PARAMETER IDENTIFICATION IN NONLINEAR DYNAMIC SYSTEMS WITH META-HEURISTIC APPROACHES

Katerina Tashkova

Doctoral Dissertation
Jožef Stefan International Postgraduate School
Ljubljana, Slovenia, May 2012

Supervisor: Prof. Dr. Sašo Džeroski, Jožef Stefan Institute, Ljubljana, Slovenia
Co-supervisor: Assist. Prof. Dr. Jurij Šilc, Jožef Stefan Institute, Ljubljana, Slovenia

Evaluation Board:
Assist. Prof. Dr. Peter Korošec, Chairman, Faculty of Mathematics, Natural Sciences and Information Technologies, University of Primorska, Koper, Slovenia
Assist. Prof. Dr. Ljupčo Todorovski, Member, Faculty of Administration, University of Ljubljana, Ljubljana, Slovenia
Dr. Eva Balsa-Canto, Member, Bioprocess Engineering Group, National Spanish Research Council, Vigo, Spain

MEDNARODNA PODIPLOMSKA ŠOLA JOŽEFA STEFANA
JOŽEF STEFAN INTERNATIONAL POSTGRADUATE SCHOOL
Ljubljana, Slovenia

Katerina Tashkova

PARAMETER IDENTIFICATION IN NONLINEAR DYNAMIC SYSTEMS WITH META-HEURISTIC APPROACHES

Doctoral Dissertation

IDENTIFIKACIJA PARAMETROV V NELINEARNIH DINAMIČNIH SISTEMIH Z METAHEVRISTIČNIMI PRISTOPI

Doktorska disertacija

Supervisor: Prof. Dr. Sašo Džeroski

Co-supervisor: Assist. Prof. Dr. Jurij Šilc

Ljubljana, Slovenia, May 2012

To my mother

На мојата мајка

Contents

Abstract

The task of mathematical modeling of dynamic systems from observed system behavior, widely known under the name of system identification, breaks down into two subtasks. The first task, referred to as structure identification, is to specify the model structure, *i.e.*, the functional form of the model. In practice, the model structure is usually given by a human domain expert and reflects prior domain knowledge: this is called knowledge-driven identification (as opposed to data-driven identification, which is based only on data). Structure identification plays an important role in modeling as it defines the choice available for the selection of the "best model".

The second task, referred to as parameter identification, aims to estimate the values of the model parameters that define a best possible fit of the model to the measured data. It assumes that the model structure is known and the observed system behavior is given in the form of measured data. Accurate estimation of the model parameters is important for describing and analyzing the behavior of the modeled system. Parameter identification is therefore a crucial step in almost all approaches for reconstructing system dynamics from measured data, including knowledge-driven and data-driven system identification as well as traditional (human) and automated modeling, *i.e.*, the automated discovery of appropriate model structures and model parameter values by equation discovery tools.

In this dissertation, we address the task of parameter identification in dynamic models of real-life systems. The models are represented by ordinary differential equations (ODEs), as considered in the fields of systems biology and ecological modeling. The task is approached as a least-squares estimation problem within the frequentist framework. The latter means that the model parameters have fixed unique values and their optimal values are the ones that minimize a quadratic cost function, *i.e.*, the sum of squared errors between the model prediction and the experimentally measured data. Least-squares estimation is essentially an optimization task. However, it can turn into a difficult problem for traditional (gradient-based) optimization methods when modeling complex system dynamics. Therefore, it should be addressed by advanced meta-heuristic approaches, such as evolutionary or swarm intelligence methods.

Typically, biological and ecosystem models are nonlinear and have many parameters, the studied systems can often be only partially observed, and their measurements are sparse and imperfect due to noise. All of these constraints can lead to identifiability problems, *i.e.*, the inability to uniquely identify the unknown model parameters, making parameter estimation an even harder optimization task. Furthermore, the implicit definition of the cost function requires expensive numerical ODE simulations that have to be performed for every parameter solution investigated during the optimization process. As a result, parameter identification is a challenging and computationally expensive step in the process of reconstructing the structure and behavior of biological and ecological systems.

This dissertation attempts to improve the quality of reconstructed system dynamics by improving parameter identification. In this context, we perform a thorough empirical evaluation of representative meta-heuristic methods on the task of estimating parameters in two nonlinear ODE models. The considered models describe two practically rele-

vant and representative real-life systems, *i.e.*, endosome maturation in endocytosis and a food web of Lake Bled. The compared meta-heuristic methods are the differential ant-stigmergy algorithm, the continuous differential ant-stigmergy algorithm, particle swarm optimization, and differential evolution. As a baseline method for the experimental comparison, we use Algorithm 717, a gradient-based local search method essentially designed for nonlinear least-squares estimation. Different experimental scenarios are considered to investigate the effect of limited observability of the system dynamics, the influence of the ODE simulation method, and the impact of the noise in the data, on the complexity of the parameter identification task, as well as the applicability and performance of different optimization methods in this context.

The empirical evaluation shows that the meta-heuristic global optimization methods for parameter identification are clearly superior and should be preferred over local optimization methods. While the differences in performance between the different methods within the class of meta-heuristics are not significant across all conditions, differential evolution yields the best results in terms of the quality of the reconstructed system dynamics as well as the speed of convergence. The observability of the system shows a strong influence, where less complete observations make the optimization task much more difficult. The results clearly indicate the importance of choosing a relevant cost function when the modeled systems dynamics is only partially observed. While the use of a simple one-step trapezoidal-based integrator for supervised prediction makes parameter identification much faster, the use of a multistep variable-coefficient integrator for unsupervised prediction produces much better parameter estimates from real-measured data.

Furthermore, we consider the problem of parameter identification within the process of automated modeling of dynamic systems, where a large number of model structures is considered. One major drawback of existing automated modeling approaches is the use of local search methods for parameter identification. In this context, we investigate the influence of parameter identification (in terms of a global and a local optimization method) on the outcome of the automated modeling process, *i.e.*, on what models are selected. We consider eight tasks of automated modeling of phytoplankton dynamics in Lake Bled from single-year data measured in eight different years. The outcome of the experiments empirically demonstrate the benefit of estimating model parameters by global optimization methods for the model (structure) selection process, opening the opportunity to model long term system dynamics.

Many challenges still remain concerning the use of optimization methods for parameter identification in dynamic systems, especially in the context of automated modeling by equation discovery methods. Besides the need to extend our study by including additional dynamic systems from different domains, several lines for further improvement of existing automated modeling methods can be followed. These include the use of more appropriate and informative cost functions, as well as more robust and faster methods for parameter identification. Finally, explicit integration of the feedback from identifiability analysis within the process of model selection is highly desirable.

Povzetek

Matematično modeliranje dinamičnih sistemov na osnovi opazovanja obnašanja sistema, pogosto imenovano identifikacija sistema, temelji v bistvu na dveh nalogah: identifikaciji strukture in identifikaciji parametrov modela.

Cilj prve naloge, imenovane identifikacija strukture, je določiti ustrezno strukturo modela, tj. funkcionalno obliko modela. V praksi navadno strokovnjak poda strukturo modela v problemski domeni, ki odraža predhodno znanje o domeni: govorimo o teoretični identifikaciji v nasprotju z empirično identifikacijo, ki temelji le na podatkih. Identifikacija strukture igra pomembno vlogo pri modeliranju, saj določa možnost izbire najboljšega modela.

Cilj druge naloge, imenovane identifikacija parametrov modela, je pri dani strukturi modela opazovanega sistema in izmerjenih podatkih tega sistema oceniti vrednosti parametrov modela tako, da se bo ta kar najbolj ujemal z izmerjenimi podatki. Natančna ocena vrednosti parametrov modela je pomembna pri opisu in analizi obnašanja modeliranega sistema ter je torej ključni korak skoraj vseh pristopov za rekonstrukcijo dinamike sistemov iz izmerjenih podatkov. To velja tako pri teoretični in empirični identifikaciji sistema, kakor tudi pri tradicionalnem (človeškem) in avtomatskem modeliranju, tj. avtomatsko odkrivanje ustrezne strukture modela in vrednosti parametrov modela z uporabo orodij za odkrivanje enačb.

V disertaciji obravnavamo nalogo identifikacije parametrov v modelih dinamičnih realnih sistemov, ki so podani s sistemov diferencialnih enačb. Obravnavamo dve področji, in sicer: sistemsko biologijo in ekološko modeliranje. Nalogo rešujemo s tradicionalnim, frekventističnim pristopom, ki temelji na oceni najmanjših kvadratov. Kar pomeni, da obravnavamo parametre kot konstante katerih optimalne vrednosti so tiste, ki minimizirajo namensko funkcijo, tj. vsoto kvadratov napak med napovedjo modela in izmerjenimi podatki. Ocena najmanjših kvadratov je v osnovi optimizacijski problem, ki lahko pri kompleksni dinamiki sistema postane pretežak za tradicionalne, gradientne optimizacijske metode. Zato se tega problema lotevamo z uporabo naprednih, metahevrističnih pristopov, kakršni so evolucijsko računanje in inteligentni roji.

Za modele bioloških in ekoloških sistemov v splošnem velja, da so nelinearni in imajo veliko parametrov. Večina parametrov je neznanih in se morajo identificirati na osnovi redkih, običajno nenatančnih meritev, ki le delno odražajo obnašanje modeliranega sistema (t. i. omejena observabilnost). Vse te omejitve lahko vodijo k problemu, ko ni mogoče najti prave vrednosti parametrov (t. i. problem identifikabilnosti parametrov), kar še otežuje nalogo identifikacije parametrov. Po drugi strani je namenska funkcija definirana posredno, kar zahteva uporabo dragih integratorjev za numerično reševanje sistema diferencialnih enačb, ki se mora izvajati med optimizacijo za vsako nastavitev parametrov posebej. Oboje se odraža v tem, da je identifikacija parametrov izjemno težek in računsko zahteven korak v procesu rekonstrukcije strukture in obnašanja bioloških in ekoloških sistemov.

Z disertacijo skušamo izboljšati kakovost rekonstruirane dinamike sistema tako, da izboljšamo identifikacijo parametrov. V povezavi s tem smo opravili obsežno empirično vrednotenje gradientne optimizacijske metode in več izbranih metahevrističnih metod

za reševanje problema identifikacije parametrov v nelinearnih dinamičnih modelih. Kot primer smo vzeli dva realna sistema: življenjski cikel endosoma v endocitozi in prehranjevalno mrežo v Blejskem jezeru. Primerjali smo gradientno optimizacijsko metodo, imenovano Algoritem 717, in naslednje metahevristične metode: diskretni in zvezni diferencialni algoritem s stigmergijo mravelj, optimizacijo z roji delcev in difrencialno evolucijo. Upoštevali smo različne eksperimentalne scenarije z namenom da bi raziskovali: učinek omejene observabilnosti dinamike sistema, vpliv metodo za numerično reševanje sistema diferencialnih enačb in vpliv šuma v podatkih na kompleksnosti problema identifikacije parametrov; v tej zvezi smo tudi ocenili uporabnost in učinkovitost različnih optimizacijskih metod.

Empirična analiza je pokazala, da so globalne, metahevristične optimizacijske metode za identifikacijo parametrov boljše in primernejše od lokalnih, gradientnih optimizacijskih metod. Med metahevrističnimi optimizacijskimi metodami ni pomembnejših razlik v učinkovitosti, a vseeno dosega diferencialna evolucija najboljše rezultate, tako glede kakovosti rekonstruirane dinamike sistema, kakor tudi glede hitrejše konvergence k rešitvi. Ugotovili smo tudi, da se z omejevanjem observabilnosti sistema povečuje težavnost optimizacijskega problema. Rezultati jasno kažejo kako pomembna je izbira prave namenske funkcije kadar je dinamika sistema le delno observabilna. Uporaba preprostega, enokoračnega trapezoidnega integratorja z nadzorovano predikcijo vodi k hitrejši identifikaciji parametrov, uporaba večkoračnega integratorja s spremenljivimi koeficienti in nenadzorovano predikcijo pa se odraža v boljši identifikaciji parametrov v primeru realnih, merjenih podatkov.

Ukvarjali smo se tudi s problemom identifikacije parametrov v procesu avtomatskega modeliranja dinamičnih sistemov, kjer imamo na voljo številne strukture modela. Ena od slabosti obstoječih pristopov avtomatskega modeliranja je uporaba lokalnih iskalnih metod za identifikacijo parametrov. Zato smo raziskovali vpliv uporabe globalne in lokalne optimizacijske metode pri identifikaciji parametrov na rezultat avtomatskega modeliranja, tj. izbiro modelov. Obravnavali smo avtomatsko modeliranje letne dinamike fitoplanktona v Blejskem jezeru na osnovi meritev opravljenih v osmih zaporednih letih, kar pomeni osem različnih nalog modeliranja. Rezultati empirične analize kažejo, da uporaba globalne optimizacijske metode prinaša prednosti v procesu izbire modelov in odpira možnost zanesljivejšega modeliranja dinamike v daljšem časovnem obdobju.

