models of visual search. *Vision Research*, *50*(14), 1304–1311. https://doi.org/10.1016/j.visres.2009.11.002

Wolfe, J. M., & Van Wert, M. J. (2010). Varying target prevalence reveals two dissociable decision criteria in visual search. *Current Biology*, *20*(2), 121–124. https://doi.org/10.1016/j.cub.2009.11.066

Wolfe, J. M., Võ, M. L. H., Evans, K. K., & Greene, M. R. (2011). Visual search in scenes involves selective and nonselective pathways. *Trends in Cognitive Sciences*, *15*(2), 77–84. https://doi.org/10.1016/j.tics.2010.12.001

Woodman, G. F., & Luck, S. J. (1999). Electrohysiological measurement of rapid shifts of attention during visual search. *Nature*, *400*(6747), 867–869.

Wu, C., & Liu, Y. (2007). Queuing network modeling of driver workload and performance. *IEEE Transactions on Intelligent Transportation Systems*, *8*(3), 528–537.

Wu, C., & Liu, Y. (2008a). Queuing network modeling of the psychological refractory period (PRP). *Psychological Review*, *115*(4), 913–954. https://doi.org/10.1037/a0013123

Wu, C., & Liu, Y. (2008b). Queuing network modeling of transcription typing. *ACM Transactions on Computer-Human Interaction*, *15*(1), 1–45.

Zenger, B., & Fahle, M. (1997). Missed targets are more frequent than false alarms: A model for error rates in visual search. *Journal of Experimental Psychology: Human Perception and Performance*, *23*(6), 1783–1791. https://doi.org/10.1037/0096-1523.23.6.1783

Zhang, J., Gong, X., Fougnie, D., & Wolfe, J. M. (2015). Using the past to anticipate the future in human foraging behavior. *Vision Research*, *111*, 66–74. https://doi.org/10.1016/j.visres.2015.04.003

Appendices

A. Arrival pattern of a finite source queue with recurrent demand

This appendix provides detail for the statement that the effective arrival rate of a finite source queue with recurrent demand is proportional to the number of customers outside the system in Section 3.3.4.

In queueing theory, finite customer population is always assumed to have recurrent service demand. That is, every customer goes through the phases of needing the service (in queue) and not in need of the service (outside of the system) alternately. Once a customer has been served and left the system, she becomes a customer who does not need the service at the moment but will potentially get back again. This makes the queue continue operating.

The arrival scheme in case of a finite source with recurrent demand is characterized as follows. Assume that there are k customers in total and every customer at any time is either in the system or outside the system. Assume further that for each customer, the time which she spends outside the system is exponentially distributed with mean $\frac{1}{\lambda}$ (i.e., the interarrival rate is λ). That is, if a customer is not in the system at time t, the probability he will have entered the system by $t + \Delta t$ is $\lambda\Delta t + o(\Delta t)$[1]. Thus, if there are k customers outside the system at time t, the probability that one of them will have entered the system during $(t, t + \Delta t]$ is $k\lambda\Delta t + o(\Delta t)$. This means that the effective arrival rate is proportional to the number of customers outside the system, i.e., those who can potentially arrive in the future. As more customers join the queue, there remain fewer customers that are able to come at all. Consequently, the effective arrival rate declines. This arrival pattern retains the Markovian property, i.e., the number of customers in the queue at time t as a random process remains a continuous-time Markov chain (discussed in Section 3.4).

[1] o is one of the Landau symbols used to describe the asymptotic behavior of a function in relation to another function. That is, $f(x) = o(g(x))$ if $\lim_{x \to C} \frac{f(x)}{g(x)} = 0$. "$f(x)$ is little-o of $g(x)$" can be understood as $f(x)$ becomes negligible compared to $g(x)$ as x approaches a limit (e.g., infinity or a certain value, here the value 0).

B. Theoretical approach to deriving RT predictions from an $M/M/c/\infty/k/FCFS$ queue with recurrent demand

This appendix presents the Infinitesimal transition rates and the forward Kolmogorov equations of a $M/M/c/\infty/k/FCFS$ with recurrent demand. They are the two steps that constitute the theoretical approach to deriving RT predictions.

Infinitesimal transition rates The infinitesimal arrival (birth) rate depends on the state of the system and is proportional to the number of customers that have not arrived yet. If there are already n customers in the system at time t, the probability that an additional customer will have arrived in the system during $(t, t+\Delta t]$ is $(k-n)\lambda\Delta t + o(\Delta t)$. Let n $(n = 0, 1, \ldots, k)$ be the number of customers in the system and λ_n the *effective arrival rate*, then

$$\lambda_n = (k-n)\lambda \tag{8.1}$$

Similarly, under the assumption that the service times are exponentially distributed with mean $\frac{1}{\mu}$ (i.e., the service rate is μ), the infinitesimal departure (death) rate depends on the state of the system and is proportional to the number of active servers. If there are c busy servers, the probability that one of the customers at the server will have left the system during $(t, t+\Delta t]$ is $c\mu\Delta t + o(\Delta t)$. Let μ_n be the *effective departure rate*, then

$$\mu_n = \begin{cases} n\mu & \text{if } n = 0, 1, \ldots, c-1 \\ c\mu & \text{if } n = c, \ldots, k \end{cases} \tag{8.2}$$

To illustrate the transitions between states in an intuitive way, Figure 8.1 shows the rate transition diagram for a $M/M/c/\infty/k/FCFS$ queue. The number in the circle denotes the state, i.e., the numbers of customers in the system. Arrows represent the transition from one state to another. The symbols above or below the arrows represent the infinitesimal arrival and departure rates, respectively. Because only an increase or a decrease by one is possible, transitions occur only between two adjacent states (i.e., states with the absolute difference of one).

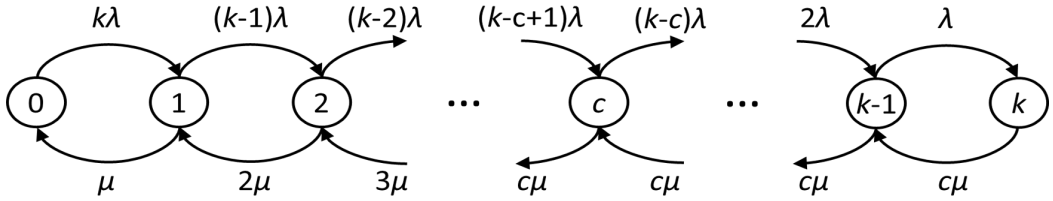

Figure 8.1: Rate transition diagram for a finite-source queue with c servers and k customers with recurrent demand. The number in the circle denotes the numbers of customers in the system.

Generator matrix Using the effective arrival and departure rates as in Equation (8.1) and (8.2), infinitesimal transitions between the states can be determined explicitly. It results in the following generator matrix for the birth-death process model of the Markovian queueing system:

$$\mathbf{Q} = \begin{array}{c} 1 \\ 2 \\ 3 \\ \vdots \\ \vdots \\ c-1 \\ c \\ c+1 \\ \vdots \\ \vdots \\ k-1 \\ k \end{array} \begin{pmatrix} -k\lambda & k\lambda & & & & & & & & & \\ \mu & -((k-1)\lambda+\mu) & (k-1)\lambda & & & & & & & & \\ & 2\mu & -((k-2)\lambda+2\mu) & (k-2)\lambda & & & & & & & \\ & & & \ddots & & & & & & & \\ & & & & \ddots & & & & & & \\ & & & & & (k-c+1)\lambda & & & & & \\ & & & & & -((k-c)\lambda+c\mu) & (k-c)\lambda & & & & \\ & & & & & c\mu & -((k-c-1)\lambda+c\mu) & & & & \\ & & & & & & & \ddots & & & \\ & & & & & & & & \ddots & & \\ & & & & & & & & & -(\lambda+c\mu) & \lambda \\ & & & & & & & & & c\mu & -c\mu \end{pmatrix}$$

The generator matrix gives us the time-evolution of the queueing system. Denote $P_n(t) = \mathbb{P}(Q(t) = n \mid Q(0) = 0)$ and let $P'_n(t)$ be the derivative of $P_n(t)$ with regard to t, then the forward Kolmogorov equations are given by $\mathbf{P}'(t) = \mathbf{P}(t)\mathbf{Q}$ in matrix notation. That is, the system of the forward Kolmogorov equations is

given by

$$
\begin{aligned}
P_0'(t) &= -k\lambda P_0(t) + \mu P_1(t) \\
P_n'(t) &= (k-n+1)\lambda P_{n-1}(t) - ((k-n)\lambda + \mu)P_n(t) + n\mu P_{n+1}(t), \\
&\qquad \text{for } n = 1, 2, \ldots, c \\
P_n'(t) &= (k-n+1)\lambda P_{n-1}(t) - ((k-c)\lambda + \mu)P_n(t) + c\mu P_{n+1}(t), \\
&\qquad \text{for } n = c+1, \ldots, k-1 \\
P_k'(t) &= \lambda P_{k-1}(t) - c\mu P_k(t)
\end{aligned}
$$

An outline of the theoretical approach In most applications where the behavior in equilibrium is of interest, this kind of differential equations are solved using standard mathematical or computational techniques. However, the commonly used techniques are not helpful in the current context because they are applied to solving the limiting result by letting $t \to \infty$ in the original equations. To obtain a finite time solution, we have to solve the equations in the original form as displayed above. Unfortunately, even in simple cases such as an $M/M/1/\infty/\infty$ queue (single server, unlimited customer population, thus transition rates do not depend on the state), where the forward Kolmogorov equations have a much simpler form, deriving $P_n(t)$ explicitly is an arduous process.

It adds to the difficulty that we are interested in the departure time eventually instead of the probabilities of being in a state. As mentioned in the beginning of this section, to derive the departure time, all finite combinations of arrival, service and departure order that can lead to the departure of interest have to be considered. Deriving T_p and T_a explicitly requires to solve a system of difference-differential equations involving $P_n(t)$, which would be extremely tedious. Furthermore, the form of $P_n(t)$ varies largely for different numbers of servers c, which is treated as a parameter in my model. Thus, even if we had obtained the explicit form of $P_n(t)$, it is very unlikely that the resulting CDF of T_p and T_a have a regular form, especially T_p. In sum, it seems hopeless to obtain useful analytical formulas for applying a parameter estimation method such as MLE.