Glede uporabe optimizacijskih metod za identifikacijo parametrov v dinamičnih sistemih ostajajo številni izzivi, zlasti v okviru avtomatskega modeliranja. Po eni strani bi bilo smiselno v analizo, ki je bila opravljena v disertaciji, vključiti še dodatne dinamične sisteme (po možnosti z drugih področij), po drugi strani pa se odpira še veliko možnosti za izboljšanje obstoječih metod za avtomatsko modeliranje. Zadnje vključuje: uporabo primernejše in bolj informativne namenske funkcije, kakor tudi bolj robustne in hitrejše metode za identifikacijo parametrov. Ne nazadnje si lahko veliko obetamo tudi od primernega upoštevanja rezultata analize identifiabilnosti v procesu izbire modela.

Abbreviations

ACO	=	ant colony optimization
AIC	=	Akaike information criterion
AO	=	active-state protein concentration observation
A717	=	Algorithm 717
BIC	=	Bayesian information criterion
CDASA	=	continuous differential ant-stigmergy algorithm
CI	=	confidence interval
CO	=	complete observation
DAE	=	differential-algebraic equation
DASA	=	differential ant-stigmergy algorithm
DE	=	differential evolution
EA	=	evolutionary algorithm
ES	=	evolutionary strategy
FIM	=	Fisher information matrix
FS	=	full simulation
GA	=	genetic algorithm
IQR	=	interquartile range
LS	=	least-squares
MAP	=	maximum a posteriori
ML	=	maximum-likelihood
NPO	=	neglecting passive-state protein concentration observation
ODE	=	ordinary differential equation
PDE	=	partial differential equation
PDF	=	probability density function
ProBMoT	=	process-based modeling tool
PSO	=	particle swarm optimization
RMSE	=	root mean squared error
RMSEm	=	root mean squared error of the completely simulated model
SSE	=	sum of weighted squared errors
SSEo	=	sum of squared errors
SSEr	=	sum of squared relative errors
TF	=	teacher forcing simulation
TO	=	total protein concentration observation

1 Introduction

Modeling is a conceptual tool widely used in science and engineering to formalize the abstract notions of reality. It provides a framework (*e.g.*, for applying mathematics and logics) that can be evaluated and reused for reasoning in a wide range of situations and across different domains. Modeling by mathematical models provides a powerful framework for scientists and engineers to analyze the behavior of real-world systems in quantitative terms. Theoretical and numerical analysis of the mathematically formulated problem can reveal important insights, answers and guidance useful for the specific system being observed as well as for the relevant domain of interest in general. Mathematical models are therefore of fundamental importance for scientists and engineers.

Building mathematical models, however, is a difficult decision-making process that needs more than just the application of scientific knowledge to result in a useful model; it relies on intuition, experience and empiricism, in addition to scientific knowledge (Gershenfeld, 1999; Klee and Allen, 2011). The typical process involves many steps from proper problem formulation, through acquisition of domain knowledge and/or experimentally measured data, to decision about the specific modeling formalism, followed by decision on the model structure and its parameters as well as their validation, usually on new experimental data, as a final test of model usefulness. The steps are carried in an iterative fashion until a specific model meets the requirements imposed by the purpose of its use. In parallel, experiments are designed to deliver data of better quality needed for accurate model identification and reliable model validation. Hence, establishing an acceptable mathematical model of an observed system is an extremely demanding and time-consuming task.

While all steps of the modeling procedure are important, it is the task of model identification that is at the heart of mathematical modeling. It is closely related to the task of modeling dynamic systems from measured (input/output) data, referred to as *system identification* in engineering (Ljung, 1987). This roughly involves definition of the modeling framework, decision on the model structure, and determination of the adjustable model parameters from experimental data on the observed system behavior.

The modeling framework includes specification of the type of the model, the relevant variables for the modeling task, and the specific mathematical formalism for model representation. In general, real-world systems exhibit complex nonlinear dynamic behavior, which is often modeled using differential equations. In particular, the formalism of ordinary differential equations (ODEs) is the most widely accepted mathematical formalism for modeling deterministic dynamic systems, *i.e.*, systems that evolve continuously over time, whose time evolution is uniquely determined by their initial state and the time evolution of the inputs in the system. ODE models arise naturally as a result of first-principle mechanistic modeling of dynamic systems, and are therefore especially convenient for scientists and engineers.

Assuming the ODE mathematical framework, inferring ODE models of complex dynamic systems from measured data essentially amounts to two tasks. The first task, referred to as *structure identification* and often solved by a modeling expert, is to spec-

ify the model structure, *i.e.*, the functional form of the model ODEs. The second task of determining appropriate values for the model parameters, based on observations and measurements, is referred to as *parameter identification*.

Structure identification plays a critical role in modeling, since it defines the choice available for the selection of the "best model". In practice, the model structure is usually given by a domain expert and reflects their prior knowledge: this means knowledge-driven structure identification. Unlike structure identification, parameter identification is almost exclusively data-driven, *i.e.*, given the structure of the model, the values of the model parameters are determined by fitting the model predictions to the experimental data.

In complex nonlinear dynamic models, as considered in this dissertation, the estimation of the model parameters is an iterative adjustment procedure, that can be formulated as a problem of optimization. In particular, we have to minimize a certain criterion, called cost function, that quantifies the goodness of the model with respect to the adjusted model parameters. The choice of the cost function is the first important problem associated with the parameter identification task: the cost function should reflect the aim of the model.

Different cost functions can be found in the literature. However, their formulation is conceptually driven by two main parameter identification paradigms: the *frequentist* (referred to as the "classical") approach and the *Bayesian* (probabilistic) approach (Rice, 2007; Samaniego, 2010). The frequentist approach assumes that the given data are random samples from the universe of data, while the model parameters are constants, *i.e.*, have unique fixed values, for which no additional information is needed in advance. The most representative approach of this class is *maximum-likelihood* (ML) estimation, according to which the most likely parameter values are the ones that maximize the likelihood, *i.e.*, probability of observing the given data. A special case of maximum-likelihood estimation, based on the assumption of independent and normally distributed errors in the experimental data, leads to the well-known approach of *least-squares* (LS) estimation.

Unlike the frequentist approach, which does not need any external information about the parameters, the Bayesian approach treats the parameters to be estimated as random variables. The prior distribution of these variables represents the knowledge about the parameter values before taking the data into account and is given as input to the approach in addition to the data. The approach then modifies the given prior distribution of the parameters by the likelihood information in order to obtain the posterior distribution of the parameters after observing the data.

It is difficult to argue in favor of one class of approaches against the other in a general manner (Samaniego, 2010), since both have shown advantages in specific situations. On one hand, Bayesian approaches can elegantly treat the uncertainty in parameter values and model structure in a uniform manner. On the other hand, frequentist approaches can be effectively used for high-dimensional models with large numbers of parameters. Nevertheless, according to the information that the end-user has to provide, LS estimation is the simplest approach, while Bayesian approaches are the most complex ones (Jaqaman and Danuse, 2006; Rice, 2007). As a result, the intuitive LS estimator, formulated as a simple sum of squared (weighted) errors between the model predictions and the measured data, is the most widely used cost function in practice (Ljung, 1987).

In this dissertation, the focus will be on the parameter identification task, approached from the frequentist point of view using least-squares estimation. The task will be addressed in the context of modeling deterministic systems represented by ODE models in two real-life domains, *i.e.*, systems biology and ecological modeling.

1.1 Motivation

The research presented in this dissertation is essentially motivated by the need to improve the quality of reconstructed system dynamics. It attempts to meet this need by improving parameter identification, which plays a central role in almost all approaches for reconstructing system dynamics. This holds for data-driven and knowledge-driven system identification as well as traditional (human) and automated modeling, *i.e.*, the automated discovery of appropriate model structures, including the values of the model parameters, by equation discovery tools.

In this context, it is important to be aware of the parameter identification related problems faced by current modeling approaches. These concern the size and complexity of the systems being modeled, on one hand, and the limitations of the measurement technology, one other hand: both pose difficulties to the underlying methods for structure and parameter identification. Along these lines, we will discuss the problems faced in modeling biological and ecological systems, related to the parameter identification task and, consequently, the structure identification task within the framework of automated modeling.

The task of (re)constructing the behavior of biological and ecological systems[1], including their components, structure, interactions and dynamics, from time course data, is at the heart of the emerging fields of systems biology (Klipp et al., 2005) and ecological modeling (with systems ecology) (Jørgensen and Bendoricchio, 2001; Jopp et al., 2011), respectively. In order to obtain a system-level (holistic) understanding of the underlying processes in the observed systems, both systems biology and ecological modeling rely on building, simulation, and analysis of mathematical models. Due to the advanced measurement methodologies, there is a rise in the amount and the quality of experimental data[2]: proper integration of these data into models of a complete system can provide a global view of the underlying mechanisms (Klipp et al., 2005).

However, many specific challenges are to be considered when dealing with parameter identification in systems biology and ecological modeling. Above all, biological and ecological systems exhibit complex nonlinear dynamic behavior, which is often modeled using ordinary differential equations (ODEs). Typically, these ODE models are nonlinear, have no analytical solutions, and require solutions that rely on numerical simulation. Numerical solutions of nonlinear ODEs can be computationally expensive, especially for stiff model dynamics[3] and systems with many states. Due to the implicit definition of the cost function (typically sum of squared errors) model simulation has to be performed for every candidate parameter solution (combination of parameter values) investigated during the optimization process, which in turn makes the complete parameter identification process extremely computationally expensive.

In case of a large number of parameters, as encountered in mechanistic models, the corresponding optimization problem may be hard to solve: the cost function has numerous local minima combined with a flat landscape, meaning that gradient information is useless and local search method can fail. Moreover, the parameters have to be estimated from measurements that are often sparse and imperfect due to noise, and the studied system can often be only partially observed. All of these constraints can lead to identifi-

[1]Ecological system, or shortly ecosystem, is a biological system of living organisms and their physical environment, interacting as a functional unit.

[2]Time course data generated by high-throughput experimentation contain a large amount of information on the cell dynamics at different levels, including genomic, proteomic, metabolic, and physiological level.

[3]The solutions of a stiff model consist of both rapidly and slowly varying components.

ability problems, *i.e.*, the inability to uniquely identify the unknown model parameters, making parameter identification an even more difficult optimization problem (Marsili-Libelli, 1992; Omlin and Reichert, 1999; Marsili-Libelli et al., 2003; Gutenkunst et al., 2007; Balsa-Canto et al., 2010).

Unfortunately, there is no well established identification method for the type of models discussed above. While the modeled systems are typically linearized or discretized in time for analysis purposes in systems theory (Ljung, 1987), this is not a desirable approach for biological and ecological systems as the physical meaning of the parameters would be lost. In this respect, parameter identification is recognized to be among the most challenging tasks as well as the computational bottleneck in the process of modeling biological and ecological systems (Chou and Voit, 2009; Jopp et al., 2011).

Parameter identification assumes that the model structure is given. In practice, therefore, this step is invoked after the structure is being specified. Typical approaches in system identification assume that the structure is specified either by a domain expert (modeler) or is chosen from some general class of model structures. The first approach is knowledge-driven, concerned with building comprehensible, mechanistic models by combining domain-specific (expert) and general scientific knowledge. In contrast to the first, the second approach is data-driven. The modeler chooses the structure from some general class of structures, such as linear equations or polynomials, that is believed to be adequate, and estimates the parameters based on the given data. In an iterative process, the structure is modified until a good fit to the data is obtained. Typically, the models obtained by the second approach do not necessarily reveal the processes governing the modeled system's dynamics. In any case, a major portion of the modeler's time is devoted to one of the two approaches.

As the space of possible structures is practically infinite, the standard (manual) structure identification problem is extremely difficult and time consuming. Motivated by this, computational methods for automated modeling of dynamic systems have been developed to simultaneously tackle both structure and parameter identification (Todorovski and Džeroski, 1997, 2003). One way to approach both tasks simultaneously originates from the area of machine learning (Langley, 1996; Mitchell, 1997), in particular *equation discovery* (Langley et al., 1987; Džeroski and Todorovski, 2007). Equation discovery is concerned with developing computational methods for automated modeling of quantitative laws and models, in the form of equations, from measured data. At the early stage, equation discovery was focused on rediscovery of well-known scientific laws in the the form of algebraic equations. As the equation discovery methods evolved, their focus shifted from rediscovery to establishment of new quantitative laws and models from measured data, moving towards automated knowledge-driven modeling of real-world systems (Džeroski and Todorovski, 2007).

The complexity of the automated modeling task depends directly on that of the parameter identification task. Namely, the size of the search space of possible model structures is can be very large. Their evaluation (in terms of how well they describe the system dynamics) requires estimation of their parameters. While state-of-the-art equation discovery methods limit the space of equation structures by integrating modeling knowledge related to the specific system and relevant domain of interest, the search in this space is guided by heuristics (typically) related to the degree of fit of the model to the data. The latter reflects the quality of the output of the parameter identification task, and consequently depends on the performance of the optimization method used to estimate the model parameters. The majority of the equation discovery approaches perform LS parameter estimation by gradient-based nonlinear optimization (Džeroski and Todorovski, 2007), such as Algorithm 717 (Bunch et al., 1993), and have difficulties to cope with the mul-

timodal and discontinuous error landscapes typical for nonlinear dynamics. As a result, the success of automated modeling strongly depends on the efficiency and effectiveness of parameter identification.