C. R code for simulation of the queueing model and calculation of the Wasserstein distance

```
queue_r <- function(miat, mst, c, k, pr, L, misidd, misidt
   , arrival, serving, tposition) {
  if (L == 0) return(rep(0, 5))
  if (pr) misidd[tposition] <- 0
  stopat <- L
  resp <- FALSE
  if (!pr) {
    if (sum(misidd[1:L])!= 0) {
      stopat <- which(misidd != 0)[1]
      resp <- TRUE
    }
  } else {
    if (sum(misidd[1:L]) == 0) {
      if (resp <- misidt == 0 && tposition <= L) stopat <-
          tposition
    } else {
      resp <- TRUE
      stopat <- if (misidt == 0) min(which(misidd != 0)
         [1], tposition) else which(misidd != 0)[1]
    }
  }
  state <- rep(0, 4)
  out <- vector()
  el <-  rep(Inf, c + 1)
  ix <- rep(0, c)
  depid <- 0
  el[1] <- arrival[state[2] + 1] / ((k - state[2])/miat)
 m <- el[1]
  idx <- NA
```

```
while (m < Inf && ((!resp && state[3] < stopat) || (resp
    && depid != stopat))) {
  m <- min(el)
  if (el[1] == m) {
    out <- cbind(out, c(Inf, Inf))
    state <- c(m, state[-1] + c(1, 0, 1))
    out[1, state[2]] <- state[1]
    if (state[2] < k) {
      el[1] <- state[1] + arrival[state[2] + 1] / ((k -
         state[2]) / miat)
      stopifnot(is.finite(el[1]))
    } else {
      el[1] <- Inf
    }
    f <- which(ix == 0)
    if (state[4] <= c && length(f) != 0) {
      el[f[1] + 1] <- state[1] + serving[state[2]] * mst
      ix[f[1]] <- state[2]
    }
  } else {
    state <- c(m, state[-1] + c(0, 1, -1))
    idx <- which(el == m) - 1
    depid <- ix[idx]
    out[2, depid] <- m
    stopifnot(depid > 0)
    ix[idx] <- 0
    f <- which(ix == 0)
    if (state[4] >= c) {
      ix[f[1]] <- state[3] + (c - length(f)) + 1
      el[f[1] + 1] <- state[1] + serving[ix[f[1]]] * mst
  } else {
      el[f + 1] <- Inf
    }
```

```
      }
  }
  deptimes <- out[2, ]
  srt <- if(!resp) max(deptimes[is.finite(deptimes)]) else
      out[2, stopat]
  wt <- out[2, ] - out[1, ]
  wt <- wt[is.finite(wt)]
  stopifnot(length(wt) == state[3])
  if (srt != state[1]) stop("Caution!")
  output <- c(length(wt), mean(wt), max(wt), srt, resp)
  return(output)
}

queue <- function(miat, mst, c, pr, L, misidd, misidt,
   arrival, serving, tposition, old = FALSE) {
  if (old) {
    N <- ncol(arrival)
    resultOld <- matrix(NA, nrow = N, ncol = 5)
    for (simu in 1:N) {
      resultOld[simu, ] <- queue_r(miat, mst, c, nrow(
          arrival), pr, L[simu], misidd[, simu], misidt[
          simu], arrival[, simu], serving[, simu],
          tposition[simu])
    }
    if (old != FALSE)
      return(resultOld)
  }
  if (old != TRUE) {
    result <- .Call(C_queue, as.double(miat), as.double(
       mst), as.integer(c), as.integer(pr), as.integer(L),
        misidd, as.integer(misidt), arrival, serving, as.
       integer(tposition))
        }
```

```
  if (old > 1) stopifnot(!all(resultOld == result))
  return(result)
}

WM <- function(par, esterrorpar, c, k, pr, N, empRT, old =
    FALSE) {
  miat <- par[1]
  mst <- par[2]
  Tres <- par[3]
  alpha <- esterrorpar[1]
  beta <- esterrorpar[2]
  a1 <- esterrorpar[3]
  a2 <- esterrorpar[4]
  b <- esterrorpar[5]
  lk <- k*(1 - exp(alpha + beta*k))
  L <- floor(lk) + rbinom(n = N, size = 1, prob = lk -
     floor(lk))
  p1 <- a1*k^(-b)
  p2 <- a2*k^b
  misidd <- matrix(as.integer(rbinom(n = N*k, size = 1,
     prob = p1)), nrow = k, byrow = TRUE)
  misidt <- rbinom(n = N, size = 1, prob = p2)
  arrival = matrix(ncol = N, rexp(N* k))
  serving = matrix(ncol = N, rexp(N* k))
  tpos = if (pr) sample.int(k, size = N, replace = TRUE)
     else rep(k+1, N)
  result <- queue(miat, mst, c, pr, L, misidd, misidt,
     arrival, serving, tpos, old = old)
  simRT <- result[ , 4]
  if (TRUE) {
    WMresult <- LqDist(simRT + Tres, empRT, 1)
  } else {
    histemp <- data2hist(as.numeric(empRT))
```

```
    histsim <- data2hist(as.numeric(simRT + Tres))
    WMresult <- WassSqDistH(histemp, histsim)
  }
  return(WMresult)
}

LqDist <- function(E1, E2, q = 1) {
  E1 <- sort(E1)
  E2 <- sort(E2)
  .E1 <<- E1
  .E2 <<- E2
  .Call(C_LqDist, as.double(E1), as.double(E2), as.double(
    q))
}
```

D. Derivatives of the log-likelihood function

This appendix displays the critical steps in calculating the partial derivatives and the gradient of $\widetilde{l}(\alpha, \beta, a_1, a_2, b)$.

First, the additive terms in $l(\pi_{md,k_1}, \pi_{fa,k_1}, \ldots, \pi_{md,k_m}, \pi_{fa,k_m})$ are derived with respect to $(\alpha, \beta, a_1, a_2, b)$. Due to the similar structure of $\pi_{md,k_1}, \ldots, \pi_{md,k_m}$, it is sufficient to determine the partial derivatives of $\ln(\pi_{md})$ and $\ln(1 - \pi_{md})$. The same applies to $\pi_{fa,k_1}, \ldots, \pi_{fa,k_m}$.

$$\frac{\mathrm{d}}{\mathrm{d}\alpha} \ln(\pi_{md}) = -k \exp\big(\alpha + \beta k\big) \ln\Big(1 - a_1 k^{-b}\Big) + \frac{(1 - a_1 k^{-b}) \exp\big(\alpha + \beta k\big) - a_2 k^b \exp\big(\alpha + \beta k\big)}{(1 - a_1 k^{-b}) \exp\big(\alpha + \beta k\big) + a_2 k^b (1 - \exp\big(\alpha + \beta k\big))}$$

$$\frac{\mathrm{d}}{\mathrm{d}\alpha} \ln(1 - \pi_{md}) = 1 - k \exp\big(\alpha + \beta k\big) \ln\Big(1 - a_1 k^{-b}\Big) + \frac{(k \exp\big(\alpha + \beta k\big) \ln\big(1 - a_1 k^{-b}\big) - 1)(1 - a_1 k^{-b})^{-(k - k \exp(\alpha + \beta k) - 1)} + a_2 k^b}{(1 - a_1 k^{-b})^{-(k - k \exp(\alpha + \beta k) - 1)} - (1 - a_1 k^{-b}) \exp\big(\alpha + \beta k\big) - a_2 k^b (1 - \exp\big(\alpha + \beta k\big))}$$

$$\frac{\mathrm{d}}{\mathrm{d}\alpha} \ln\big(\pi_{fa}\big) = \frac{(1 - a_1 k^{-b})^{k - k \exp(\alpha + \beta k)}}{1 - (1 - a_1 k^{-b})^{k - k \exp(\alpha + \beta k)}} \cdot k \exp\big(\alpha + \beta k\big) \ln\Big(1 - a_1 k^{-b}\Big)$$

$$\frac{\mathrm{d}}{\mathrm{d}\alpha} \ln\big(1 - \pi_{fa}\big) = -k \exp\big(\alpha + \beta k\big) \ln\Big(1 - a_1 k^{-b}\Big)$$

$$\Rightarrow \frac{\mathrm{d}}{\mathrm{d}\alpha}\widetilde{l}(\alpha,\beta,a_1,a_2,b) = \sum_{i=1}^{m} A_i \text{ with}$$

$$\begin{aligned}
A_i = &\left(n - \frac{y_{tp,k_i}}{1-\pi_{md,k_i}}\right)\left(1 - k_i \exp\left(\alpha+\beta k_i\right)\ln\left(1 - a_1 k_i^{-b}\right)\right.\\
&\left. - \frac{a_2 k_i^b}{(1-a_1 k_i^{-b})\exp\left(\alpha+\beta k_i\right) + a_2 k_i^b (1-\exp\left(\alpha+\beta k_i\right))}\right)\\
&+ \left(\frac{y_{ta,k_i}}{\pi_{fa,k_i}} - n\right) k_i \exp\left(\alpha+\beta k_i\right)\ln\left(1 - a_1 k_i^{-b}\right)\\
&= \left(n - y_{tp,k_i}\left(1 - (1-a_1 k_i^{-b})^{k_i - k_i \exp\left(\alpha+\beta k_i\right)}\exp\left(\alpha+\beta k_i\right)\right.\right.\\
&\left.\left. - a_2 k_i^b (1-a_1 k_i^{-b})^{k_i - k_i\exp\left(\alpha+\beta k_i\right)-1}(1-\exp\left(\alpha+\beta k_i\right))\right)^{-1}\right)\\
&\cdot \left(1 - k_i \exp\left(\alpha+\beta k_i\right)\ln\left(1 - a_1 k_i^{-b}\right)\right.\\
&\left. - a_2 k_i^b \left((1-a_1 k_i^{-b})\exp\left(\alpha+\beta k_i\right) + a_2 k_i^b (1-\exp\left(\alpha+\beta k_i\right))\right)^{-1}\right)\\
&+ \left(y_{ta,k_i}\left(1 - (1-a_1 k_i^{-b})^{k_i - k_i\exp\left(\alpha+\beta k_i\right)}\right)^{-1} - n\right) k_i \exp\left(\alpha+\beta k_i\right)\ln\left(1 - a_1 k_i^{-b}\right)
\end{aligned}$$

$$\frac{\mathrm{d}}{\mathrm{d}\beta}\widetilde{l}(\alpha,\beta,a_1,a_2,b) = \sum_{i=1}^{m} k_i A_i$$

$$\frac{\mathrm{d}}{\mathrm{d}a_1}\widetilde{l}(\alpha,\beta,a_1,a_2,b) = \sum_{i=1}^{m} k_i^{-b} B_i \text{ with}$$

$$\begin{aligned}
B_i = &\left(\frac{y_{tp,k_i}}{\pi_{md,k_i}(1-\pi_{md,k_i})} - \frac{n}{\pi_{md,k_i}}\right)\cdot\left(\left(k_i - k_i\exp\left(\alpha+\beta k_i\right)\right)\right.\\
&(1-a_1 k_i^{-b})^{k_i - k_i\exp\left(\alpha+\beta k_i\right)-1}\exp\left(\alpha+\beta k_i\right)\\
&\left. + a_2 k_i^b (k_i - k_i\exp\left(\alpha+\beta k_i\right) - 1)(1-a_1 k_i^{-b})^{k_i - k_i\exp\left(\alpha+\beta k_i\right)-2}(1-\exp\left(\alpha+\beta k_i\right))\right)\\
&+ \left(\frac{y_{ta,k_i}}{\pi_{fa,k_i}} - n\right)\frac{k_i - k_i\exp\left(\alpha+\beta k_i\right)}{1-a_1 k_i^{-b}}
\end{aligned}$$

$$\frac{\mathrm{d}}{\mathrm{d}a_2}\widetilde{l}(\alpha,\beta,a_1,a_2,b) =$$
$$\sum_{i=1}^{m} k_i^b\left(\frac{n}{\pi_{md,k_i}} - \frac{y_{tp,k_i}}{\pi_{md,k_i}(1-\pi_{md,k_i})}\right)(1-a_1k_i^{-b})^{k_i-k_i\exp(\alpha+\beta k_i)-1}(1-\exp(\alpha+\beta k_i))$$

$$\frac{\mathrm{d}}{\mathrm{d}b}\widetilde{l}(\alpha,\beta,a_1,a_2,b) = \sum_{i=1}^{m}\ln(k_i)C_i + a_1k_i^{-b}\ln(k_i)D_i \text{ with}$$

$$\begin{aligned}
C_i = &\left(\frac{n}{\pi_{md,k_i}} - \frac{y_{tp,k_i}}{\pi_{md,k_i}(1-\pi_{md,k_i})}\right)\Big(a_1k^{-b}\exp(\alpha+\beta k_i)(k_i - k_i\exp(\alpha+\beta k_i)) \\
&(1-a_1k_i^{-b})^{k_i-k_i\exp(\alpha+\beta k_i)-1} \\
&+ a_2k_i^b(1-a_1k_i^{-b})^{k_i-k_i\exp(\alpha+\beta k_i)-1}(1-\exp(\alpha+\beta k_i)) \\
&+ a_1a_2(1-\exp(\alpha+\beta k_i))(k_i - k_i\exp(\alpha+\beta k_i) - 1)(1-a_1k_i^{-b})^{k_i-k_i\exp(\alpha+\beta k_i)-2}\Big) \\
D_i = &\left(\frac{n_{ta,k_i}}{1-a_1k_i^{-b}} - \frac{y_{ta,k_i}}{\pi_{fa,k_i}}\right. \\
&\left.(1-a_1k_i^{-b})^{k_i-k_i\exp(\alpha+\beta k_i)-1}\right)(k_i - k_i\exp(\alpha+\beta k_i))
\end{aligned}$$

E. Supplementary information on the profiling technique

This appendix supplements the description in Section 6.1.2. It explains the profiling technique used in this dissertation in detail.

The approach of profiling used in this dissertation is derivative of the profile likelihood method commonly applied to deal with nuisance parameters[2] (see e.g. Barndorff-Nielsen & Cox, 1994; Murphy & van der Vaart, 2000; Patefield, 1977). The principle of profile likelihood is to profile out the nuisance parameters by replacing them with their "optimal value given the parameter of interest".

Suppose the parameter vector is partitioned into (θ, η), where θ denotes the subset of parameters of interest and η the nuisance parameters. We can maximize the likelihood $L(\theta, \eta)$ over η for each fixed θ, denoting the maximum-likelihood estimate of η given θ by $\hat{\eta}(\theta)$. The profile likelihood is then obtained by substituting η with the optimal value of η as a function of θ: $L_{profile}(\theta) = L(\theta, \hat{\eta}(\theta)) = \sup_{\eta} L(\theta, \eta)$. The profile likelihood function can be understood as essentially the projection of the profile of the full likelihood function onto the θ axes, through which the relevant information of the likelihood regarding θ is retained (Patefield, 1977). Most importantly, the value of θ that maximizes the profile likelihood equals its maximum likelihood estimate $\hat{\theta}$.

The profiling technique used here can be seen as profiling the likelihood on a lower level of resolution. Instead of calculating the curvature of the profile likelihood function for every fixed $\tilde{\beta}$, only snippets of the curvatures will be visible to the modeler – for a $\tilde{\beta}$ in each interval where a local optimum appears. Nevertheless, the search in a small interval has implicitly explored the curvatures of the profile likelihood for those $\tilde{\beta}$ within the interval. The profile likelihood method was not applied directly for two reasons. First, all parameters in this model are of equal interest, so that a reduction of parameters is neither necessary nor desired. Second, the profile likelihood function with regard to $\tilde{\beta}$ does not have an explicit analytical form, such that numerical approximation would have been necessary anyway, if the profile likelihood method was applied strictly.

[2] Parameters which are not of primary interest but need to be estimated.

F. R code for the estimation of accuracy-related parameters using MLE

```
## source("estimating_accpar.R")
## estimating accuracy-related parameters using MLE

#----- Define functions -----

# log-likelihood function
# par = (alpha, betatilde, a1, a2, btilde), beta =
   betatilde*(-alpha/max(k)), b = btilde*(-log(a2)/log(max
   (k)))
loglikeli <- function(par, errordata) {
  k <- errordata$set_size
  md <- errordata$miss_detection
  cr <- errordata$correct_rejection
  fa <- errordata$false_alarm
  hit <- errordata$hit
  alpha <- par[1]
  beta <- par[2]*(-par[1]/max(k))
  a1 <- par[3]
  a2 <- par[4]
  b <- par[5]*(-log(a2)/log(max(k)))
  cv <- exp(alpha+beta*k)
  lk <- k*(1-cv)
  p1 <- a1*k^(-b)
  p2 <- a2*k^b
  mp1 <- 1-p1
  l1mp1 <- log(mp1)
  loghit <- (lk-1)*l1mp1+log(mp1^(-lk+1)-mp1*cv-p2*(1-cv))
  logmd <- (lk-1)*l1mp1+log(mp1*cv+p2*(1-cv))
  logcr <- lk*l1mp1
  logfa <- log(1-mp1^lk)
```

```
  ans <- sum(hit*loghit + md*logmd + cr*logcr + fa*logfa)
  if(is.nan(ans)) stop(Print(k,par,cv,p1,p2,ans,alpha,beta
    ,hit,loghit,md,logmd,cr,logcr,fa,logfa,mp1,mp1^k,lk,
    a1,a2))
  ans
}

# gradient of the log-likelihoodfunction
gr <- function(par, errordata) {
  k <- errordata$set_size
  hit <- errordata$hit
  md <- errordata$miss_detection
  cr <- errordata$correct_rejection
  fa <- errordata$false_alarm
  alpha <- par[1]
  tbeta <- -par[1]/max(k)
  beta <- par[2]*tbeta
  a1 <- par[3]
  a2 <- par[4]
  tb <- -log(a2)/log(max(k))
  b <- par[5]*tb
  cv <- exp(alpha+beta*k)
  lk <- k*(1-cv)
  p1 <- a1*k^(-b)
  p2 <- a2*k^b
  fgm1 <- (1-p1)^(-lk+1)
  fgm2 <- log(1-p1)*cv
  fgm3 <- p2*cv
  fgm4 <- (1-p1)*cv
  fgm5 <- p2*(1-cv)
  fgm6 <- k^b
  fgm7 <- ((-lk+1)*(1-p1)^(-lk))/k^b
  fgm8 <- (lk-1)/(k^b*(1-p1))
```

```
fgm9 <- fgm6*(1-cv)
fgm10 <- a1*log(k)
fgm11 <- fgm5*log(k)
fgm12 <- k*(1-p1)^lk*log(1-p1)*cv
fgm13 <- 1-(1-p1)^lk
fgm14 <- lk*(1-p1)^(lk-1)
fgm15 <- -par[5]/(a2*log(max(k)))
dloghitdalpha <- (k*fgm1*fgm2+fgm3-fgm4)/(-fgm4-fgm5+
    fgm1)-k*fgm2
dloghitdbeta <- k*dloghitdalpha*tbeta
dloghitda1 <- (cv/fgm6-fgm7)/(fgm1-fgm4-fgm5)-fgm8
dloghitda2 <- ((fgm10*fgm7-fgm10*cv/fgm6)*fgm15-fgm9+
    fgm9*par[5]*log(k)/log(max(k)))/(-fgm4-fgm5+fgm1)+
    fgm10*fgm8*fgm15
dloghitdb <- ((fgm10*fgm7-fgm10*cv/fgm6-fgm11)/(-fgm4-
    fgm5+fgm1)+fgm10*fgm8)*tb
dlogmddalpha <- (fgm4-fgm3)/(fgm4+fgm5)-k*fgm2
dlogmddbeta <- k*dlogmddalpha*tbeta
dlogmdda1 <- -cv/(fgm6*(fgm4+fgm5))-fgm8
dlogmdda2 <- ((p1*log(k)*cv+fgm11)*fgm15+fgm9)/(fgm4+
    fgm5)+fgm10*fgm8*fgm15
dlogmddb <- ((p1*log(k)*cv+fgm11)/(fgm4+fgm5)+fgm10*fgm8
    )*tb
dlogcrdalpha <- -k*fgm2
dlogcrdbeta <- k*dlogcrdalpha*tbeta
dlogcrda1 <- lk/(a1-k^b)
dlogcrda2 <- -fgm10*dlogcrda1*fgm15
dlogcrdb <- -fgm10*dlogcrda1*tb
dlogfadalpha <- fgm12/fgm13
dlogfadbeta <- k*dlogfadalpha*tbeta
dlogfada1 <- fgm14/(fgm6*fgm13)
dlogfada2 <- (-(fgm10*fgm14)/(fgm6*fgm13))*fgm15
dlogfadb <- (-(fgm10*fgm14)/(fgm6*fgm13))*tb
```

```
  c(sum(c(hit*dloghitdalpha, md*dlogmddalpha, cr*
     dlogcrdalpha, fa*dlogfadalpha)*(1-k*par[2]/max(k))),
     sum(hit*dloghitdbeta + md*dlogmddbeta + cr*
     dlogcrdbeta + fa*dlogfadbeta),
    sum(hit*dloghitda1 + md*dlogmdda1 + cr*dlogcrda1 + fa*
       dlogfada1), sum(hit*dloghitda2 + md*dlogmdda2 + cr*
       dlogcrda2 + fa*dlogfada2),
    sum(hit*dloghitdb + md*dlogmddb + cr*dlogcrdb + fa*
       dlogfadb))
}

# transformation from new parameterization to old
   parameterization
ntoo <- function(par, errordata) {
  k <- errordata$set_size
  tbeta <- -par[1]/max(k)
  tb <- -log(par[4])/log(max(k))
  par*c(1, tbeta, 1, 1, tb)
}

# trasformation from old parameterization to new
   parameterization
oton <- function(opar, errordata) {
  k <- errordata$set_size
  revtrbeta <- -max(k)/opar[1]
  revtrb <- -log(max(k))/log(opar[4])
  opar*c(1, revtrbeta, 1, 1, revtrb)
}

# calculating model predictions
pfa <- function(par, errordata) {
  alpha <- par[1]
  beta <- par[2]
```

```
  a1 <- par[3]
  a2 <- par[4]
  b <- par[5]
  k <- errordata$set_size
  1-(1-a1*k^(-b))^(k-k*exp(alpha + beta*k))
}
pmd <- function(par, errordata) {
  alpha <- par[1]
  beta <- par[2]
  a1 <- par[3]
  a2 <- par[4]
  b <- par[5]
  k <- errordata$set_size
  (1-a1*k^(-b))^(k-k*exp(alpha + beta*k)-1)*((1-a1*k^(-b))
     *exp(alpha+beta*k) + a2*k^b*(1-exp(alpha+beta*k)))
}

#----- Estimation procedure -----
errordata <- data.frame("set_size" = setsize, "hit" = rep(
   NA, length(setsize)), "miss_detection" = rep(NA, length
   (setsize)), "correct_rejection" = rep(NA, length(
   setsize)), "false_alarm" = rep(NA, length(setsize)))
for (i in 1:length(setsize)) {
    errordata$hit[i] <- length(get(name_data_p[i])[,2][get
       (name_data_p[i])[,2] == 1])
    errordata$miss_detection[i] <- length(get(name_data_p[
       i])[,2][get(name_data_p[i])[,2] == 0])
    errordata$correct_rejection[i] <- length(get(name_data
       _a[i])[,2][get(name_data_a[i])[,2] == 0])
    errordata$false_alarm[i] <- length(get(name_data_a[i])
       [,2][get(name_data_a[i])[,2] == 1])
}
bound <- seq(10^(-6), 1-10^(-6), len = 11)
```

```
midp <- bound[1:10] + (bound[-1] - bound[1:10])/2
errorresults <- vector('list', 10)
for (i in 1:10) {
    errorresults[[i]] <- optim(par = c(-3.5, midp[i], 0.5,
        0.5, 0.5), fn = loglikeli, gr=gr, errordata=
       errordata, method = 'L-BFGS-B', lower = c(-7, bound
       [i], 0.001, 0.00001, 0.001), upper = c(-0.001,
       bound[i+1], 0.999, 0.999, 0.999), control=list(
       fnscale = -1, parscale = c(4.4, 0.45, 0.019,
       0.0002, 0.3732)))
}
value <- sapply(errorresults, function(a) a$value)
index <- which(value == max(value))
esterrorpar <- ntoo(errorresults[[index]]$par, errordata)
maxvalue <- value[index]
```

G. MDE based on distribution functions as the method of choice

In this appendix, reasons for the choice of MDE based on distribution functions are expounded, following the discussion in Section 6.2.1.