1.2 Hypothesis and aims

This dissertation will consider the problem of parameter identification in models represented by nonlinear ODEs from observed behavior(s) of the modeled system, in form of time course data. Our hypothesis is formulated as follows.

H. *Meta-heuristic optimization methods can be successfully used for parameter identification in nonlinear ODEs under a wide range of conditions. More specifically, they can deal effectively with large numbers of parameters, partial observability and noise in the data, for different systems modeled and different ODE simulation methods. We expect better performance (as compared to local optimization methods), both in the specific context of an individual model structure and in the context of structure identification, where a large number of model structures is considered.*

To prove our hypothesis, we set a number of more specific aims for our research, defined as follows.

A1. *Formulate parameter identification in nonlinear ODE models as a global black-box optimization problem and approach it with meta-heuristic optimization methods.*

A2. *Conduct a thorough and fair empirical evaluation of the performance of different optimization methods, both local and global (different meta-heuristics).*

A3. *Investigate the performance of different optimization methods on the parameter identification problem in two representative ODE models, describing the dynamics of two real-life systems in the domains of systems biology and ecological modeling.*

A4. *Investigate the influence of limited observability of the modeled system dynamics on the parameter identification task.*

A5. *Investigate the influence of ODE simulation methods on the parameter identification task.*

A6. *Investigate the influence of parameter identification (using global vs. local optimization) on the model structure selection step within the process of automated modeling of dynamic systems.*

1.3 Methodology

In order to prove the hypothesis stated above, we employ the methodology outlined below. For each of the above aims, we briefly describe the approach taken to achieve it.

A1. *Formulate parameter identification in nonlinear ODE models as a global black-box optimization problem and approach it with meta-heuristic optimization methods.*

Due to the nonlinear and constrained dynamics of real-world system, parameter identification can be formulated as a nonlinear programming problem with

differential-algebraic constraints (Moles et al., 2003). The latter falls into the domain of global optimization problems. Traditionally, nonlinear programming problems have been solved by using derivative-based methods, such as steepest-descent or Quasi-Newton methods, and so-called direct-search methods, such as the Nelder-Mead simplex algorithm (Nocedal and Wright, 1999). Most of these methods fail to cope with complex and large-scale nonlinear problems, due to the the difficulties posed by the ruggedness of the search landscape. Modern optimization approaches, such as those from the emerging field of meta-heuristic optimization (Talbi, 2009), are designed to cope efficiently and effectively with different types of optimization problems. Related work in the domain of nonlinear system identification (Nelles, 2001; Pintér, 2001), including the fields of systems biology (Chou and Voit, 2009; Sun et al., 2012) and ecological modeling (Athias et al., 2000; Zhang et al., 2009), has shown that to overcome the above mentioned difficulties, parameter identification should be addressed by global (stochastic) optimization methods.

To this end, we will apply meta-heuristics, as general-purpose solvers of high-dimensional black-box[1] nonlinear optimization problems. In addition to the well-established differential evolution (DE) (Storn and Price, 1997) and particle swarm optimization (PSO)(Kennedy and Eberhart, 1995), we introduce the usage of a recently proposed swarm-based meta-heuristics, the differential ant-stigmergy algorithm (DASA) (Korošec, 2006) for parameter identification in ODE models. Motivated by the fact that DASA has shown promising results in solving large-scale continuous global optimization problems (Korošec et al., 2010, 2012), we expect its performance on the challenging task of parameter identification in nonlinear ODE models to be competitive to the performance of the other two meta-heuristics. As a baseline method for the experimental comparison, we use Algorithm 717 (A717), a derivative-based method essentially designed for nonlinear LS estimation.

A2. *Conduct a thorough and fair empirical evaluation of the performance of different optimization methods, both local and global (different meta-heuristics).*

The considered meta-heuristic methods will be integrated in the same computational framework and will use the same platform for ODE integration. The experimental design and comparison methodology will be the same for the two different parameter identification tasks addressed. All optimization methods will be given the same time budget, in terms of the number of cost functions evaluation, related to the size of the problem, *i.e.*, the number of parameters to be estimated. The optimization methods will be compared with respect to their convergence performance, the best solutions found and their reconstructed observed (and complete) system dynamics. We will use both artificial data, simulated with different levels of Gaussian noise, and data from real measurements. The use of the artificial data will allow us a more controlled study of the influence of noise and observability on the performance of the parameter identification task (optimization methods). In addition, in order to obtain the best performance of the meta-heuristic methods for the given problem, the control parameters defining their operational behavior will be tuned (for the parameter identification task in the endosome maturation model).

[1]Problems for which the cost function cannot be mathematically formulated. This means that methods, such as derivative-based methods, which exploit the structure of the cost function can not be applied by default.

A3. *Investigate the performance of different optimization methods on the parameter identification problem in two representative ODE models, describing the dynamics of two real-life systems in the domains of systems biology and ecological modeling.*

The performance of the optimization methods will be tested on two real-life problems. The first model, relevant for the domain of systems biology, concerns an important cellular process, *i.e.*, endocytosis. More precisely, the model captures the dynamics of a key endocytotic regulatory system that switches from cargo transport in early endosomes to cargo disintegration in mature endosomes (Zerial and McBride, 2001; Rink et al., 2005). The regulatory system is based on the process of conversion of Rab5 domain proteins to Rab7 domain proteins. Using both a theoretical and an experimental approach to model this process, Del Conte-Zerial et al. (2008) show that a cut-out switch ODE model structure provides the best fit to the biological observations.

The second model concerns the dynamics of an aquatic food web, relevant for the domain of ecological modeling. Food webs represent a set of interconnected food chains by which energy and materials circulate within an ecosystem. Aquatic food webs are especially challenging due to their complexity, as they are organized in many levels and include a variety of organisms, each playing a specific role within its environment. These roles are dependent on the habitat as well as the environmental variables within the ecosystem, such as temperature, dissolved oxygen concentration, and human disturbances, such as pollution and overfishing. The problem with the food webs is preserving their balance: the smallest changes can have severe detrimental effects on the organisms within the ecosystem. In this respect, we consider a model of the food web dynamics in Lake Bled (Slovenia), describing the dynamics of one nutrient (phosphorus) and two populations, (phytoplankton and the zooplankton species *Daphnia hyalina*).

The tasks of parameter identification in these models are representative and challenging due to the characteristics of the modeled dynamics and the measurement processes. The example models are nonlinear (due to the nonlinearity of the behavior of the modeled system) and have many parameters (high dimensionality). The measurements available are sparse (food web of Lake Bled) or incomplete (endosome maturation), and imperfect (due to measurement noise). These properties make the parameter identification tasks at hand challenging optimization problems, calling for the use of advanced optimization methods.

A4. *Investigate the influence of limited observability of the modeled system dynamics on the parameter identification task.*

Due to the physical (or cost) limitations of the measurement methodology, typical for the domain of systems biology and ecological modeling, the observations of the modeled system dynamics can be sparse, noisy and incomplete. We will study the effect of incomplete (and misinterpreted) observations of the system dynamics on the complexity of the parameter identification task, as well as the applicability and performance of the four different optimization methods in this context. We encounter these observability problems with the task of modeling the endosome maturation dynamics, due to the limited measurability of the concentrations of the Rab5 and Rab7 domain proteins in the endosome: the different (*i.e.*, active and passive) states of these proteins can not be distinguished in the measurement process.

In order to study the effect of different kinds of limited observability, we will define four different observation scenarios and generate artificial data for each of them. The scenarios will cover a wide range of situations, from the simplest one of complete observability, where the concentrations of all protein states are assumed to be directly measurable, to the most complex (and realistic) scenario, where the observations are limited to the total concentrations of each protein in its different states. We will test the performance of the selected optimization methods in the different observation scenarios and compare the ability of different methods to cope with them. A final set of experiments will be performed on real-experimental data in order to check the validity of the results obtained on artificial data.

A5. *Investigate the influence of ODE simulation methods on the parameter identification task.*

In this context, we consider two approaches for numerical integration, teacher-forcing simulation and full simulation. Teacher forcing simulation makes supervised predictions one time step ahead, while full simulation makes unsupervised predictions, where the predictions for each time step is based on the prediction for the previous time step (and in general the longer time periods history of predictions). Unlike full simulation, teacher forcing simulation assumes complete observability of the system: the one-step-ahead prediction are calculated based on the measured data at the current time step. Its advantage is in the simple trapezoidal integration schema, computationally cheaper than the complex multistep variable-coefficient integration schemas, such as the backward differentiation method, used in full simulation. The question here is how the noise in the data will affect the teacher forcing and full simulation, and consequently the parameter identification task. In this respect, we use both real-experimental data and artificial pseudo-experimental data with varying amounts of noise. Due to the complete observability requirement, this analysis will be performed only for the parameter identification task in the food web model of Lake Bled.

A6. *Investigate the influence of parameter identification (using global vs. local optimization) on the model structure selection step within the process of automated modeling of dynamic systems.*

One major drawback of existing automated modeling approaches (such as LAGRAMGE, LAGRAMGE 2.0, IPM, HIPM, for an overview see Džeroski and Todorovski (2010)) is the use of local search methods for parameter identification. In this context, we will investigate the effects of using DASA as a global optimization method instead of the previously used local optimization method A717. For this purpose, we will devise an extensive experimental setup, which will consider eight tasks for automated modeling of phytoplankton dynamics in Lake Bled from single-year data measured in eight different years.

1.4 Contributions

The research presented in this dissertation addresses important aspects of parameter identification in the context of both single-model identification and multiple-model identification arising within automated modeling of dynamic system from experimental data. The corresponding research findings are published in several conference and journal publications, such as (Tashkova et al., 2010b, 2011a, 2012b; Čerepnalkoski et al., 2011a): the

complete list of related publications is given in *Appendix 1*. In the following, we summarize the main contributions of the work presented in this dissertation.

- **An overview of existing approaches for parameters identification** and their application to the class of problems of our interest (nonlinear ODE models in the domain of systems biology and ecological modeling) is presented. Given that parameter identification in ODE models of nonlinear dynamic systems belongs to the class of nonlinear optimization problems, methods for nonlinear optimization are surveyed. Furthermore, an overview of the most promising meta-heuristics and their application to the parameter identification problem is provided as well.

- **A thorough empirical evaluation of representative meta-heuristic methods is performed on the task of estimating parameters in nonlinear dynamic (ODEs) models.** In this respect, four global-search meta-heuristic algorithms for continuous optimization, *i.e.*, the differential ant-stigmergy algorithm (DASA) and its continuous variant (CDASA), particle-swarm optimization (PSO), and differential evolution (DE), as well as Algorithm 717 (A717), a derivative-based local search method, are considered for comparison. These are applied to representative tasks of estimating parameters in the field of systems biology and ecological modeling.

  - **Identification of parameters in ODE models of two practically relevant real-life systems, *i.e.*, endosome maturation in endocytosis and the Lake Bled food web**. The first systems is a key cellular process that switches from cargo transport in early endosomes to cargo disintegration in mature endosomes within the process of phagocytosis, through which bacteria are eliminated from human cells. Understanding the underlying mechanism in this case is important for properly treating bacterial infections in living organisms (in the conetxt of drug development). The second system represents the food web dynamics in a representative aquatic ecosystem, *i.e.*, Lake Bled (Slovenia). Understanding the food web mechanism is important for preserving the natural ecosystem in the lake (environmental protection).
  - **The performance of the methods on the considered tasks is evaluated along a number of metrics, including the quality of reconstructing the system dynamics as well as speed of convergence, both on real-experimental data and on artificial pseudo-experimental data with varying amounts of noise** (Tashkova et al., 2011a, 2012b). The evaluation shows that the meta-heuristic global optimization methods for parameter identification are clearly superior and should be preferred over local optimization methods. While the differences in performance between the different methods within the class of meta-heuristics are not significant across all conditions, DE yields the best results in terms of the quality of the reconstructed (observed) system dynamics as well as the speed of convergence.
  - **The impact of limited observability on the difficulty of the parameter identification task (and consequently on the performance of different optimization methods) is evaluated in the context of reconstructing endosome maturation dynamics** (Tashkova et al., 2011a). The observability of the system (as varied through the observation scenarios - where data of different completeness and accuracy of interpretation are given as input), shows a strong influence on the success of the parameter identification, where

less complete observations make the optimization task much more difficult. Worst results are obtained when the observations are misinterpreted. Furthermore, the observation that a good model with respect to the observed system dynamics does not necessarily mean a good match of the complete system dynamic, clearly shows the importance of choosing a relevant cost function when the modeled system dynamics is partially observed.