Selecting the parametric model that minimizes the closeness between the data and the model is one of the most natural idea in statistics and numerical analysis. Although it seems unambiguous what is referred to by the name, the term minimum distance estimation is applicable to a very extensive scope of methods. The generality lies in the numerous ways of measuring the closeness. When applying an MDE approach, the first choice a modeler faces is the type of the distance. There are three (equivalent) ways of characterizing a random variable: by the density, the distribution function or the characteristic function. All of them are mathematical objects a distance can based upon (see e.g. Basu et al., 2010; Parr, 1981). In practice, density-based and distribution-function-based distances are commonly used. Within each type, there are many options for constructing a distance (see Basu et al., 2010, for details).

As distinct from simpler parametric models, an explicit analytical PDF or likelihood function of the queueing model is not available, so that statistical inference is based on Monte Carlo simulation. In order to apply likelihood methods (e.g., quasi-maximum likelihood estimate), a non-parametric smoothing technique such as kernel density estimation must be applied first to the simulated data to obtain the estimated likelihood (quasi-likelihood), which will be evaluated at the locations of the empirical data points. If an explicit analytical likelihood function or PDF is available, the smoothing technique is not necessary for calculating the likelihood given the empirical data. In contrast, distances between two CDFs can be calculated directly without any smoothing techniques, regardless of whether neither, one or both of the CDFs are empirical distribution function (ECDF)[3]. Merely one simple step of computation is required to obtain a measure of distribution-function-based distance given two sets of data points, which

[3]The term "empirical distribution function" is used here and henceforth in statistical sense, referring to the distribution function associated with the empirical measure of a sample of realizations/observations. The sample is not necessarily collected in empirical studies.

is easy to implement. Routines implementing smoothing techniques are not necessary because the ECDF itself is an estimate of the theoretical CDF from which the sample of data points arise[4]. Hence, for every single iteration of the optimization, an MDE approach based on CDF requires one step of computation less in comparison to a likelihood method . Although the gain in computation time depends on the concrete application case and the implementation, a lower computational cost by using the method specified here is to expect in general.

A robust estimator is resistant to slight violations of model assumptions. It is nearly fully efficient under ideal conditions (i.e., when the model is correct) and still "estimates a meaningful quantity with reasonable efficiency" (Parr & De Wet, 1981, p.178) under moderate deviations from the model (i.e., the specified model is incorrect but close to the correct model). Robustness is also reflected in the fact that the estimator is less sensitive to outliers. This property is particularly valuable for Monte Carlo simulation-based parameter estimation procedures because a small misspecification of the model is inherent in such procedures, e.g., the theoretical CDF is approximated by the ECDF of simulated data. Furthermore, since the model is complex, some of the assumptions may be met only approximately. Outliers are also very likely since the empirical data are aggregated across individuals. Whereas robustness is a general property present for MDE (e.g., Basu et al., 2010; Boos, 1981, 1982; Parr, 1981; Parr & De Wet, 1981; Parr & Schucany, 1980), it is not always shared by MLE. Since the estimated density using smoothing technique is a quasi-likelihood, the estimation may not perform well (i.e., desirable statistical properties that are present under optimal condition may not given) even if the model is essentially correct. On the other hand, the feature of asymptotic efficiency which is generally present for MLE (under some regularity conditions) does appear to be shared generally by MDE based on the distribution function approach. However, as Boos (1981, 1982) pointed out, asymptotic efficiency of distribution-function-based MDE can be achieved by constructing appropriate distances. In general, reasonable efficiency

[4]The ECDF is not the only estimate of the CDF. Non-parametric smoothing techniques can be applied to obtain a smooth estimator for the CDF. Smooth estimators obtained in this way have both advantages and disadvantages compared to the ECDF. However, these aspects are of little relevance for the purpose of the current application. The statistical properties of the ECDF as estimate of the CDF are sufficient.

properties are given for MDE and it possesses other desirable properties such as strong consistency (Parr & Schucany, 1980). As robustness is the more important concern in the current case, I consider MDE as the more advantageous approach.

Although distances between PDFs, such as the Hellinger distance and the chi-square distance, can also be calculated based histogram data directly and their MDE share the desirable statistical properties, they have two major drawbacks compared to distances between CDFs given the current context. On the one hand, they are insensitive to the extent of the translation. Since the residual time is modeled as an additive constant captured by a translation parameter T_{res}, insensitiveness to translation is a disadvantage for estimating T_{res} accurately. On the other hand, these distances are more sensitive to small wiggles in the distribution than the distances between CDFs. Since both the empirical and the simulated data are noisy (especially the empirical data aggregated over individuals), MDE approach based on the distribution function is more likely to yield generalizable results.

H. A comparison of the distance measures

In this appendix, reasons for the choice of the Wasserstein metric as the distance measure are expounded, following the discussion in Section 6.2.1.

Let F_θ denote parametric model CDF with parameter θ and G_n denote the empirical cumulative distribution function (ECDF) of the empirically observed sample of data with sample size n. The Kolmogorov-Smirnov distance is defined as the largest pointwise absolute difference between the two CDFs:

$$D(G_n, F_\theta) = \sup_{x \in \mathbb{R}} |G_n(x) - F_\theta(x)|.$$

The Cramér-von Mises distance, defined as

$$\omega^2(G_n, F_\theta) = \int_{-\infty}^{\infty} (G_n(x) - F_\theta(x))^2 \, \mathrm{d}F_\theta(x) \tag{8.3}$$

calculates the expected value of the squared difference. Weighted version of Kolmogorov-Smirnov distance or Cramér-von Mises distance involves a weight function.

A further distance measure that is less frequently applied and studied in the context of MDE is the Wasserstein metric. It is defined as a metric of probability measures on a metric space. The notion originates from Optimal Transport theory and the Wasserstein metric is interpreted as the minimum amount of "energy" or "cost" of "transporting" the "mass" of a probability distribution on another The formal definition of the Wasserstein metric is very abstract and therefore intractable for calculation in concrete numerical application directly[5]. Vallender (1974) proofed that for probability measures on $\mathbb{R}$ with the Euclidean metric (i.e., in one-dimensional case and for $p = 1$), the Wasserstein metric of order 1 between them is equal to

$$W_1(G_n, F_\theta) = \int_0^1 \left| {G_n}^{-1}(y) - {F_\theta}^{-1}(y) \right| \mathrm{d}y = \int_{-\infty}^{\infty} |G_n(x) - F_\theta(x)| \, \mathrm{d}x, \tag{8.4}$$

[5]For $p \in [1, \infty)$, the Wasserstein distance of order p between two probability measures P, Q on $\mathbb{R}$ with finite p-moments is defined as

$$W_p(P, Q) := \inf_{\pi \in \Pi(P,Q)} \left(\int_{\mathbb{R}\times\mathbb{R}} \|x - y\|^p \, \mathrm{d}\pi(x, y) \right)^{1/p}$$

, where $\Pi(P, Q)$ is the set of all probability measures on $\mathbb{R} \times \mathbb{R}$ with marginal distributions P and Q. (see e.g., Ramdas et al., 2017; Sommerfeld, 2017).

where G_n and F_θ are the corresponding distribution functions and ${G_n}^{-1}$ and ${F_\theta}^{-1}$ the corresponding quantile functions. Geometrically, the Wasserstein metric of order 1 in one-dimensional case calculates the area between the functions G_n and F_θ, which is equal to the area between ${G_n}^{-1}$ and ${F_\theta}^{-1}$. Note that by this equivalent closed form, a connection between the Wasserstein metric and the Kolmogorov-Smirnov distance becomes apparent: The former is the L^1 norm of the difference of the CDFs, whereas the latter is the L^∞ norm (essential supremum) of the same magnitude. The closed form involving distribution functions (or quantile functions) makes it tractable to calculate the Wasserstein metric in application cases where the (approximate) distribution function (or the quantile function) is known, thus also applicable in the current case. A more general result is that the Wasserstein distance of order p between two probability measures on $\mathbb{R}$ is equal to the L^p norm of the difference of the corresponding quantile functions:

$$W_p(P, Q) = \left(\int_0^1 \left| F^{-1}(y) - G^{-1}(y) \right|^p \mathrm{d}y \right)^{1/p},$$

see e.g., Ramdas et al. (2017) for a proof[6]. Therefore, in one-dimensional case, the Wasserstein distance essentially quantifies the information a Q-Q plot illustrates. The order p has an influence on how much outliers are weighted. For larger p, outliers have larger weights on the distance. Although the Wasserstein metric of order 1 in one-dimensional case possesses intuitive accessibility and simple interpretation, it is less popular than the Kolmogorov-Smirnov distance and the Cramér-von Mises distance as a distance measure in the MDE approach. This may be in part owing to the fact that it is usually difficult to evaluate integrands involving absolute value in most application cases. This difficulty, however, is absent in the current case because the two functions involved are both step functions as ECDF.

These three distance measures have different properties which may be beneficial or obstructive in different aspects for a specific application case. The

[6]However, a relation analogous to the second equality in Equation (8.4) does not exist for $p \geq 2$ in general. The reason is that the integral along the x-axis does not necessarily match the integral along the y-axis for $p \geq 2$. A simple example for $p = 2$ is two Dirac measures δ_1 and δ_3. The L^2 norm of the difference between the associated quantile functions is 2, whereas the L^2 norm of the difference between the associated distribution functions is $\sqrt{2}$.

first aspect I consider important for the current case is the sensitivity in detecting various types of deviations. Among these three distance measures, the Kolmogorov-Smirnov distance is sensitive to local deformations and less sensitive to (total or partial) translations (the insensitivity becomes more obvious from certain extent of translation on due to the fact that Kolmogorov-Smirnov distance takes values between 0 and 1). On the contrary, the Wasserstein metric is less sensitive to local deformations and sensitive to (total or partial) translations. The Cramér-von Mises distance has a medium sensitivity to local deformations and translations among these three distance measures. A high sensitivity to local deformations is undesirable in the current case. Behavioral data is usually subject to inexplicable noise which causes small, local distortions in the pattern. Even if inexplicable noise in the data is negligible, the data can exhibit data-specific characteristics which manifest in local deformations. With a distance measure that is too sensitive for such distortions, the optimization algorithm is more likely to work in favor of parameter values that are able to explain such local deformations rather than the overall pattern. In other words, the risk of overfitting is higher for such distance measures. In contrast, a high sensitivity to translations is desirable in the current context. Since the residual time is modeled as an additive constant, meaning that the translation of the random variable that represents the residual time is a model parameter to estimate. Although this parameter is not of primary interest, the accuracy of its estimation can influence the accuracy of the estimations of other parameters (see Section 6.2.3 for detailed discussion on this issue). If the distance measure is less sensitive to translations, the residual time cannot be estimated accurately, which can lead to identifiability issues. In addition, the optimization based on Kolmogorov-Smirnov distance can be numerically more difficult to handle (?) since the value is limited between 0 and 1. This is especially the case for complex models with many parameters. The Kolmogorov-Smirnov distance may not be able to reflect the changes in the pattern of the model prediction due to changes in parameters in a differentiated way. This can cause identifiability issues as well (cf. Section 6.2.3).