  - **The impact of the ODE simulation method on the parameter identification task is evaluated in the context of reconstructing Lake Bled dynamics** (Tashkova et al., 2012b). The experimental evaluation showed that the use of teacher forcing simulation (one-step trapezoidal-based integrator for supervised predictions) makes parameter identification much faster, while the use of full simulation (multistep variable-coefficient integrator for unsupervised predictions) produces much better parameter estimates from real-experimental data. In the case of artificial (noisy) data, however, there is no clear difference between the models obtained based on teacher forcing and full simulation: the obtained models are of comparable quality as long as the noise in the data is not very high. This result opens the opportunity to significantly reduce the computational time of the overall parameter identification process by combining teacher forcing with full simulation in some kind of a hybrid schema for model simulation.

- **Empirical analysis of the interplay between parameter identification and model structure selection within the process of automated modeling of dynamic systems**. More precisely, we empirically evaluated the influence of the parameter identification method (global - DASA, vs. local - A717 optimization method) on the final output of automated modeling (with ProBMoT) phytoplankton dynamics in Lake Bled from measured data. The experimental evaluation on eight single-year modeling tasks empirically proved the benefit of estimating model parameters by global optimization methods for the model (structure) selection process (Čerepnalkoski et al., 2011a,b). The latter is furthermore important in the context of modeling long term system dynamics, since automated modeling of the Lake Bled food web has so far focused mostly on discovering single-year models, as considered here as well.

1.5 Organization

This chapter has provided an introduction to the material presented in this dissertation. It has formulated the problem of interest and outlined the methodology to approach it. In addition, it has specified the aims of the research work and summarized its main contributions. This section closes the introduction by providing a brief outline of the structure of the doctoral dissertation.

Chapter 2 introduces the essentials of the mathematical modeling task, with a focus on the task of parameter identification. It provides an overview of the existing approaches to parameter identification, in general, and in the case of nonlinear dynamic systems described by ODEs in particular. Related to the latter, important issues and challenges are discussed in the context of modeling biological and ecological systems, providing references to relevant related work. This chapter further outlines the concept of automated modeling of dynamic systems (from the machine learning point of view) and the relevance of the parameter identification task within the framework of automated modeling.

Subject to the formulation of the parameter identification task as a nonlinear optimization problem, Chapter 3 formally introduces the problem of optimization. It provides a brief overview of the existing types of optimization problems and discusses the characteristics of difficult optimization problems. Furthermore, it gives a survey of the existing methods for solving nonlinear optimization problems, the class of problem of our interest. Finally, it provides a brief description of a representative local optimization method used as a baseline algorithm in our empirical analysis.

This dissertation addresses the parameter identification task by using meta-heuristic optimization methods. In this context, Chapter 4 is devoted to meta-heuristic optimization. It summarizes the most important properties and elements of meta-heuristic design, and includes a relevant classification of the existing meta-heuristics. The focus of the chapter then narrows to two most successful population-based meta-heuristics paradigms, *i.e.*, evolutionary and swarm intelligence algorithms. As representative meta-heuristic methods from both classes are used to conduct our parameter identification experiments, their description is provided as well. Finally, the chapter address the problem of tuning meta-heuristics, *i.e.*, selecting the right values of the optimization method parameters that give the best performance, and outlines the tuning approach considered in this study.

Chapter 5 describes the experimental methodology common for both case studies considered. It provides details about the Monte Carlo-based approach for practical parameter identifiability. It also outlines the methodology for performance comparison of the selected optimization methods on the considered parameter identification tasks.

Chapter 6 addresses the task of parameter identification in a practically relevant model of endocytosis, *i.e.*, the life-cycle of endosomes. The model is nonlinear and has many, potentially unidentifiable, parameters which have to be estimated from partial observations. In this context, the chapter provides and discusses the results of the conducted experimental analysis regarding the influence of limited system observability on the success of parameter identification from artificial data (both noise-free and with different amounts of noise) and real-experimental data, using different (meta-heuristic) optimization methods.

In a similar way, Chapter 7 addresses the task of parameter identification in a practically relevant model of the food web dynamics in Lake Bled. The typical model is nonlinear with many parameters to be estimated from sparse and noisy data. The chapter presents and discusses the findings of two experimental studies. The first performs experimental comparison of different (meta-heuristic) optimization methods considered in conjunction with two model simulation approaches (teacher forcing simulation and full simulation) and two types of data (real observations and artificial data with different amounts of artificial noise) in the context of parameter identification in a single model structure. The second study investigates the benefit of using global (over local) optimization methods within the process of automated modeling of aquatic ecosystems, in this case Lake Bled, which involves estimating parameters in many model structures.

Finally, the conclusions drawn from the experimental studies, including an overall summary of the dissertation and its original contributions, as well as possible directions for further work are given in Chapter 8.

2 Parameter identification

Mathematical models are powerful tools that allow us to increase the knowledge about the system being studied, to predict and control the future behavior of the system, or to test hypotheses that can hardly be tested in reality. Nevertheless, identifying the model structure along with the model parameters is not an easy problem when modeling complex dynamic systems (such as biological and ecological systems) from measured data, a task referred to as *system identification* in engineering (Ljung, 1987). Roughly, system identification is a coupled two-step methodology that involves *structure identification* and *parameter identification* of mathematical models of dynamic systems from measured data.

The focus in this dissertation is on the second task, that is, *parameter identification*, also known as *parameter estimation*, *parameter fitting* or *model calibration*[1]. Given a model structure and measured data, the goal of parameter identification is to estimate the unknown model parameters that will determine a best possible fit between the measurements and model predictions. It is usually approached as an optimization task of minimizing a cost function that measures the goodness of fit of the model simulation to the measured data (Janssen and Heuberger, 1995; Nelles, 2001; Schittkowski, 2002; Banga, 2008).

Parameter identification is one aspect of a larger problem, called the *inverse problem*, which also includes structural and practical identifiability of the unknown model parameters. The latter are concerned with the problem of unique identification of the unknown model parameters. The inverse problem is a challenging task in mathematical modeling of dynamic systems in general, and in the field of systems biology and ecological modeling, in particular. Namely, biological and ecological systems exhibit complex nonlinear dynamic behavior, which is often modeled using ordinary differential equations (ODEs). Typically, these ODE models are nonlinear and have many parameters, do not have analytical solutions, and their solution relies on numerical simulations that can be computationally expensive. In addition, the measurements are sparse and imperfect due to noise, and the studied system can often be only partially observed. All of these constraints can lead to identifiability problems, making parameter estimation a difficult optimization problem (Marsili-Libelli, 1992; Omlin and Reichert, 1999; Marsili-Libelli et al., 2003; Gutenkunst et al., 2007; Balsa-Canto et al., 2010).

The remainder of the chapter is organized as follows. Section 2.1 introduces the essentials of the mathematical modeling task, defining basic concepts, classes of models, and the important steps in the process of building mathematical models. Section 2.2 focuses on the crucial step of parameter estimation, in general, and in the case of nonlinear dynamic systems described by ODEs, in particular. Furthermore, it discusses properties of the task of parameter estimation in models of biological and ecological systems, with appropriate references to the relevant related work. Section 2.3 explains the problem of parameter identifiability, more precisely, practical parameter identifiability as used in this study. Finally, Section 2.4 discusses the concept of model selection and automated modeling of dynamic systems, and their relation to the parameter estimation task.

[1]We will mainly use the first two terms throughout the dissertation.

2.1 Mathematical modeling

Modeling is a conceptual tool widely used in science and engineering to formalize the abstract notions of reality. It provides a framework (*e.g.*, for applying mathematics and logics) that can be evaluated and reused for reasoning in a wide range of situations. In fact, *models* can be seen as simplified representation of complex reality, that can help us understand, analyze, control or predict the behavior of the considered real processes and phenomena, in general referred to as the *system*. Based on the questions we want to answer regarding the investigated system and the purpose of the model use, we can choose among different models, such as conceptual, mathematical, non-numerical, predictive/descriptive, and quantitative/qualitative.

Here, we will focus on the mathematical representation of real systems by mathematical models, derived from scientific principles or/and empirical observations of the systems. Formally, *mathematical models* are an aggregation of equations (or, in general, logical and computational structures) and numerical data, aiming to describe the behavior of the studied physical system in quantitative terms (Walter and Pronzato, 1997; Dochain and Vanrolleghem, 2001; Bellomo et al., 2008; Velten, 2009; Klee and Allen, 2011). A very important aspect of mathematical models is that they can reveal the processes that govern the behavior of the observed system. Ultimately, it is impossible to exactly describe all aspects of the system behavior due to the complexity of real systems. However, mathematical models that are able to adequately describe the most important aspects have already shown to be very useful tools for both scientists and engineers.

Another important aspect of modeling is to distinguish between the system being modeled and the model itself. The system is unique, while the mathematical model may assume different forms and complexity under different operating conditions. Regardless of the details included into the mathematical model, the model nevertheless represents an incomplete and inexact picture of the observed system. Finally, mathematical modeling is not an exact science. It is an iterative decision making process that relies on a combination of intuition, experience, empiricism, and the application of scientific laws of nature, balancing between complexity and usefulness of the obtained model (Gershenfeld, 1999; Klee and Allen, 2011).

2.1.1 Concepts and classification

In general, a mathematical model of a system is a rule that transforms certain quantities called *inputs* (known a priori or measured from the system) into quantities called *outputs* by some defined relationship. The outputs are the quantities that are of interest to the modeler/user and the inputs are the quantities that affect the outputs (the system behavior) in the form of disturbances and manipulations that are usually under our control. The output of the model should resemble the actual values as observed in the system. Note, however, that due to imperfect measurement procedures the inputs and the outputs actually measured in the system are subject to some noise. Here, we will assume measurement noise (and no systematic noise) that affects only the system output in additive form (as depicted in Figure 2.2 in Section 2.2).

The structure of the model is defined by the specific relationship between the input and the output, which is represented in the form of equations. These equations consist of *variables* and *parameters*. Variables are entities of the model such as states, inputs and outputs. While inputs are independent, states and outputs are completely determined based on the values of the parameters and the inputs. Parameters are model entities that tend to be unknown (usually cannot be measured experimentally) and need to be

estimated from prior knowledge or measured data. In the most general case, model parameters can evolve over time. Here, we will assume that the parameters are time-invariant, but can have different values depending on the specific application. Variables and parameters can be arranged in different ways to form the final equations leading to different types of models. Below we provide and briefly discuss possible classifications of mathematical models (Walter and Pronzato, 1997; Dochain and Vanrolleghem, 2001).

Input-output *vs.* state-space models. An input-output model involves a relationship that combines the inputs, outputs and parameters into a single equation. The state-space model explicitly includes the internal states of the system, as additional variables, through the so-called state equation. The output, modeled by the output equations, is given in terms of a combination of the current system states, and the current system inputs. These two equations form a system of equations collectively referred to as a state-space model.

Dynamic *vs.* static models. Models in which the variables evolve over time are called dynamic, as opposed to the models, called static or steady-state, in which variables are constant in time.

Discrete *vs.* continuous models. This classification can be with respect to time (default) or space, and it is directly related to the model being formulated by difference/differential equations. While discrete changes are modeled with difference equations, continuous changes are modeled by differential equations. *Ordinary differential equations* (ODEs) are most often used to model system behavior that changes continuously with respect to a single variable, such as time, while *partial differential equations* (PDEs) are used to capture changes in the system behavior with respect to many variables/dimensions, such as time and space. Space-dependent models, *i.e.*, PDEs, are also referred to as *distributed-parameter models*.

Linear *vs.* nonlinear models. Linearity is a model property related to the structure of the model equations. Models can be linear or nonlinear with respect to their variables and/or parameters. The linearity of the model is an important property that determines the way the model can be solved. For linear models, analytical solutions can be obtained, while nonlinear models require numerical techniques to solve the model equations.

Deterministic *vs.* stochastic models. This classification takes into account the uncertainty encapsulated in the model. Namely, deterministic models neglect all aspects of uncertainty (in the input variables, parameter values and model structure) and their output is uniquely determined by the initial conditions (only for state-space models) and the input, as opposed to the stochastic models with outputs described by a probability density function. Note that models based on ODEs and PDEs are purely deterministic. Nevertheless, the differential equation formalism can be adapted to represent probabilistic models, as done, *e.g.*, in the formalism of stochastic differential equations (Gershenfeld, 1999).

Phenomenological *vs.* mechanistic models. Phenomenological models are constructed using no prior information about the system studied, that is, they assume that there is no knowledge of physical or internal relationship of the system inputs and outputs other than that the input should yield observable response in the output. Parameters in these models have no physical meaning. Basically, the system being studied is seen as a

black box and it is modeled from empirical data only, an approach known as *empirical* or *data-driven modeling*. In this context, phenomenological models are also referred to as *black-box*, *empirical*, *data-driven*, or *heuristic models*.