The second aspect I considered important for the choice of the distance measure is the computational cost. The importance of the computational efficiency has been elucidated in Section 5.1.3. In general, the Cramér-von Mises distance

is computationally more costly to implement compared to the other two distance measures because it requires the calculation of the integral of the squared difference with respect to the Lebesgue-Stieltjes measure associated to the "baseline" CDF of F_θ (here the CDF of the model prediction), as is apparent from Equation (8.3). As mentioned above, computing the Wasserstein metric in the current case is extricated from the difficulty of evaluating integrands involving absolute value in general application, since both functions are step functions.

I. R code for the estimation of RT-related parameters using MDE

```
## source("required_functions_RTpar.R")
## functions necessary for estimating RT-related
   parameters using MDE

#----- Define functions -----

# function that picks out only correct responses to
   calculate distance. par = c(miat, mst, Tres)
WMonlycorrect <- function(par, esterrorpar, c, k, pr, N,
   empRT, empresp, old = FALSE) {
  set.seed(0)
  miat <- par[1]
  mst <- par[2]
  Tres <- par[3]
  alpha <- esterrorpar[1]
  beta <- esterrorpar[2]
  a1 <- esterrorpar[3]
  a2 <- esterrorpar[4]
  b <- esterrorpar[5]
  lk <- k*(1 - exp(alpha + beta*k))
  L <- floor(lk) + rbinom(n = N, size = 1, prob = lk -
     floor(lk))
  p1 <- a1*k^(-b)
  p2 <- a2*k^b
  misidd <- matrix(as.integer(rbinom(n = N*k, size = 1,
     prob = p1)), nrow = k, byrow = TRUE)
  misidt <- rbinom(n = N, size = 1, prob = p2)
  arrival = matrix(ncol = N, rexp(N * k))
  serving = matrix(ncol = N, rexp(N * k))
```

```
  tpos = if (pr) sample.int(k, size = N, replace = TRUE)
    else rep(k + 1, N)
  result <- queue(miat, mst, c, pr, L, misidd, misidt,
    arrival, serving, tpos, old = old)
  simRT <- result[ , 4]
  simresp <- result[ , 5]
  WMresult <- LqDist(simRT[simresp==pr] + Tres, empRT[
    empresp==pr], 1)
  return(WMresult)
}

# function that fixes Tresy and estimates (miat, mst,
  Tresn). par = c(miat, mst, Tresn)
WMfixTresy <- function(par123, Tresy, esterrorpar, c, k,
  pr, N, empRT, empresp, old = FALSE) {
  WMdiffrespshiftweight(par = c(par123, Tresy),
    esterrorpar = esterrorpar, c = c, k = k, pr = pr, N =
     N, empRT = empRT, empresp = empresp, old = old)
}

# function that fixes Tresn and estimates (miat, mst,
  Tresy). par = c(miat, mst, Tresy)
WMfixTresn <- function(par124, Tresn, esterrorpar, c, k,
  pr, N, empRT, empresp, old=FALSE) {
  WMdiffrespshiftweight(par = c(par124[1:2], Tresn, par124
    [3]), esterrorpar = esterrorpar, c = c, k = k, pr =
    pr, N = N, empRT = empRT, empresp = empresp, old =
    old)
}

# functions that extract estimated parameters or value
  from the list
extractpar <- function(a) a$par
```

```
extractvalue <- function(a) a$value

sim.ny <- function(par, esterrorpar, c, k, N, pr, seed) {
  set.seed(seed)
  miat <- par[1]
  mst <- par[2]
  Tresn <- par[3]
  Tresy <- par[4]
  alpha <- esterrorpar[1]
  beta <- esterrorpar[2]
  a1 <- esterrorpar[3]
  a2 <- esterrorpar[4]
  b <- esterrorpar[5]
  lk <- k*(1 - exp(alpha + beta*k))
  L <- floor(lk) + rbinom(n = N, size = 1, prob = lk-floor
     (lk))
  p1 <- a1*k^(-b)
  p2 <- a2*k^b
  misidd <- matrix(as.integer(rbinom(n = N*k, size = 1,
     prob = p1)), nrow = k, byrow = TRUE)
  misidt <- rbinom(n = N, size = 1, prob = p2)
  arrival <- matrix(ncol = N, rexp(N* k))
  serving <- matrix(ncol = N, rexp(N* k))
  tpos <- if (pr) sample.int(k, size = N, replace=TRUE)
     else rep(k+1, N)
  simdata <- queue(miat=miat, mst=mst, c=c, pr=pr, L=L,
     misidd = misidd, misidt = misidt, arrival = arrival,
     serving = serving, tposition = tpos, old = FALSE)
  simRT <- simdata[,4] + Tresn + simdata[,5]*(Tresy-Tresn)
  simoutput <- cbind(simRT, simdata[,5])
  colnames(simoutput) <- c('simRT','simresponse')
  return(simoutput)
}
```

```
## Functions for set-size-invariant mu

# For all data sets (with any number of set size levels).
   Function that fixes miat and mst and estimates (Tresn,
   Tresy), whereby k is a vector containing all set size
   levels and empRT and empresp the RTs and responses for
   all set size levels as list element.
WMfixqtimes.all <- function(par34, miat, mst, esterrorpar,
    c, k, N, empRTa, empRTp, emprespa, emprespp, Wa, Wp,
   old = FALSE) {
    dist_a <- rep(NA, length(k))
    dist_p <- rep(NA, length(k))
    for (i in 1:length(k)) {
        dist_a[i] <- WMdiffrespshiftweight(par = c(miat,
           mst, par34), esterrorpar = esterrorpar, c = c,
           k = k[i], pr = FALSE, N = N, empRT = empRTa[[i
           ]], empresp = emprespa[[i]], old = old) * Wa[i]
        dist_p[i] <- WMdiffrespshiftweight(par = c(miat,
           mst, par34), esterrorpar = esterrorpar, c = c,
           k = k[i], pr = TRUE, N = N, empRT = empRTp[[i
           ]], empresp = emprespp[[i]], old = old) * Wp[i]
    }
    return(sum(dist_a + dist_p))
}

# For all data sets (with any number of set size levels).
   Function that fixes Tresn and Tresy and estimates (miat
   , mst), whereby k is a vector containing all set size
   levels and empRT and empresp the RTs and responses for
   all set size levels as list element.
WMfixTres.all <- function(par12, Tresn, Tresy, esterrorpar
```

```
    , c, k, N, empRTa, empRTp, emprespa, emprespp, Wa, Wp,
    old = FALSE) {
      dist_a <- rep(NA, length(k))
      dist_p <- rep(NA, length(k))
      for (i in 1:length(k)) {
          dist_a[i] <- WMdiffrespshiftweight(par = c(par12,
             Tresn, Tresy), esterrorpar = esterrorpar, c = c
             , k = k[i], pr = FALSE, N = N, empRT = empRTa[[
             i]], empresp = emprespa[[i]], old = old)*Wa[i]
          dist_p[i] <- WMdiffrespshiftweight(par = c(par12,
             Tresn, Tresy), esterrorpar = esterrorpar, c = c
             , k = k[i], pr = TRUE, N = N, empRT = empRTp[[i
             ]], empresp = emprespp[[i]], old = old)*Wp[i]
      }
      return(sum(dist_a + dist_p))
}

ind.dist <- function(par, esterrorpar, c, k, N, empRTa,
    empRTp, emprespa, emprespp, Wa, Wp, old = FALSE) {
      dist_a <- rep(NA, length(k))
      dist_p <- rep(NA, length(k))
      for (i in 1:length(k)) {
          dist_a[i] <- WMdiffrespshiftweight(par = par,
             esterrorpar = esterrorpar, c = c, k = k[i], pr
             = FALSE, N = N, empRT = empRTa[[i]], empresp =
             emprespa[[i]], old = old)
          dist_p[i] <- WMdiffrespshiftweight(par = par,
             esterrorpar = esterrorpar, c = c, k = k[i], pr
             = TRUE, N = N, empRT = empRTp[[i]], empresp =
             emprespp[[i]], old = old)
      }
      wdist_a <- dist_a * Wa
      wdist_p <- dist_p * Wa
```

```
    display <- rbind(dist_a, wdist_a, dist_p, wdist_p)
    display <- cbind(display, rowSums(display))
    rownames(display) <- c('absent', 'absent_relative', '
       present', 'present_relative')
    colnames(display) <- c(k, 'sum')
    return(display)
}


## Functions for set-size-dependent mu

# For pair of data sets with the same set size level.
   Function that fixes miat, Tresn and Tresy and estimates
    mst
WMfixpar134.sdmst <- function(par2, miat, Tresn, Tresy,
   esterrorpar, c, k, N, empRTa, empRTp, emprespa,
   emprespp, Wa, Wp, old=FALSE) {
    dist_a <- WMdiffrespshiftweight(par = c(miat, par2,
       Tresn, Tresy), esterrorpar = esterrorpar, c = c, k
       = k, pr = FALSE, N = N, empRT = empRTa, empresp =
       emprespa, old = old)*Wa
    dist_p <- WMdiffrespshiftweight(par = c(miat, par2,
       Tresn, Tresy), esterrorpar = esterrorpar, c = c, k
       = k, pr = TRUE, N = N, empRT = empRTp, empresp =
       emprespp, old = old)*Wp
    return(dist_a + dist_p)
}

# For all data sets (with any number of set size levels).
   Function that fixes mst shared by data sets with the
   same set size level and estimates the common (miat,
   Tren, Tresy) for all data sets across set size level,
   whereby mst is a vector of the length length(setsize),
```

```
   k is a vector containing all set size levels and empRT
   and empresp the RTs and responses for all set size
   levels as list element
WMfixmst.sdmst <- function(par134, mst, esterrorpar, c, k,
    N, empRTa, empRTp, emprespa, emprespp, Wa, Wp, old=
   FALSE) {
    dist_a <- rep(NA, length(k))
    dist_p <- rep(NA, length(k))
    for (i in 1:length(k)) {
        dist_a[i] <- WMdiffrespshiftweight(par = c(par134
           [1], mst[i], par134[2], par134[3]), esterrorpar
            = esterrorpar, c = c, k = k[i], pr = FALSE, N
           = N, empRT = empRTa[[i]], empresp = emprespa[[i
           ]], old = old)*Wa[i]
        dist_p[i] <- WMdiffrespshiftweight(par = c(par134
           [1], mst[i], par134[2], par134[3]), esterrorpar
            = esterrorpar, c = c, k = k[i], pr = TRUE, N =
            N, empRT = empRTp[[i]], empresp = emprespp[[i
           ]], old = old)*Wp[i]
    }
    return(sum(dist_a + dist_p))
}

# For all data sets. Calculate the unweighted and the
   weighted distances and display the sum, whereby par is
   a 4 x length(setsize) matrix whose rows are indicated
   by the parameter and columns by setsize
ind.dist.sdmst <- function(par, esterrorpar, c, k, N,
   empRTa, empRTp, emprespa, emprespp, Wa, Wp, old=FALSE)
   {
    dist_a <- rep(NA, length(k))
    dist_p <- rep(NA, length(k))
    for (i in 1:length(k)) {
```

```
        dist_a[i] <- WMdiffrespshiftweight(par = par[,i],
           esterrorpar = esterrorpar, c = c, k = k[i], pr
           = FALSE, N = N, empRT = empRTa[[i]], empresp =
           emprespa[[i]], old = old)