On the other hand, mechanistic models encapsulate certain information about the internal mechanism of the system, that is, their structure is based on physical, chemical, and biological laws. Deriving models from prior domain knowledge is referred to as *knowledge-driven* or *theoretical modeling.* In the literature, mechanistic models are also referred to as *physical*, *knowledge-based*, or *white-box models*. In fact, white-box models are models for which all the necessary information for the system is available. However, most of the mechanistic models can be located somewhere between the extreme black-box and white-box cases, belonging to the category of *gray-box* or *semi-empirical models*, since they are derived as a combination of the knowledge-driven and empirical modeling approach.

In sum, phenomenological models are universally applicable, easy to set up and typically require much less computing time and resources, but have limited scope. The mechanistic models are easy to physically interpret, allow deeper insights into the system performance and better predictions. However, they demand a priori information and usually are computationally more expensive to simulate.

2.1.2 Building mathematical models

Building mathematical models is an iterative decision-making process (Gershenfeld, 1999). The typical model building cycle as considered in the area of systems identification (Ljung, 1987; Walter and Pronzato, 1997; Dochain and Vanrolleghem, 2001) is depicted in Figure 2.1. It involves four basic steps: *problem definition*, *prior knowledge and data collection*, *model identification*, and *model validation*.

The building process starts with a clear definition of the problem given the purpose of the model (goal), followed by a step of collecting available information on the specific problem, that is, prior knowledge from the literature and domain experts, including preliminary or already published experimental data. It proceeds with the model identification step, the heart of the mathematical modeling task, which basically involves structure identification and parameter identification.

Once the model framework is defined (*e.g.*, the type of the model, specification of the variables/parameters and their ranges), a model structure is proposed. Given the structure of the model, the next step is the estimation of the unknown model parameter from the available data. The result of the estimation task is an initial working (fully specified) model that can be further analyzed and tested.

The last step of the building exercise involves model validation, the highest-level of model quality assessment. It tests whether the given model is a correct representation of the system for which it is intended. Thus, the initial model must be validated with new experimental data. The latter in most cases will reveal a number of deficiencies. If the model fails the validation step, a new model structure and/or a new experimental design must be planned, and the outlined procedure is repeated iteratively until the validation step confirms that a satisfactory model is obtained.

Essentially, inferring models from measured (input/output) data, involves a decision on the model structure (functional form) and determination of the adjustable parameters of the selected model using a given behavior (data). Structure identification plays a critical role in modeling, since it defines the choice available for the selection of the "best model". In practice, the model structure is usually given by a domain expert and reflects prior knowledge and/or experimental data. If no data are used to identify the struc-

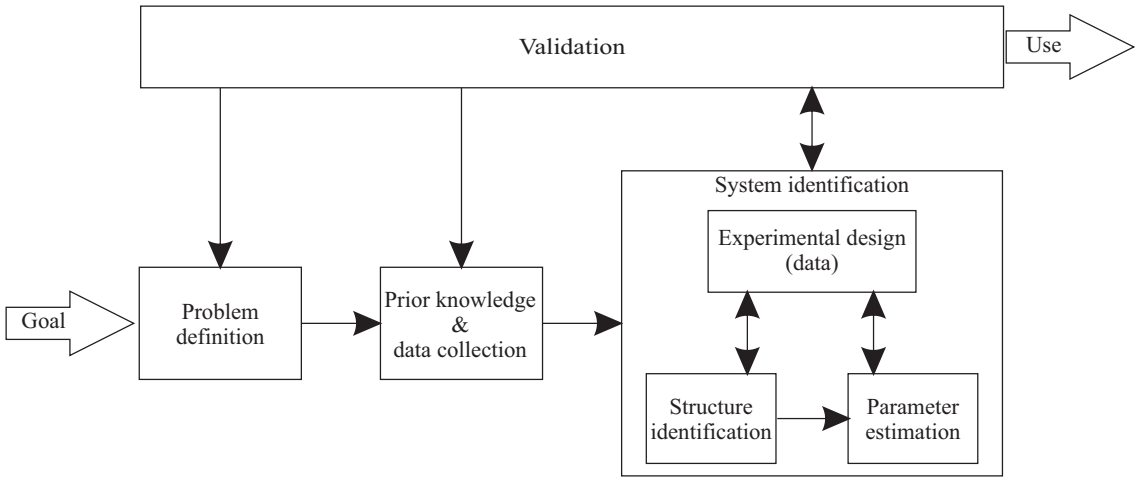


Figure 2.1: **A conceptual diagram of the model building cycle.** Boxes represent different steps in the building process, while arrows represent the flow of information during the process.

ture of the model then the process is referred to as knowledge-driven identification. In contrast to the knowledge-driven identification, the empirical identification of the model structure is based only on data. However, given that knowledge and data are available for the modeled system, most probable structure identification scenario is a combination of both approaches. Unlike structure identification, parameter estimation is almost exclusively data-driven, *i.e.*, for known input-output relations, the values of parameters are determined by fitting the model predictions to the experimental data.

Model validation, the procedure of assessing the goodness of the model for its purpose, if possible, should be based on the performance of the model with respect to unseen data – measurements obtained for conditions other than those used for estimation of the model parameters (*e.g.*, longer time periods). Often, it is required that the model not only quantitatively fits the data, but is also able to qualitatively match the data, *i.e.*, the shape of the data. In contrast to parameter estimation, that mainly relies on quantitative fits to the given data, validation is usually qualitative (visual inspection of the model predictions) and based on the modeler's judgement.

The success of the modeling process is tightly coupled with the quality and quantity of the experimental data, which is directly connected with the important sub-step of *experimental design*. Experimental design aims at identifying experimental conditions adapted to the final purpose of modeling, *i.e.*, to discriminate between competing model candidates in the context of model selection or structure identification, to validate a given model and to improve the accuracy of the estimated model parameters (Walter and Pronzato, 1997; Dochain and Vanrolleghem, 2001). The optimal experiment design should provide the most informative data for minimal (or at least reduced) experimental costs (Balsa-Canto et al., 2010). The problem is usually formulated as a process of optimization of the experimental conditions with respect to a defined objective function, *i.e.*, optimality criteria, that express the goal of the experimental design. Traditionally, experimental designs that aim to improve the accuracy of the estimated parameters are based on minimal-variance (or, equivalently, maximal-information) criteria, algebraically formulated as functionals of the eigenvalues of the Fisher information matrix[1] (FIM), while the ones that aim at structure identification (better model discrimination) are based on

[1]Fisher information matrix is the inverse of the error covariance matrix of the estimated parameters.

maximization of the disagreement in the predictions of the competing model candidates (Walter and Pronzato, 1997; Dochain and Vanrolleghem, 2001).

Here, we will focus on the identification step, in particular the inverse problem, which involves parameter estimation and parameter identifiability. The following two sections will cover these two topics.

2.2 Parameter estimation in nonlinear dynamic models

To assess the quality of a certain model, we need to simulate its behavior. Model simulation, on the other hand, requires that the model $m(c)$ is completely specified, that is, the structure m and the values of the parameters c are known. Model simulation is also known as the *direct problem*, as opposed to the inverse problem.

Assuming a model structure m, which includes a set of D adjustable parameters $c = \{c_1, \ldots, c_D\}$, was provided by the structure identification step, and moreover, $m(c)$ is a sufficiently accurate representation of the modeled system, parameter estimation aims at determining the optimal values c^{opt} of the unknown model parameters that reproduce the observed system behavior, *i.e.*, in the measured data d, in the best possible way. By the term "unknown", we here refer to the model parameters that are expensive or impossible to measure experimentally, or cannot be determined based on the available domain knowledge.

By default, the above formulation of the parameter estimation task does not imply a specific model structure or even a modeling formalism. However, the systems (processes) observed in biology and ecology have mainly a dynamic nature, and moreover, have complex nonlinear dynamics, which is usually represented by (ordinary) differential equations (ODEs) or differential-algebraic equations (DAEs). Here, we will assume that the studied dynamic systems are represented with nonlinear continuous-time deterministic state-space models, *i.e.*, ODEs, obtained usually by knowledge-driven modeling. In this context, one section is devoted to ODE models and their simulation.

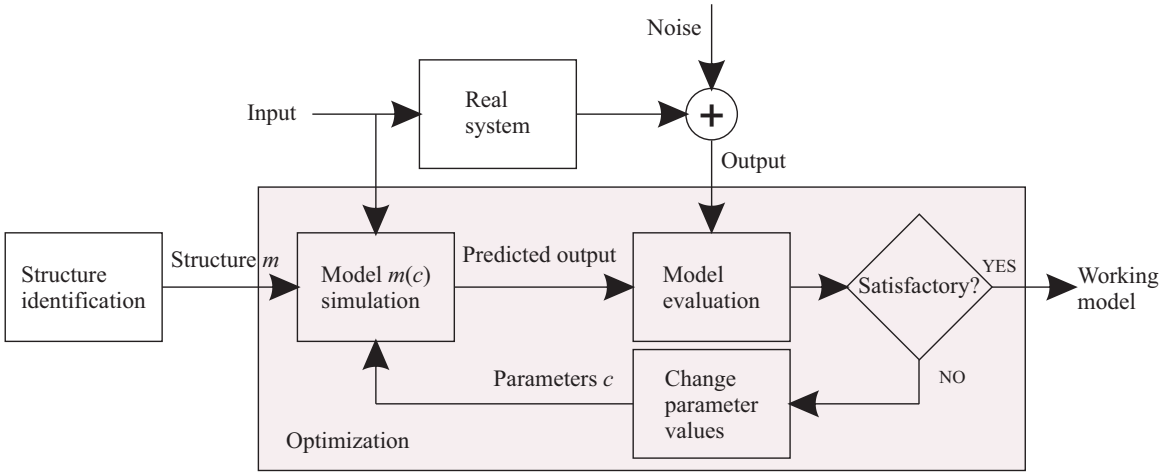


Figure 2.2: **A conceptual diagram of the parameter estimation process.** The process is formulated as a problem of optimization of a certain cost function that accounts for the quality of the fit of the model to the data with respect to the unknown model parameters. Boxes represent the modeled system, the model itself and the optimization process. Arrows represent the possible flow of information in the estimation process.

For complex nonlinear dynamic models, as considered here, the estimation of the model parameters is an iterative adjustment procedure, that can be formulated as optimization (minimization) of a certain criterion, cost (objective) function $J(m(c), d)$, that quantifies the goodness of the model, given the data d, with respect to the adjusted model parameters c. Figure 2.2 depicts the parameter estimation task as an optimization process.

The choice of the cost function is the first important problem associated with the parameter estimation task: it should reflect the aim of the model. The most widely used cost function is the ordinary least-squares estimator formulated as a simple sum of squared errors between the model predictions and the measured data (Ljung, 1987). Although different cost functions can be found in the literature, conceptually, their formulation is driven by two main estimation paradigms: one assumes that the model parameters have unique fixed values, while the other treats the parameters as random variables with a certain probability distribution. In the following section, we will briefly outline both, and focus on the least-squares estimator as used in this study.

2.2.1 Parameter estimation approaches

There are two broad classes of approaches to the parameter estimation task: the *frequentist* (referred to as the "classical" or point-estimation) approach and the *Bayesian* (probabilistic) approach (Rice, 2007; Samaniego, 2010). The most representative approach of the first class is *maximum-likelihood* (ML) estimation, according to which the most likely parameter values are the ones that maximize the probability (likelihood) of observing the given data. Unlike ML estimation, which does not need any external information about the parameters, Bayesian estimation treats the parameters to be estimated as random variables, with a prior distribution representing the knowledge about the parameter values before taking the data into account.

A special case of maximum-likelihood estimation, based on the assumption of independent and normally distributed errors in the experimental data, leads to the well-known approach of *least-squares* (LS) estimation. According to the information that the end-user has to provide, LS estimation is the simplest approach, while the Bayesian approach is the most complex (demanding) one (Jaqaman and Danuse, 2006; Rice, 2007). But, according to the a priori assumptions included in the estimation process, LS estimation is the most constrained approach, while the Bayesian approach is the most flexible (relaxed). The relationship between the Bayesian and ML estimation approaches is depicted in Figure 2.3: all of the listed approaches can be theoretically deduced from the Bayesian approach (Nelles, 2001).

Bayesian estimation treats both data d and unknown model parameters c as random variables, with prior distribution $p(d)$ and $p(c)$, respectively. The prior distribution $p(c)$ represent our prior beliefs (what we know) about the parameters before observing the data, while $p(d)$ represents the prior knowledge about the data. The dependence of the measured data d on the parameters c can be expressed as the conditional probability density function, $p(d \mid c)$, often referred to as the *likelihood* function. The likelihood measures the match of the data and the model predictions given the parameters c. Based on the above formulation, Bayesian estimation combines prior knowledge of the model parameters with measured data to obtain an updated posterior distribution of the model parameters. This updating is performed using the Bayes' rule, expressed as

$$p(c \mid d) = \frac{p(c)\ p(d \mid c)}{p(d)} = \frac{p(c)p(d \mid c)}{\int_c p(c)p(d \mid c)dc}, \tag{2.1}$$

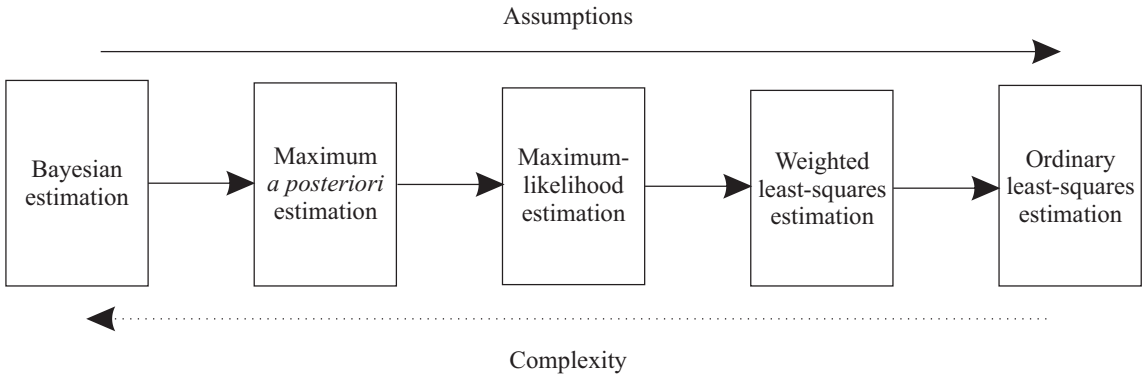


Figure 2.3: **Approaches to the parameter estimation task.** The different approaches are represented by boxes, while their relationship with regard to two criteria, assumptions and complexity in terms of demanding prior information, is represented by arrows. The Bayesian estimation is the most general and most complex approach, while ordinary least-squares estimation is the most simple and intuitive approach for parameter estimation. The maximum a posteriori, maximum-likelihood, weighted least-squares, and ordinary least-squares estimation are special cases of the Bayesian approach subject to specific (and increasingly stringent) assumptions regarding the prior knowledge of data and model parameters.

where $p(c \mid d)$ is called the posterior distribution and expresses the probability of the parameter values after observing the data. Because the denominator in Eq. (2.1) is a normalizing constant, the Bayes' theorem is often expressed as

$$p(c \mid d) \propto p(c)\ p(d \mid c), \tag{2.2}$$

indicating that the prior expectations are modified by the likelihood function to yield posterior beliefs. Because, Eq. (2.1) does not have a closed form solution, the posterior distribution is inferred by stochastic simulation, such as the Gibbs sampler (Chen et al., 2000).

As already discussed, Bayesian estimation delivers as output the complete distribution of the parameters values for the given data and prior knowledge. However, if specific parameter estimates are required, the posterior can be analyzed in different ways to obtain a parameter point estimate. For example, the mode (the most likely value) of the prior distribution leads to the *maximum a posteriori* (MAP) estimator (Eq. (2.3)) (Walter and Pronzato, 1997).

$$\max p(c \mid d) \tag{2.3}$$

Maximum-likelihood estimation quantifies the intuition that, given certain data, some values of the model parameters are more likely than others, as they result in model predictions that closely resemble the data. Here, the likelihood of the parameters is equated to the probability of observing the available data from a process that is represented by the model tested. The latter is defined as a function of the measurement errors in the data, *i.e.*, differences between the model prediction (data predicted by the model) and the experimentally observed data. These differences arise from model inadequacy and/or measurement noise. Assuming that the model structure is correct, in ML estimation one can take into account the available information about the noise into the design of the cost function. Subject to the above formulation, ML parameter estimates are the values of the model parameters that maximize the probability of observing the data (Eq. (2.4)).

$$\max p(d \mid c) \tag{2.4}$$

Given an infinitely large data sample, ML estimation yields (asymptotically) unbiased and normally distributed parameter estimates with a minimal variance (Rice, 2007). Unlike Bayesian estimation, ML estimation assumes that the model parameters are constants, *i.e.*, have unique values, while the data are a random sample drawn from the universe of datasets. Finally, note that if we assume constant priors on the parameters, *i.e.*, all parameters are equally likely before the data are observed, then the MAP estimation is practically an ML estimation.

Least-squares estimation delivers the best estimates that minimize a quadratic cost function, *i.e.*, the *sum of squared errors* between the model prediction and the experimentally observed data. Due to its simple and intuitive formulation, it is widely used in mathematical modeling.

The most general form of least-squares estimation is known as weighted least-squares estimation. The estimation includes positive real-valued weighting coefficients w_{ij} in the cost function, that express the relative confidence in the experimental data, *i.e.*, the importance of the i-th measured output at the j-th measurement time point for the model performance (Eq. (2.5)).

$$\mathrm{SSE} = \mathrm{SSE}\left(m(c), d\right) = \mathrm{SSE}(Y, \hat{Y}) = \sum_{i=1}^{M}\sum_{j=1}^{N} w_{ij}\left(Y_i[j] - \hat{Y}_i[j]\right)^2 \tag{2.5}$$

Here the observed data are defined as $d = \{Y_i[j], 1 \leq i \leq M, 1 \leq j \leq N\}$, where $Y_i[j]$ is the value of the i-th measured output at the j-th time point, M is the number of measured outputs, N is the number of samples per observed output, and $\hat{Y}_i[j]$ is the value of the i-th output at the j-th time point, as predicted by the model $m(c)$.

In our work, we chose the weighting coefficients to be $w_{ij} = Y_i[j]^{-2}$, provided $Y_i[j] \neq 0$, which makes the error relative and improves the fit of small system output values (Eq. (2.6)). This choice is justified especially in the case of simultaneous fitting of system outputs with amplitudes of different magnitudes. This is the case with the food web dynamics in Lake Bled, which we consider as a case study.

$$\mathrm{SSE}_\mathrm{r} = \sum_{i=1}^{M}\sum_{j=1}^{N}\left(\frac{Y_i[j] - \hat{Y}_i[j]}{Y_i[j]}\right)^2 \tag{2.6}$$

If all data points are equally contributing to the sum, then the weighted LS estimation maps to the ordinary LS estimation (Eq. (2.7)).

$$\mathrm{SSE}_\mathrm{o} = \sum_{i=1}^{M}\sum_{j=1}^{N}\left(Y_i[j] - \hat{Y}_i[j]\right)^2 \tag{2.7}$$

Formally, LS estimation can be defined as ML estimation subject to uncorrelated Gaussian measurement errors e in the data with constant (unknown) variance σ, which we assume does not depend on the parameters c. In this case, the likelihood can be constructed as a product of N Gaussian probability density functions, corresponding to the N observations in the dataset, as expressed by Eq. (2.8).

$$p(d \mid c) = \prod_{i=1}^{N} p(d_i \mid c) = \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{{e_i}^2}{2\sigma^2}) \tag{2.8}$$

Maximizing likelihood is the same as maximizing the log-likelihood due to the monotonicity of the logarithmic function. The latter, because of the negative sign in the exponential functions, maps into minimization of the sum of squared errors (Eq. (2.9)).

$$\max_c \ln p(d \mid c) = \max_c \ln \prod_{i=1}^{N} p(d_i \mid c) \approx \max_c -\frac{1}{2} \sum_{i=1}^{N} \frac{{e_i}^2}{\sigma^2} \approx \min_c \sum_{i=1}^{N} {e_i}^2 \tag{2.9}$$

Alternative approaches

Robust estimation (Staudte and Sheather, 1990) includes estimators that have robust performance with respect to deviations from the idealized assumptions for which the estimator is optimized. For instance, LS estimation is based on a Gaussian model for the measurement error: if the data are contaminated with outliers (data drawn from other distributions than the Gaussian) then LS will perform poorly. Among the existing robust estimators, M-estimators are particulary suitable for the parameter estimation task. In fact, M-estimators follow the ML principle as well, but unlike LS estimators, they try to reduce the effect of the outliers by replacing the squared error in the cost function by another function of the error, yielding

$$\min \sum_{i=1}^{N} \phi(e_i), \tag{2.10}$$

where ϕ is a symmetric, positive-definitive function with a unique minimum at zero, chosen to grow slower than the square function. Note that the M-estimator where $\phi(e) = e^2$ is the LS estimator.

Recursive estimation is based on methods for state estimation as considered in the framework of control theory. These can be used for estimation of model parameters in models of nonlinear dynamic systems, if the latter are formulated as state-space models (Voss et al., 2004). Most of these approaches, *e.g.*, the Kalman filter and the unscented Kalman filter (Gershenfeld, 1999), are based on a *prediction-correction schema* for estimating the temporal dynamics of the unobserved states of a dynamic system: predict the state at the time $t + \triangle t$ ($\triangle t$ is a fixed sampling interval) based on its value at time t, and then correct this estimate by taking the new measurement at time $t + \triangle t$ into account. In this setup, parameters can be estimated by using the augmented state vector approach, which considers the parameters as additional state variables with a zero rate of change.

While recursive estimation approaches are widely used in engineering, they have been only recently considered for estimation model parameters in the context of biological or biochemical systems (Lillacci and Khammash, 2010). Note that, for weakly nonlinear systems, recursive estimation becomes ML estimation.

2.2.2 ODE models and simulation

A model based on ODEs defines the temporal change of a set of *system* variables S (also referred to as endogenous variables or states in the state-space model formulation) as a function of the variables S and a set of *exogenous* variables E. The exogenous variables E are the input variables on which the model depends and appear on the right-hand side of the ODEs only, while the system variables S are dependent variables, the behavior of which is being modeled, and appear both on the left-hand side and the right-hand side of the ODEs. The ODE model of the observed system takes the form

$$\frac{d}{dt} S = F\left(S(t), E(t), c\right), \tag{2.11}$$

where t represents time, $\frac{d}{dt}S$ represents the time derivatives of the system variables S, F asserts the structure of the ODEs, and c is the set of model parameters. Such an ODE model, given the values of system variables S at the initial time point t_0, $S(t_0)$ and the values of the exogenous variables E on the observed time interval $[t_0, t_{N-1}]$, can be simulated to obtain the values of the system variables $S(t)$ in the time interval $(t_0, t_{N-1}]$. This formulation of ODEs defines the so-called *initial-value problem*, as opposed to the *boundary-value problem* that requires the values of the system variables at the bounds of the observed time interval (Press et al., 1992; Gershenfeld, 1999). Here, the focus is given only to the initial-value problem. However, note that one general approach to solving the boundary-value problem is based on converting it into an equivalent initial-value problem, that is then solved by a trial-error approach using so-called *shooting methods* (Press et al., 1992).

Note that, although by c we denoted the unknown model parameters, in practice, c can be extended to include the initial conditions of the system variables (which can be unknown as well). This will be the case with the task of parameter estimation in the endocytosis model addressed in Chapter 6.

The ODE model represented by Eq. (2.11) captures the behavior of the system variables S, which do not directly correspond to the output variables Y, represented by the observed data d. In general, the output Y of the ODE model at a time point t is modeled as

$$\hat{Y}(t) = G(S(t), E(t)). \tag{2.12}$$

In the simplest observation scenario, the values of all the system variables are directly observed (measured), *i.e.*, $\hat{Y} = S$. While this can happen in some cases, as in the case of the specific food web model for Lake Bled considered in Chapter 7, this is more of an exception than a rule: in most real-world cases in ecology and biology, we can not observe the values of all the system variables directly. This is mostly due to the physical limitations of the measurement methodology and the costs of performing the measurements.

Thus, when modeling real-life dynamic systems, we deal with a variety of observation scenarios with different complexities. One possible scenario corresponds to situations where only some of the system variables are directly observed, *i.e.*, $\hat{Y} \subseteq S$. In an alternative, more complex observation scenario, G is a linear function, denoting, for example, that only the sum of the system variables can be directly observed, *i.e.*, $Y = \sum_{s \in S} s$. Finally, the most complex observation scenarios involve arbitrary nonlinear functions G. The complexity of the observation scenario can drastically influence the performance of parameter estimation methods, as we have to reconstruct the complete model based on incomplete observations. Related to this, Chapter 6 discusses the the influence of the observation scenario on the performance of parameter estimation methods in the context of modeling endocytosis.

The model given by Eqs. (2.11) and (2.12) is completely defined with the values of the model parameters c. Different sets of parameter values will in general lead to different model realizations. As already stated, the goal of parameter estimation is to find the values of the model parameters specifying a model behavior that is closest to the observed system behavior with respect to the objective function. In the case of ODE models, the evaluation of the objective function is usually non-trivial, as it requires the previous calculation of the values of the system variables over a period of time. In order to find $S(t)$ and, in turn, $\hat{Y}(t)$, one has to integrate the ODE model, unless special strategies are used to avoid

the integration[1]. An analytical solution for complex nonlinear ODE integration problems does not exist in general and one has to apply numerical approximation methods for ODE integration[2].

This may be a very time consuming procedure, especially for stiff ODE models (that consist of both rapidly and slowly varying components), making the whole parameter estimation problem quite expensive and complicated (Chou and Voit, 2009). In this context, we consider two approaches for numerical integration, teacher-forcing simulation and full simulation.