        dist_p[i] <- WMdiffrespshiftweight(par = par[,i],
           esterrorpar = esterrorpar, c = c, k = k[i], pr
           = TRUE, N = N, empRT = empRTp[[i]], empresp =
           empresppp[[i]], old = old)
    }
    wdist_a <- dist_a*Wa
    wdist_p <- dist_p*Wa
    display <- rbind(dist_a, wdist_a, dist_p, wdist_p)
    display <- cbind(display, rowSums(display))
    rownames(display) <- c('absent', 'absent_relative', '
       present', 'present_relative')
    colnames(display) <- c(k, 'sum')
    return(display)
}

## source("unified_fitting_procedure_RTpar.R")
## estimating RT-related parameters using MDE

## Estimation procedure with set-size-invariant mu

## needs input: taskname, setsize, c, esterrorpar, miat_
   lower, miat_upper
## Make sure that the functions "WMonlycorrect", "
   extractpar", "extractvalue", "WMfixTresy", "WMfixTresn
   ", "WMfixqtimes.all", "WMfixTres.all", "ind.dist" and "
   sim.ny" are available

resultfile <- paste('Results_fitting_', taskname, '_c=', c
   )
backupfile <- paste('Backup_outputs_', taskname, '_c=', c)
```

```
cat('Results␣of␣fitting', taskname, 'with␣c␣=', c, '
    accuracy␣parameter␣=', esterrorpar, 'miat␣lower␣=',
    miat_lower, 'miat␣upper␣=', miat_upper, '\n', file =
    resultfile, append = TRUE)
cat('Backup␣outputs␣of␣fitting', taskname, 'with␣c␣=', c,
    '\n', file = backupfile, append = TRUE)

# create variables that contain the names of the data sets
    of different set size levels as character vector
name_estpar2a <- character(length(setsize))
name_estpar2p <- character(length(setsize))
name_value2a <- character(length(setsize))
name_value2p <- character(length(setsize))
name_rel_value2a <- character(length(setsize))
name_rel_value2p <- character(length(setsize))
for (i in 1:length(setsize)) {
    name_estpar2a[i] <- paste('estpar2a', setsize[i], sep
       = "_")
    name_estpar2p[i] <- paste('estpar2p', setsize[i], sep
       = "_")
    name_value2a[i] <- paste('value2a', setsize[i], sep =
       "_")
    name_value2p[i] <- paste('value2p', setsize[i], sep =
       "_")
    name_rel_value2a[i] <- paste('rel_value2a', setsize[i
       ], sep = "_")
    name_rel_value2p[i] <- paste('rel_value2p', setsize[i
       ], sep = "_")
}

# collect data across set size levels in a list of the
    length as set size levels, whose elements are the RT
    and response data for each set size level
```

```
RTa_all <- RTp_all <- respa_all <- respp_all <- list()
for (i in 1:length(setsize)) {
    RTa_all[[i]] <- get(name_data_a[i])[,1]
    RTp_all[[i]] <- get(name_data_p[i])[,1]
    respa_all[[i]] <- get(name_data_a[i])[,2]
    respp_all[[i]] <- get(name_data_p[i])[,2]
}

G1W_a <- rep(NA, length(setsize))
G1W_p <- rep(NA, length(setsize))

# divide the range of miat into 10 intervals
edge <- seq(miat_lower, miat_upper, length = 11)
ivmiat <- (edge[-1] + edge[1:10])/2

## Do the fitting procedure 0 to 1 on data sets sharing
   the same set size. In Generation 0, 10 optimazations
   will be run in parallel. The only difference is the
   initial value and allowed range of miat. 10 equi-
   distant points within the range of miat are chosen as
   initial values. The lower and upper bounds are adjusted
    accordingly. The estimates of the one that yields the
   smallest distance in Generation 1 will be selected as
   input for Generation 2
for (i in 1:length(setsize)) {
    cat('Fitting␣the␣model␣to␣data␣set', name_data_a[i], '
       and', name_data_p[i], 'with␣c␣=', c, 'and␣set␣size␣
       =', setsize[i], '\n', file = resultfile, append =
       TRUE)
    cat('Fitting␣the␣model␣to␣data␣set', name_data_a[i], '
       and', name_data_p[i], 'with␣c␣=', c, 'and␣set␣size␣
       =', setsize[i], '\n', file = backupfile, append =
       TRUE)
```

```
# Generation 0: estimate (miat, mst, Tresn) from only
   correct responses of target absent and (miat, mst,
   Tresy) from target present separately
G0a <- vector('list', 10)
G0p <- vector('list', 10)
for (j in 1:10) {
    G0a[[j]] <- optim(par = c(ivmiat[j], 500, 500), fn
        = WMonlycorrect, esterrorpar = esterrorpar, c
       = c, k = setsize[i], pr = FALSE, N = 100000,
       empRT = RTa_all[[i]], empresp = respa_all[[i]],
        old = FALSE, method = 'L-BFGS-B', lower = c(
       edge[j], 100, 50), upper = c(edge[j+1], 800,
       1000), control = list(fnscale = 100, parscale =
        c(50, 400, 150)))
    G0p[[j]] <- optim(par = c(ivmiat[j], 500, 500), fn
        = WMonlycorrect, esterrorpar = esterrorpar, c
       = c, k = setsize[i], pr = TRUE, N = 100000,
       empRT = RTp_all[[i]], empresp = respp_all[[i]],
        old = FALSE, method = 'L-BFGS-B', lower = c(
       edge[j], 100, 50), upper = c(edge[j+1], 800,
       1000), control = list(fnscale = 100, parscale =
        c(50, 400, 150)))
}
estpar1a <- sapply(G0a, extractpar)
value1a <- sapply(G0a, extractvalue)
estpar1p <- sapply(G0p, extractpar)
value1p <- sapply(G0p, extractvalue)
cat('Generation␣0␣target␣absent', name_data_a[i], '
   estimates:', capture.output(estpar1a), 'value:',
   capture.output(value1a), 'Generation␣0␣target␣
   present', name_data_p[i], 'estimates:', capture.
   output(estpar1p), 'value:', capture.output(value1p)
```

```
    , sep = '\n', file = resultfile, append = TRUE)
  cat('Generation␣0␣target␣absent', name_data_a[i],
    capture.output(G0a), 'Generation␣0␣target␣present',
     name_data_p[i], capture.output(G0p), sep = '\n',
    file = backupfile, append = TRUE)

  # Generation 1: estimate the parameters in generation
    0 again fixing Tresy (for target absent) and Tresn
    (for target present), using complete datasets
  G1a <- vector('list', 10)
  G1p <- vector('list', 10)
  for (j in 1:10) {
      G1a[[j]] <- optim(par = estpar1a[,j], fn =
         WMfixTresy, Tresy = estpar1p[3,j], esterrorpar
         = esterrorpar, c = c, k = setsize[i], pr =
         FALSE, N = 100000, empRT = RTa_all[[i]],
         empresp = respa_all[[i]], old = FALSE, method =
          'L-BFGS-B', lower = c(edge[j], 100, 50), upper
          = c(edge[j+1], 800, 1000), control = list(
         fnscale = 100, parscale = c(50, 400, 150)))
      G1p[[j]] <- optim(par = estpar1p[,j], fn =
         WMfixTresn, Tresn = estpar1a[3,j], esterrorpar
         = esterrorpar, c = c, k = setsize[i], pr = TRUE
         , N = 100000, empRT = RTp_all[[i]], empresp =
         respp_all[[i]], old = FALSE, method = 'L-BFGS-B
         ', lower = c(edge[j], 100, 50), upper = c(edge[
         j+1], 800, 1000), control = list(fnscale = 100,
          parscale = c(50, 400, 150)))
  }
  estpar2a <- sapply(G1a, extractpar)
  value2a <- sapply(G1a, extractvalue)
  G1W_a[i] <- 1/min(value2a)
  rel_value2a <- value2a/min(value2a)
```

```
    estpar2p <- sapply(G1p, extractpar)
    value2p <- sapply(G1p, extractvalue)
    G1W_p[i] <- 1/min(value2p)
    rel_value2p <- value2p/min(value2p)
    cat('Generation␣1␣target␣absent', name_data_a[i], '
       estimates:', capture.output(estpar2a), 'value:',
       capture.output(value2a), 'relative␣value:', rel_
       value2a, 'Generation␣1␣target␣present', name_data_p
       [i], 'estimates:', capture.output(estpar2p), 'value
       :', capture.output(value2p), 'relative␣value:', rel
       _value2p, sep = '\n', file=resultfile, append =
       TRUE)
    cat('Generation␣1␣target␣absent', name_data_a[i],
       capture.output(G1a), 'Generation␣1␣target␣present',
        name_data_p[i], capture.output(G1p), sep = '\n',
       file = backupfile, append = TRUE)
    assign(name_estpar2a[i], estpar2a)
    assign(name_value2a[i], value2a)
    assign(name_rel_value2a[i], rel_value2a)
    assign(name_estpar2p[i], estpar2p)
    assign(name_value2p[i], value2p)
    assign(name_rel_value2p[i], rel_value2p)
}

# calculate the weighted sum for each interval, select the
    least weighted sum and save the function values of the
    selected parameters
sumvalue <- rep(0, 10)
for (i in 1:length(setsize)) sumvalue <- sumvalue + get(
   name_rel_value2a[i]) + get(name_rel_value2p[i])
idx <- which(sumvalue == min(sumvalue))
selpar_a <- matrix(0, nrow=3, ncol=length(setsize))
selpar_p <- matrix(0, nrow=3, ncol=length(setsize))
```

```
for (i in 1:length(setsize)) {
    selpar_a[, i] <- get(name_estpar2a[i])[, idx]
    selpar_p[, i] <- get(name_estpar2p[i])[, idx]
}

# calculate the average (miat, mst) and (Tresn, Tresy)
estpar2 <- (rowSums(selpar_a) + rowSums(selpar_p))/(2 *
   length(setsize))
estTresn2 <- rowSums(selpar_a)[3]/length(setsize)
estTresy2 <- rowSums(selpar_p)[3]/length(setsize)
cat('average␣estimates␣of␣miat␣and␣mst␣from␣Generation␣1:'
   , capture.output(estpar2[1:2]), 'average␣estimates␣of␣
   Tresn␣from␣Generation␣1:', capture.output(estTresn2), '
   average␣estimates␣of␣Tresy␣from␣Generation␣1:', capture
   .output(estTresy2), 'weights␣of␣target␣absent␣data␣sets
   :', capture.output(G1W_a), 'weights␣of␣target␣present␣
   data␣sets:', capture.output(G1W_p), sep = '\n', file =
   resultfile, append = TRUE)
itstep <- 0
estqtimesold <- estpar2[1:2]
estTresnyold <- c(estTresn2, estTresy2)

while(TRUE) {
# Generation 2: fixing miat and mst, estimate common Tresn
    and Tresy from both datasets
    G2_all <- optim(par = estTresnyold, fn = WMfixqtimes.
       all, miat = estqtimesold[1], mst = estqtimesold[2],
        esterrorpar = esterrorpar, c = c, k = setsize, N =
        100000, empRTa = RTa_all, empRTp = RTp_all,
       emprespa = respa_all, emprespp = respp_all, Wa =
       G1W_a, Wp = G1W_p, old = FALSE, method = 'L-BFGS-B'
       , lower = c(50, 50), upper = c(1000, 1000), control
        = list(fnscale = 10, parscale = c(150, 150)))
```

```
    estTresnynew <- G2_all$par
    rel_value2 <- G2_all$value