The first approach considered is the so-called *teacher forcing simulation* (TF). It is inspired by the teacher forcing approach introduced by Williams and Zisper (1989) for training neural networks. It uses the measured values of the system variables at a given time point as initial values for the calculation of the system response at the next time point: this defines some kind of supervised simulation of the system. Consequently, teacher forcing simulation is possible only in the case when all system variables are observable (measured). The specific TF implementation that we use is based on the trapezoidal integration rule and can be described as follows

$$\begin{aligned} \hat{S}[t_i] &= \hat{S}[t_{i-1}] + \triangle F[t_i]\frac{\triangle t_i}{2}, \\ \triangle F[t_i] &= F\left(S[t_i], E[t_i], c\right) + F\left(S[t_{i-1}], E[t_{i-1}]c\right), \\ \triangle t_i &= t_i - t_{i-1}, \\ i &\in [1, N], \end{aligned} \tag{2.13}$$

where $\hat{S}[t_i]$ and $\hat{S}[t_{i-1}]$ are the simulated values of the system variables at the i-th and $i-1$)-th time point, respectively, while F is the function that defines the changes in the system variables (introduced in Eq. (2.11)). The changes in the system variables are calculated using the observed values of the system variables ($S[t_i]$, and $S[t_{i-1}]$) along with the observed values of the exogenous variables ($E[t_i]$, and $E[t_{i-1}]$) at the i-th and $(i-1)$-th time point, respectively.

The second one, which we refer to as *full simulation* (FS), is a standard approach for solving the initial-value problem (Gershenfeld, 1999), which needs only the initial values of the system variables. The values of the system variables at the current time point is estimated using the changes calculated with the ODE model and the simulated value of the system variables from the previous time point (typical for one-step methods, such as Euler or Runge-Kuta methods) or previous time points (typical for multistep methods, such as Adams-Moulton method) (Press et al., 1992; Klee and Allen, 2011). Full simulation in our experimental evaluation is performed with the CVODE package, a general-purpose ODE solver that uses linear multistep variable-coefficient methods for integration, *i.e.*, Adams-Moulton method (for non-stiff problems) and backward differentiation method (for stiff problems) (Cohen and Hindmarsh, 1996). More precisely, we use the backward differentiation method combined with Newton iteration and a preconditioned Krylov method, which is a powerful combination for large stiff ODE models.

Estimation of the model parameters in ODE model based on the solution of the initial-value problem is referred to as *initial-value* or *single-shooting* method for parameter estimation (Schittkowski, 2002). This method relies on the strength of the optimization

[1]For example, one can transform the problem to parameter estimation in a model of algebraic equations using approaches for slope estimation or collocation methods. Appropriate references with application regarding biological systems are outlined by Chou and Voit (2009).

[2]These methods are based on discretization of the governing equations, *i.e.*, converting the differential equations to difference equations (Press et al., 1992; Klee and Allen, 2011). For example, the Euler method is the simplest finite difference scheme.

method: if the cost function is multimodal, then parameter estimation by local gradient-based optimization methods can lead to suboptimal parameter solutions. An alternative approach is to use the *multiple-shooting* method for parameter estimation, which is said to be more stable with respect to local convergence, even if combined with local optimization methods (Schittkowski, 2002; Voss et al., 2004). The idea is to divide the observed time interval into subintervals, for which local initial values are introduced as additional parameters to be estimated. The ODE model is solved piece-wise and the cost function is evaluated in the same way as in the initial-value approach. While the model parameters are assumed to be constant over the entire time interval, the local initial values are optimized separately in each subinterval. As the piece-wise integration will lead to discontinuities in the state dynamics at the joins of the the subintervals, constraints are introduced to enforce smoothness. Thus, the problem is formulated as a constrained nonlinear optimization problem.

2.2.3 Parameter estimation in biological and ecological models

Representative methods from both ML and Baysian approaches are commonly used for parameter estimation in the domain of systems biology (Moles et al., 2003; Polisetty et al., 2006; Rodriguez-Fernandez et al., 2006a; Lawrence et al., 2007; Rogers et al., 2007; Dräger et al., 2009; Toni et al., 2009) and ecological modeling (Omlin and Reichert, 1999; Athias et al., 2000; Dowd and Mayer, 2003; Qian et al., 2003; Zhang et al., 2009; Jones et al., 2010). It is difficult to argue in favor of one class of approaches against the other in a general manner (Samaniego, 2010), since both have shown advantages in specific situations. On one hand, Bayesian approaches can elegantly treat the uncertainty in parameter values and model structure in a uniform manner. On the other hand, frequentist approaches (such as the ones considered in this dissertation) can be effectively used for high-dimensional models with large numbers of parameters.

Here, we will focus on the parameter estimation task from the frequentist point of view, in particular least-squares estimation. For probabilistic parameter estimation we refer the reader to the work (and references therein) by Omlin and Reichert (1999); Dowd and Mayer (2003); Qian et al. (2003); Lawrence et al. (2007); Rogers et al. (2007); Toni et al. (2009); Jones et al. (2010).

Due to the highly nonlinear dynamics and the limited observability of biological systems and ecosystems, parameter estimation is a challenging and computationally expensive optimization task. Most parameter estimation tasks are multi-modal, *i.e.*, have many local optima that prohibit the use of local search methods. Furthermore, the models are often high-dimensional, making the parameter estimation task computationally complex. Finally, the measurability of systems in cell and molecular biology is highly limited. Many system variables are not directly observable. For the few that can be measured, measured data are noisy and taken at a coarse time resolution. All these constraints, combined with the complex dynamics of the considered models, can lead to identifiability problems, i.e, the impossibility of unique estimation of the unknown model parameters, making parameter estimation an even harder optimization task (Marsili-Libelli, 1992; Omlin and Reichert, 1999; Marsili-Libelli et al., 2003; Gutenkunst et al., 2007; Balsa-Canto et al., 2010).

In general, parameter estimation can be formulated as a nonlinear programming problem with differential-algebraic constraints, that falls into the domain of global optimization problems. As the problem of optimization (including an overview of the important concepts and methods) will be discussed in the following two chapters, we will outline here only related work concerning the use of different optimization methods to estimate the

model parameters in ODE/DAE models representing typical systems in systems biology and ecological modeling.

Biological models. The identification of the unknown model parameters from observations, given the model structure, is one of the main issues of computational systems biology (Goel et al., 2008; Ashyraliyev et al., 2009; Chou and Voit, 2009; Sun et al., 2012). Due to the size and complexity of the modeled systems (such as signaling, genetic and metabolic pathways), the problem has been approached with different optimization methods in different setups, including different cost functions, artificial and/or real data. Classical parameter estimation is based on least-square estimation by gradient-based methods, such as Gauss-Newton and Levenberg-Marguardt: for example, Levenberg-Marguardt method was applied to estimate the parameters in a metabolic network model (Marino and Voit, 2006). A comparison of these methods in the context of pathways model is given by Mendes and Kell (1998); del Rosario et al. (2008).

Depending upon the degree of system nonlinearity and the initial starting point, gradient-based methods run the risk of getting trapped in local optima (Mendes and Kell, 1998). Therefore, global optimization methods have been proposed to cope with the above mentioned challenges of parameter estimation in biological models (Moles et al., 2003). While there are successful application of deterministic global methods, *e.g.*, parameter estimation in a metabolic pathway model by the branch-and-reduce method (Polisetty et al., 2006), they tend to be very time-consuming and highly computationally intensive for practical estimation problems in systems biology with realistic problem sizes.

Recent related work is almost completely devoted to the use of stochastic global optimization methods and hybrid methods (that combine global and local search methods), especially meta-heuristics, due to their ability of handling black-box optimization problems and relatively quick convergence to the vicinity of global optima (Moles et al., 2003; Rodriguez-Fernandez et al., 2006a; Dräger et al., 2009; Sun et al., 2012). Related work includes numerous applications of different type meta-heuristics. For example, genetic algorithms have been used for parameter estimation in signaling and metabolic pathway models (Tominaga et al., 2000; Arisi et al., 2006; Modchang et al., 2008). Next, evolutionary strategies and their hybrids have been successfully applied for parameter estimation in biochemical pathway models (Moles et al., 2003; Rodriguez-Fernandez et al., 2006b; Balsa-Canto et al., 2008). Furthermore, differential evolution and its hybrids, considered to be one of the most successful meta-heuristics, have been used for parameter estimation in biochemical models (Moles et al., 2003; Balsa-Canto et al., 2008; Dräger et al., 2009; Liu and Wang, 2010). Swarm intelligence methods have been applied to parameter estimation problems as well, *e.g.*, particle swarm optimization in a metabolic network model (Dräger et al., 2009) and ant-colony optimization in S-system models (Zuniga, 2011). In addition, scatter search methods have shown to be very efficient and effective for estimation of parameters in biological network models (Rodriguez-Fernandez et al., 2006a). Finally, various modification of simulated annealing have been used for parameter estimation in biological networks as well, *e.g.*, S-systems models (Gonzalez et al., 2007) and synthetic gene networks (Braun et al., 2005).

For further details, the interested reader can consult the paper by Sun et al. (2012) that provides a comprehensive review of the application of meta-heuristics to parameter estimation problems in systems biology (including some applications for inference of the topology of biological networks). Furthermore, Chou and Voit (2009) discuss the recent developments in mathematical modeling of biological systems, as considered within the biochemical systems theory framework, and focus on parameter estimation and structure identification of S-systems and generalized mass action models.

Ecological models. Classical approaches used for parameter estimation in ecological models include local optimization methods, such as derivative-based methods (*e.g.*, the Quasi-Newton method used for calibration of a phytoplankton model in the work by Mocenni et al. (2008)) and direct-search methods (*e.g.*, parameter estimation in well-known ecological models with a flexible method based on polyhedron search by Marsili-Libelli (1992)). In order to deal with the problem of convergence to local optima, these have been restarted from different initial parameter solutions, *e.g.*, Atanasova et al. (2006) used an augmented Quasi-Newton method plus multiple restarts for parameter estimation in population dynamic models (within a procedure for automated modeling of aquatic ecosystems).

However, restarts are not a reliable solution of the typical ill-posed parameter estimation task. Therefore, global optimization approaches, deterministic and stochastic, that are more robust regarding the dimensionality and the landscape characteristics of the search space, should be preferred for parameter estimation in ecological and environmental models, especially recent methods on meta-heuristic optimization. An example application of a deterministic approach is the parameter estimation in a groundwater model with Lipschitzian global optimization (Russell Finley et al., 1998). There are numerous applications of genetic algorithms, *e.g.*, calibration of water quality models (Mulligan and Brown, 1998), calibration of phytoplankton dynamics for lake Kasumigaura in Japan (Whigham and Recknagel, 2001), and calibration of an ecosystem model of lake Kinneret in Israel (Gilboa et al., 2009). There are also some applications of genetic programming to the calibration of lake ecosystems models (Cao et al., 2008). Particle swarm optimization has been used for the calibration of water quality models (Afshar et al., 2011). Simulated annealing has been applied to the calibration of marine ecosystem models (Matear, 1995). Some comparisons of global (meta-heuristic) optimization methods have been also performed in this context, *e.g.*, calibration of an oceanic biogeochemical model (Athias et al., 2000) and calibration of a hydrological model (Zhang et al., 2009).

2.3 Parameter identifiability

The problem of uniqueness of the estimated parameters in a given model is related to the issue of parameter identifiability. We can distinguish between structural and practical identifiability. Structural identifiability is a theoretical property of the model structure, depending only on the model input (stimulation function) and output (observation function): it is not related to the specific values of the model parameters. The parameters of a given model are structurally globally identifiable, if they can be uniquely estimated from the designed experiment under the ideal conditions of noise-free observations and error-free model structure (Walter and Pronzato, 1997). If the model is not structurally identifiable, one should consider reformulating the model.

Even when we deal with a structurally identifiable model, it can still happen that the model parameters can not be uniquely identified from the available experimental data. In this case, we experience a practical identifiability problem, related to the amount and quality of available experimental data. Practical identifiability analysis can also help us to assess the uncertainty of the parameter estimates and to compare possible experimental designs without performing experiments. Parameters uncertainties (confidence intervals) may be computed by using, for example, the Fisher information matrix (FIM), the Hessian matrix of the objective function, or a Monte Carlo-based approach. For further details on this topic, we refer the reader to the work by Marsili-Libelli et al. (2003) that addresses this issues for ecological models, or the work by Balsa-Canto et al. (2010) that formalizes

parameter identification for biological models. Moreover, Miao et al. (2011) provide a comprehensive review of methods for identifiability analysis in nonlinear ODE models, including structural, practical and sensitivity-based identifiability analysis.

We are going to assess the practical identifiability of the parameters in the considered models using the Monte Carlo-based sampling approach (Joshi et al., 2006; Balsa-Canto et al., 2010). The approach estimates the expected uncertainties of the parameters by re-applying the parameter estimation method to a large number of replicate datasets generated by using different realizations of the chosen level of artificial (experimental) noise. In this way, we generate a cloud of parameter estimates that represents the confidence region. Based on this cloud of solutions, we can obtain the distribution (represented by histograms) of values for each uncertain parameter (as well as its mean value and standard deviation) and perform correlation analysis to determine the most correlated parameters. In contrast to the FIM-based approach, that assumes a linearization of the model, the Monte Carlo approach estimates are reliable also for highly non-linear models or very large parameter uncertainties. The generality of the Monte Carlo approach, however, comes at a high computational cost.