    cat('Generation 2, iteration', itstep, ',', name_data_
       a, name_data_p, 'estimates of Tresn and Tresy:',
       capture.output(estTresnynew), 'relative value:',
       capture.output(rel_value2), sep = '\n', file =
       resultfile, append = TRUE)
    cat('Generation 2, iteration', itstep, ',', name_data_
       a, name_data_p, capture.output(G2_all), sep = '\n',
        file = backupfile, append = TRUE)

# Generation 3: fixing Tresn and Tresy, estimate miat and
   mst from both datasets
    G3_all <- optim(par = estqtimesold, fn = WMfixTres.all
       , Tresn = estTresnynew[1], Tresy = estTresnynew[2],
        esterrorpar = esterrorpar, c = c, k = setsize, N =
        100000, empRTa = RTa_all, empRTp = RTp_all,
       emprespa = respa_all, emprespp = respp_all, Wa =
       G1W_a, Wp = G1W_p, old = FALSE, method = 'L-BFGS-B'
       , lower = c(10, 100), upper = c(100, 800), control
       = list(fnscale = 10, parscale = c(50, 500)))
    estqtimesnew <- G3_all$par
    rel_value3 <- G3_all$value
    cat('Generation 3, iteration', itstep, ',', name_data_
       a, name_data_p, 'estimates of miat and mst:',
       capture.output(estqtimesnew), 'relative value:',
       capture.output(rel_value3), sep = '\n', file =
       resultfile, append = TRUE)
    cat('Generation 3, iteration', itstep, ',', name_data_
       a, name_data_p, capture.output(G3_all), sep = '\n',
        file = backupfile, append = TRUE)
    if (max(abs(estqtimesnew - estqtimesold)) < 0.5 && max
       (abs(estTresnynew - estTresnyold)) < 0.5 || itstep
```

```
        > 30) break
    itstep <- itstep + 1
    estqtimesold <- estqtimesnew
    estTresnyold <- estTresnynew
}

estpar3 <- c(estqtimesnew, estTresnynew)
cat('estimates from all data sets:', capture.output(
    estpar3), sep = '\n', file = resultfile, append = TRUE)
ind_dist <- ind.dist(par = estpar3, esterrorpar =
    esterrorpar, c = c, k = setsize, N = 100000, empRTa =
    RTa_all, empRTp = RTp_all, emprespa = respa_all,
    empRespp = respp_all, Wa = G1W_a, Wp = G1W_p, old =
    FALSE)
cat('Individual distances of data sets:', capture.output(
    ind_dist), sep = '\n', file = resultfile, append = TRUE
    )

# plot histograms comparing simulated and empirical data
png(filename = paste(taskname, '_c=', c, '_a.png'), width
    = 2880, height = 3840, res = 300)
par(mfrow = c(4,2))
for (i in 1:length(setsize)) {
    simdata <- sim.ny(par = estpar3, esterrorpar =
        esterrorpar, c = c, k = setsize[i], N = 100000, pr
        = FALSE, seed = 0)
    hist(RTa_all[[i]][respa_all[[i]] == FALSE], breaks =
        100, freq = FALSE, xlab = 'RT', ylab = 'Relative
        frequencies', main = paste('set size = ', setsize[i
        ], ', target absent, correct responses', sep = ""))
    hist(subset(simdata, simdata[,2] == FALSE, select = 1)
        , freq = FALSE, breaks = 50, border = 'red', add =
        TRUE)
```

```
    legend("topright", c("data", "model"), bty = 'n', lty
      = c(0, 0), pt.bg = c(1, 2), pt.cex = 2, pch = 22,
      col = c("black", "red"))
    hist(RTa_all[[i]][respa_all[[i]] == TRUE], breaks =
      100, freq = FALSE, xlab = 'RT', ylab = 'Relative␣
      frequencies', main = paste('set␣size␣=␣', setsize[i
      ], ',␣target␣absent,␣incorrect␣responses', sep = ""
      ))
    hist(subset(simdata, simdata[,2] == TRUE, select = 1),
       freq = FALSE, breaks = 50, border = 'red', add =
      TRUE)
    legend("topright", c("data", "model"), bty = 'n', lty
      = c(0, 0), pt.bg = c(1, 2), pt.cex = 2, pch = 22,
      col = c("black", "red"))
}
dev.off()

png(filename = paste(taskname, '_c=', c, '_p.png'), width
   = 2880, height = 3840, res = 300)
par(mfrow = c(4,2))
for (i in 1:length(setsize)) {
    simdata <- sim.ny(par = estpar3, esterrorpar =
      esterrorpar, c = c, k = setsize[i], N = 100000, pr
      = TRUE, seed = 0)
    hist(RTp_all[[i]][respp_all[[i]] == TRUE], breaks =
      100, freq = FALSE, xlab = 'RT', ylab = 'Relative␣
      frequencies', main = paste('set␣size␣=␣', setsize[i
      ], ',␣target␣present,␣correct␣responses', sep = "")
      )
    hist(subset(simdata, simdata[,2] == TRUE, select = 1),
       freq = FALSE, breaks = 50, border = 'red', add =
      TRUE)
    legend("topright", c("data", "model"), bty = 'n', lty
```

```
        = c(0, 0), pt.bg = c(1, 2), pt.cex = 2, pch = 22,
        col = c("black", "red"))
    hist(RTp_all[[i]][respp_all[[i]] == FALSE], breaks =
        100, freq = FALSE, xlab = 'RT', ylab = 'Relative␣
        frequencies', main = paste('set␣size␣=␣', setsize[i
        ], ',␣target␣present,␣incorrect␣responses', sep = "
        "))
    hist(subset(simdata, simdata[,2] == FALSE, select = 1)
        , freq = FALSE, breaks = 50, border = 'red', add =
        TRUE)
    legend("topright", c("data", "model"), bty = 'n', lty
        = c(0, 0), pt.bg = c(1, 2), pt.cex = 2, pch = 22,
        col = c("black", "red"))
}
dev.off()


## Estimation procedure with set-size-dependent mu

## source("unified_fitting_procedure_sdmst.R")

## needs input: taskname, setsize, c, esterrorpar, miat_
   lower, miat_upper
## Make sure that the functions "WMonlycorrect", "
   extractpar", "extractvalue", "WMfixTresy", "WMfixTresn
   ", "WMfixpar134.sdmst", "WMfixmst.sdmst", "ind.dist.
   sdmst" and "sim.ny" are available

resultfile <- paste('Results_fitting_', taskname, '_c=', c
   , '_sdmst')
backupfile <- paste('Backup_outputs_', taskname, '_c=', c,
    'sdmst')
cat('Results␣of␣fitting', taskname, 'with␣c␣=', c, '
```

```
    accuracy␣parameter␣=', esterrorpar, 'miat␣lower␣=',
    miat_lower, 'miat␣upper␣=', miat_upper, 'sdmst', '\n',
    file = resultfile, append = TRUE)
cat('Backup␣outputs␣of␣fitting', taskname, 'with␣c␣=', c,
    'sdmst', '\n', file = backupfile, append = TRUE)

# create variables that contain the names of the data sets
    of different set size levels as character vector
name_estpar2a <- character(length(setsize))
name_estpar2p <- character(length(setsize))
name_value2a <- character(length(setsize))
name_value2p <- character(length(setsize))
name_rel_value2a <- character(length(setsize))
name_rel_value2p <- character(length(setsize))
for (i in 1:length(setsize)) {
    name_estpar2a[i] <- paste('estpar2a', setsize[i], sep
       = "_")
    name_estpar2p[i] <- paste('estpar2p', setsize[i], sep
       = "_")
    name_value2a[i] <- paste('value2a', setsize[i], sep =
       "_")
    name_value2p[i] <- paste('value2p', setsize[i], sep =
       "_")
    name_rel_value2a[i] <- paste('rel_value2a', setsize[i
       ], sep = "_")
    name_rel_value2p[i] <- paste('rel_value2p', setsize[i
       ], sep = "_")
}

# collect data across set size levels in a list of the
    length as set size levels, whose elements are the RT
    and response data for each set size level
RTa_all <- RTp_all <- respa_all <- respp_all <- list()
```

```
for (i in 1:length(setsize)) {
    RTa_all[[i]] <- get(name_data_a[i])[,1]
    RTp_all[[i]] <- get(name_data_p[i])[,1]
    respa_all[[i]] <- get(name_data_a[i])[,2]
    respp_all[[i]] <- get(name_data_p[i])[,2]
}
G1W_a <- rep(NA, length(setsize))
G1W_p <- rep(NA, length(setsize))

# divide the range of miat into 10 intervals
edge <- seq(miat_lower, miat_upper, length = 11)
ivmiat <- (edge[-1] + edge[1:10])/2

## Do the fitting procedure 0 to 1 on data sets sharing
   the same set size. In Generation 0, 10 optimazations
   will be run in parallel. The only difference is the
   initial value and allowed range of miat. 10 equi-
   distant points within the range of miat are chosen as
   initial values. The lower and upper bounds are adjusted
    accordingly. The estimates of the one that yields the
   smallest distance in Generation 1 will be selected as
   input for Generation 2
for (i in 1:length(setsize)) {
    cat('Fitting␣the␣model␣to␣data␣set', name_data_a[i], '
       and', name_data_p[i], 'with␣c␣=', c, 'and␣set␣size␣
       =', setsize[i], '\n', file = resultfile, append =
       TRUE)
    cat('Fitting␣the␣model␣to␣data␣set', name_data_a[i], '
       and', name_data_p[i], 'with␣c␣=', c, 'and␣set␣size␣
       =', setsize[i], '\n', file = backupfile, append =
       TRUE)

    # Generation 0: estimate (miat, mst, Tresn) from only
```

```
    correct responses of target absent and (miat, mst,
    Tresy) from target present separately
G0a <- vector('list', 10)
G0p <- vector('list', 10)
for (j in 1:10) {
    G0a[[j]] <- optim(par = c(ivmiat[j], 500, 500), fn
        = WMonlycorrect, esterrorpar = esterrorpar, c
       = c, k = setsize[i], pr = FALSE, N = 100000,
       empRT = RTa_all[[i]], empresp = respa_all[[i]],
        old = FALSE, method = 'L-BFGS-B', lower = c(
       edge[j], 100, 50), upper = c(edge[j+1], 800,
       1000), control = list(fnscale = 100, parscale =
        c(50, 400, 150)))
    G0p[[j]] <- optim(par = c(ivmiat[j], 500, 500), fn
        = WMonlycorrect, esterrorpar = esterrorpar, c
       = c, k = setsize[i], pr = TRUE, N = 100000,
       empRT = RTp_all[[i]], empresp = respp_all[[i]],
        old = FALSE, method = 'L-BFGS-B', lower = c(
       edge[j], 100, 50), upper = c(edge[j+1], 800,
       1000), control = list(fnscale = 100, parscale =
        c(50, 400, 150)))
}
estpar1a <- sapply(G0a, extractpar)
value1a <- sapply(G0a, extractvalue)
estpar1p <- sapply(G0p, extractpar)
value1p <- sapply(G0p, extractvalue)
cat('Generation␣0␣target␣absent', name_data_a[i], '
    estimates:', capture.output(estpar1a), 'value:',
    capture.output(value1a), 'Generation␣0␣target␣
    present', name_data_p[i], 'estimates:', capture.
    output(estpar1p), 'value:', capture.output(value1p)
    , sep = '\n', file = resultfile, append = TRUE)
cat('Generation␣0␣target␣absent', name_data_a[i],
```

```
    capture.output(G0a), 'Generation␣0␣target␣present',
     name_data_p[i], capture.output(G0p), sep = '\n',
    file = backupfile, append = TRUE)

# Generation 1: estimate the parameters in generation
   0 again fixing Tresy (for target absent) and Tresn
   (for target present), using complete datasets
G1a <- vector('list', 10)
G1p <- vector('list', 10)
for (j in 1:10) {
    G1a[[j]] <- optim(par = estpar1a[,j], fn =
       WMfixTresy, Tresy = estpar1p[3,j], esterrorpar
       = esterrorpar, c = c, k = setsize[i], pr =
       FALSE, N = 100000, empRT = RTa_all[[i]],
       empresp = respa_all[[i]], old = FALSE, method =
        'L-BFGS-B', lower = c(edge[j], 100, 50), upper
        = c(edge[j+1], 800, 1000), control = list(
       fnscale = 100, parscale = c(50, 400, 150)))
    G1p[[j]] <- optim(par = estpar1p[,j], fn =
       WMfixTresn, Tresn = estpar1a[3,j], esterrorpar
       = esterrorpar, c = c, k = setsize[i], pr = TRUE
       , N = 100000, empRT = RTp_all[[i]], empresp =
       respp_all[[i]], old = FALSE, method = 'L-BFGS-B
       ', lower = c(edge[j], 100, 50), upper = c(edge[
       j+1], 800, 1000), control = list(fnscale = 100,
        parscale = c(50, 400, 150)))
}
estpar2a <- sapply(G1a, extractpar)
value2a <- sapply(G1a, extractvalue)
G1W_a[i] <- 1/min(value2a)
rel_value2a <- value2a/min(value2a)
estpar2p <- sapply(G1p, extractpar)
value2p <- sapply(G1p, extractvalue)
```

```
    G1W_p[i] <- 1/min(value2p)
    rel_value2p <- value2p/min(value2p)
    cat('Generation␣1␣target␣absent', name_data_a[i], '
       estimates:', capture.output(estpar2a), 'value:',
       capture.output(value2a), 'relative␣value:', rel_
       value2a, 'Generation␣1␣target␣present', name_data_p
       [i], 'estimates:', capture.output(estpar2p), 'value
       :', capture.output(value2p), 'relative␣value:', rel
       _value2p, sep = '\n', file=resultfile, append =
       TRUE)
    cat('Generation␣1␣target␣absent', name_data_a[i],
       capture.output(G1a), 'Generation␣1␣target␣present',
        name_data_p[i], capture.output(G1p), sep = '\n',
       file = backupfile, append = TRUE)
    assign(name_estpar2a[i], estpar2a)
    assign(name_value2a[i], value2a)
    assign(name_rel_value2a[i], rel_value2a)
    assign(name_estpar2p[i], estpar2p)
    assign(name_value2p[i], value2p)
    assign(name_rel_value2p[i], rel_value2p)
}

# calculate the weighted sum for each interval, select the
    least weighted sum and save the function values of the
    selected parameters
sumvalue <- rep(0, 10)
for (i in 1:length(setsize)) sumvalue <- sumvalue + get(
   name_rel_value2a[i]) + get(name_rel_value2p[i])
idx <- which(sumvalue == min(sumvalue))

selpar_a <- matrix(0, nrow=3, ncol=length(setsize))
selpar_p <- matrix(0, nrow=3, ncol=length(setsize))
```

```
for (i in 1:length(setsize)) {
    selpar_a[, i] <- get(name_estpar2a[i])[, idx]
    selpar_p[, i] <- get(name_estpar2p[i])[, idx]
}

# calculate the average (miat, mst) and (Tresn, Tresy)
estmiat2 <- (rowSums(selpar_a)[1] + rowSums(selpar_p)[1])/
   (2*length(setsize))
estmst2 <- (selpar_a[2, ] + selpar_p[2, ])/2
estTresn2 <- rowSums(selpar_a)[3]/length(setsize)
estTresy2 <- rowSums(selpar_p)[3]/length(setsize)

cat('average estimate of miat from Generation 1:', capture
   .output(estmiat2), 'average estimate of mst from
   Generation 1:', capture.output(estmst2), 'average
   estimates of Tresn from Generation 1:', capture.output(
   estTresn2), 'average estimates of Tresy from Generation
   1:', capture.output(estTresy2), 'weights of target
   absent data sets:', capture.output(G1W_a), 'weights of
   target present data sets:', capture.output(G1W_p), sep
   = '\n', file = resultfile, append = TRUE)

itstep <- 0
estpar134old <- c(estmiat2, estTresn2, estTresy2)
estmstold <- estmst2

while(TRUE) {
# Generation 2: fixing miat, Tresn and Tresy for all data
   sets, estimate common mst from both datasets with the
   same set size
    G2 <- vector('list', length(setsize))
    for (i in 1:length(setsize)) {
        G2[[i]] <- optim(par = estmstold[i], fn =
```

```
            WMfixpar134.sdmst, miat = estpar134old[1],
            Tresn = estpar134old[2], Tresy = estpar134old
            [3], esterrorpar = esterrorpar, c = c, k =
            setsize[i], N = 100000, empRTa = RTa_all[[i]],
            empRTp = RTp_all[[i]], emprespa = respa_all[[i
            ]], emprespp = respp_all[[i]], Wa = G1W_a[i],
            Wp = G1W_p[i], old = FALSE, method = 'L-BFGS-B'
            , lower = 100, upper = 800, control = list(
            fnscale = 10, parscale = 400))
    }
    estmstnew <- sapply(G2, extractpar)
    rel_value2 <- sapply(G2, extractvalue)
    cat('Generation␣2,␣iteration', itstep, ',', name_data_
       a, name_data_p, 'estimates␣of␣mst:', capture.output
       (estmstnew), 'relative␣value:', capture.output(rel_
       value2), sep = '\n', file = resultfile, append =
       TRUE)
    cat('Generation␣2,␣iteration', itstep, ',', name_data_
       a, name_data_p, capture.output(G2), sep = '\n',
       file = backupfile, append = TRUE)

# Generation 3: fixing mst, estimate miat, Tresn and Tresy
    from all datasets
    G3_all <- optim(par = estpar134old, fn = WMfixmst.
       sdmst, mst = estmstnew, esterrorpar = esterrorpar,
       c = c, k = setsize, N = 100000, empRTa = RTa_all,
       empRTp = RTp_all, emprespa = respa_all, emprespp =
       respp_all, Wa = G1W_a, Wp = G1W_p, old = FALSE,
       method = 'L-BFGS-B', lower = c(10, 50, 50), upper =
        c(100, 1000, 1000), control = list(fnscale = 10,
       parscale = c(50, 150, 150)))
    estpar134new <- G3_all$par
    rel_value3_all <- G3_all$value
```

```
    cat('Generation␣3,␣iteration', itstep, ',', name_data_
      a, name_data_p, 'estimates␣of␣miat,␣Tresn␣and␣Tresy
      :', capture.output(estpar134new), 'relative␣value:'
      , capture.output(rel_value3_all), sep = '\n', file
      = resultfile, append = TRUE)
    cat('Generation␣3,␣iteration', itstep, ',', name_data_
      a, name_data_p, capture.output(G3_all), sep = '\n',
       file = backupfile, append = TRUE)
    if (max(abs(estpar134new - estpar134old)) < 0.5 && max
      (abs(estmstnew - estmstold)) < 0.5 || itstep > 30)
      break
    itstep <- itstep + 1
    estpar134old <- estpar134new
    estmstold <- estmstnew
}

estpar3 <- rbind(rep(estpar134new[1], length(setsize)),
   estmstnew, matrix(estpar134new[2:3], nrow = 2, ncol =
   length(setsize)))
cat('estimates␣from␣all␣data␣sets:', capture.output(
   estpar3), sep = '\n', file = resultfile, append = TRUE)

ind_dist <- ind.dist.sdmst(par = estpar3, esterrorpar =
   esterrorpar, c = c, k = setsize, N = 100000, empRTa =
   RTa_all, empRTp = RTp_all, emprespa = respa_all,
   emprespp = respp_all, Wa = G1W_a, Wp = G1W_p, old =
   FALSE)
cat('Individual␣distances␣of␣data␣sets:', capture.output(
   ind_dist), sep = '\n', file = resultfile, append = TRUE
   )

# plot histograms comparing simulated and empirical data
png(filename = paste(taskname, '_c=', c, '_a.png'), width
```

```
   = 2880, height = 3840, res = 300)
par(mfrow = c(4,2))
for (i in 1:length(setsize)) {
    simdata <- sim.ny(par = estpar3[,i], esterrorpar =
       esterrorpar, c = c, k = setsize[i], N = 100000, pr
       = FALSE, seed = 0)
    hist(RTa_all[[i]][respa_all[[i]] == FALSE], breaks =
       100, freq = FALSE, xlab = 'RT', ylab = 'Relative␣
       frequencies', main = paste('set␣size␣=␣', setsize[i
       ], ',␣target␣absent,␣correct␣responses', sep = ""))
    hist(subset(simdata, simdata[,2] == FALSE, select = 1)
       , freq = FALSE, breaks = 50, border = 'red', add =
       TRUE)
    legend("topright", c("data", "model"), bty = 'n', lty
       = c(0, 0), pt.bg = c(1, 2), pt.cex = 2, pch = 22,
       col = c("black", "red"))
    hist(RTa_all[[i]][respa_all[[i]] == TRUE], breaks =
       100, freq = FALSE, xlab = 'RT', ylab = 'Relative␣
       frequencies', main = paste('set␣size␣=␣', setsize[i
       ], ',␣target␣absent,␣incorrect␣responses', sep = ""
       ))
    hist(subset(simdata, simdata[,2] == TRUE, select = 1),
        freq = FALSE, breaks = 50, border = 'red', add =
       TRUE)
    legend("topright", c("data", "model"), bty = 'n', lty
       = c(0, 0), pt.bg = c(1, 2), pt.cex = 2, pch = 22,
       col = c("black", "red"))
}
dev.off()

png(filename = paste(taskname, '_c=', c, '_p.png'), width
   = 2880, height = 3840, res = 300)
par(mfrow = c(4,2))
```

```
for (i in 1:length(setsize)) {
    simdata <- sim.ny(par = estpar3[,i], esterrorpar =
       esterrorpar, c = c, k = setsize[i], N = 100000, pr
       = TRUE, seed = 0)
    hist(RTp_all[[i]][respp_all[[i]] == TRUE], breaks =
       100, freq = FALSE, xlab = 'RT', ylab = 'Relative␣
       frequencies', main = paste('set␣size␣=␣', setsize[i
       ], ',␣target␣present,␣correct␣responses', sep = "")
       )
    hist(subset(simdata, simdata[,2] == TRUE, select = 1),
        freq = FALSE, breaks = 50, border = 'red', add =
       TRUE)
    legend("topright", c("data", "model"), bty = 'n', lty
       = c(0, 0), pt.bg = c(1, 2), pt.cex = 2, pch = 22,
       col = c("black", "red"))
    hist(RTp_all[[i]][respp_all[[i]] == FALSE], breaks =
       100, freq = FALSE, xlab = 'RT', ylab = 'Relative␣
       frequencies', main = paste('set␣size␣=␣', setsize[i
       ], ',␣target␣present,␣incorrect␣responses', sep = "
       "))
    hist(subset(simdata, simdata[,2] == FALSE, select = 1)
       , freq = FALSE, breaks = 50, border = 'red', add =
       TRUE)
    legend("topright", c("data", "model"), bty = 'n', lty
       = c(0, 0), pt.bg = c(1, 2), pt.cex = 2, pch = 22,
       col = c("black", "red"))
}
dev.off()
```