If model parameters are estimated by the augmented state vector approach, identifiability can be approached in terms of another system property called *observability* in control theory, which considers whether the state variables can be inferred from the measured (input/output) data (Geffen et al., 2008). However, if model parameters are estimated within a probabilistic framework, such as Bayesian inference (Omlin and Reichert, 1999; Jones et al., 2010), there is no need for post-hoc calculation of the parameter uncertainties as they are provided with the estimation itself. Finally, note that non-identifiability of model parameters does not imply a poor fit of the model to the data, but means that unique values cannot be associated with the model parameters. Therefore, the predictive power of the model will be limited to simulations (model predictions) insensitive to the non-identifiable parameters.

2.4 Model selection and automated modeling

Likelihood methods provide a good inference framework if the model structure of the studied system is given. In practice, the model structure is not known in advance, and has to be carefully defined given previous knowledge and/or experimental data. The structure identification step is concerned with determination of the model structure, *i.e.*, functional relationship between the model variables and parameters, and its complexity, *e.g.*, model terms or basis functions, number of system variables and parameters. As the space of all possible structures is practically infinite, the task of structure identification is to provide the "best" model structure(s), selected among the different model structure candidates based on the degree of fit of the model to the data. This basically amounts to the *model (structure) selection* problem.

2.4.1 Model selection

Model selection is the task of choosing the "best" model from a given set of models $\mathcal{M} = \{m_1, \ldots, m_k\}$ in light of the evidence provided by the observed data d. It offers a way to draw inferences from a set of multiple competing hypotheses. The candidate models, $m_1, \ldots, m_k$, each representing one hypothesis, are simultaneously evaluated in terms of support from the observed data d: the model that is best supported by the data is chosen among the candidate models.

There is no unique notion of what a "best" model is: *ultimately, the strongest useful statement is that the best model is the one that works best for you* (Gershenfeld, 1999). However, one of the most commonly accepted notions of quality is based on the fundamental principle of parsimony, according to which the best model provides the simplest adequate representation of the observed data, or translated to the statistical modeling framework, the model that provides the best bias-variance error tradeoff (Gershenfeld, 1999; Burnham and Anderson, 2002). Namely, the "true" model (structure and parameters) of the observed system is unknown, and every model chosen from the space of model candidates approximates the real system up to some error. This error comes from two sources: *bias error* reflects the unmodeled dynamics of the system due to the limited complexity of the selected model structure, while *variance error* reflects the uncertainty of the estimated parameters due to limited data, noise in the data and overparametrization.

In this context, model selection aims for the most accurate and parsimonious model, balancing between *underfitting* (which means that the structure is not flexible enough to capture the regularity in the data) and *overfitting* (which means that structure is too complex and fits the noise in the data or introduces artifacts not present in the data).

Different formal tools from both frequentist and Bayesian schools, such as F tests for nested[1] models, Akaike information criterion (AIC), minimum description length, exhaustive search, stepwise selection, backward/forward selection, cross-validation, various-type Bayes factors, Bayesian information criterion (BIC), and Bayesian model averaging, have been proposed for comparing and selecting models in the context of data (Kadane and Lazar, 2004). Common to all of them is that they are based to some extent on the principle of parsimony and implicitly incorporate some bias-variance tradeoff. For example, the most famous and widely used information criteria, such as AIC (Akaike, 1974) given in Eq. (2.14) and BIC (Schwarz, 1978) given in Eq. (2.15), select the model that minimizes a penalized maximum-likelihood.

$$\min_{m,c} -2 \ln p(d \mid c) + 2D \tag{2.14}$$

$$\min_{m,c} -2 \ln p(d \mid c) + D \ln N \tag{2.15}$$

The first term reflects the model accuracy and measures how well a particular model $m(c)$ (including the estimated parameters c) fits the data sample d with size N. It basically reflects the result of the ML parameter estimation task, which in practice most often amounts to LS estimation subject to uncorrelated Gaussian measurement errors in the data. However, maximum likelihood often overfits the data because it does not reflect the model complexity. Thus, the second additive term plays the penalty role and punishes the complexity (overparametrization, *i.e.*, number of parameters D) of the model being considered. While the first term gets better as the model gets more complex, the second term gets worse as the model gets more complex. Essentially, selecting the best model according to the penalized maximum-likelihood criteria formulates the problem of model selection as an optimization problem across a set of model candidates.

2.4.2 Automated modeling of dynamic systems

Even when the processes governing the internal system dynamics being modeled are well understood (as a result of existing domain knowledge), the functional forms and the

[1]Model m_1 is nested in model m_2 if it can be derived from model m_2 as a special case, setting specific values of one or more parameters.

precise values of the model parameters are often unknown. Furthermore, the space of candidate (ODE) models for dynamic systems is large enough (or infinite) to prohibit manual examination by human modelers in any systematic way. Thus, computational methods have been developed to assist in identification of models of dynamic systems from experimental data.

Automated modeling of dynamic systems simultaneously tackles both structure identification and parameter estimation (Todorovski and Džeroski, 1997, 2003). One way to approach both tasks simultaneously originates from the area of machine learning (Langley, 1996; Mitchell, 1997), more specifically *computational scientific discovery* and in particular *equation discovery* (Langley et al., 1987; Džeroski and Todorovski, 2007). Computational scientific discovery is concerned with developing computational methods that automate or support some aspects of scientific discovery, while equation discovery is concerned with developing computational methods for automated modeling of quantitative laws and models, in the form of equations, from measured data. At the early stage of its development, the field of equation discovery focused on the rediscovery of well-known scientific laws in the form of algebraic equations. As the equation discovery methods evolved, their focus shifted from rediscovery to establishment of new quantitative laws and models from measured data, and ultimately, automated modeling of real-world systems (Džeroski and Todorovski, 2007).

State-of-the-art equation discovery approaches, such as CIPER (Džeroski et al., 2003; Todorovski et al., 2004; Peckov et al., 2008), LAGRAMGE (Todorovski and Džeroski, 1997), LAGRAMGE 2.0 (Džeroski and Todorovski, 2002; Todorovski and Džeroski, 2003, 2006), IPM (Bridewell et al., 2008) and HIPM (Todorovski et al., 2005), are concerned with establishing and refining comprehensible models of real-world dynamic systems from measured data and domain knowledge. Common to all of them is the integration of formalized domain-specific knowledge[1] about modeling dynamics into the process of model induction from data with the aim to constrain the huge space of candidate model structures. The type of domain knowledge considered is closely related to the space of model structures explored: CIPER focuses on learning models represented by polynomial algebraic equations using subsumption constraints[2], LAGRAMGE searches the space of candidate models specified using a context-free grammar, while LAGRAMGE 2.0, IPM, and HIPM learn process-based models. Note that IPM and HIPM directly search the space of process-based models, while LAGRAMGE 2.0 is an upgrade of LAGRAMGE that learns process-based models by transforming process-based domain knowledge into grammars and applying LAGRAMGE in turn. Typically, these approaches perform heuristic search in the space of candidate ODE structures, guided by heuristics related to the degree of fit of the model to the data, *e.g.*, some form of LS or penalized[3] LS criteria with a preference towards simpler models. The majority of them perform LS parameter estimation by a gradient-based nonlinear optimization method, *i.e.*, Algorithm 717 (Bunch et al., 1993) and invoke numerical ODE integrators for full model simulations.

[1] In order to use domain knowledge in the equation discovery process, it must be appropriately coded. Different formalisms for representing knowledge for ODE model discovery can be used. These include constraints for polynomial models, context-free grammars, and process-based models and knowledge (Džeroski and Todorovski, 2010).

[2] Simple constraints on the search space of polynomial models include limit on the degree of the polynomials and numbers of terms involved. More complex subsumption constraints take into account a specific domain knowledge to constraint the search, such as a specific term to appear as part/subpolynomial of the polynomial model being learned.

[3] They include model complexity as measured, *e.g.*, by the length of the mathematical expression in terms of the numbers of variables, parameters and operators.

Among the stated approaches (formalisms), process-based modeling stands out as generally applicable and accessible to mathematical modelers and scientist, who typically use the notion of a process to explain and describe the behavior of an observed dynamic system. For a detailed overview and further references on this topic, the reader is referred to Džeroski and Todorovski (2007, 2010). In the remainder of this section, we will introduce the automated modeling system ProBMoT used for modeling the phytoplankton dynamics in Lake Bled as presented in Chapter 7.

ProBMoT (process-based modeling tool) is a recently developed tool for automated modeling of dynamic systems that follows the process-based modeling paradigm (Čerepnalkoski et al., 2011a,b). ProBMoT approaches modeling by combining domain knowledge about the given system with experimental data. The general domain knowledge about the considered class of systems is formulated in a library of domain knowledge. Using the components of the library and a conceptual model of the system at hand, which serves as a constraint on the space of possible model structures, candidate model structures are generated. Then, a parameter optimization method uses the available data to estimate the numerical parameters of each model structure resulting in a completely specified candidate model. The model whose behavior matches the measurements best is the result of the automated modeling process. The procedure of automated modeling with ProBMoT is depicted in Figure 2.4.

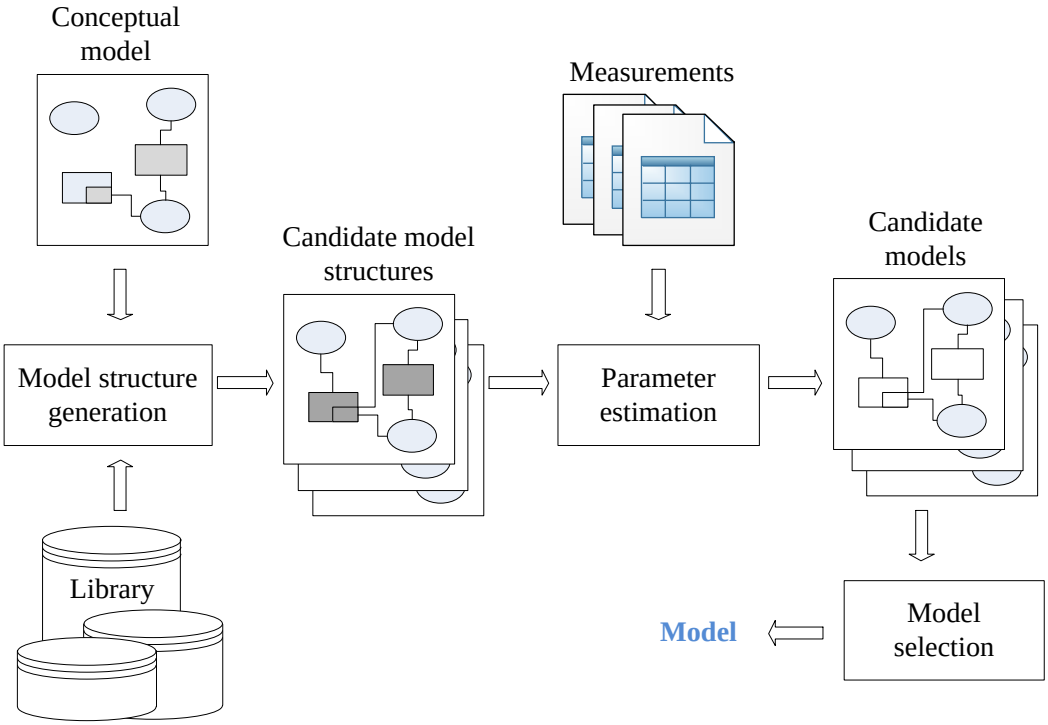


Figure 2.4: **A high-level conceptual diagram describing the process of automated modeling with ProBMoT.**

The existing version of ProBMoT performs an exhaustive search in the space of candidates models, constrained only by the conceptual model: all generated models fit the conceptual template in terms of the number and type of processes they are composed of. However, the model complexity of a specific model structure is determined by the specific functional form of the process types. A specific candidate model is obtained by ordinary LS parameter estimation subject to full simulation (CVODE). The same LS criteria is

used to rank and select the best model. Consequently, the model selection output basically relies on the parameter estimation output. Note, however, that ProBMoT is designed to be flexible and take into account different user-defined modeling task specifications: for example, arbitrary objective functions (and not necessarily LS error) or different output functions related to the observability of the modeled system dynamics can be handled by ProBMoT.

The task of automated modeling of dynamic systems is computationally expensive as the parameters of thousands of models have to be estimated in subsequent cycles. Moreover, the search in the structure space is guided by heuristics related to the degree of fit of the model to the data. The latter reflects the quality of the output of the parameter identification task, and therefore depends on the performance of the optimization method used to estimate the model parameters. As stated above, the majority of equation discovery tools employ LS parameter estimation by a gradient-based nonlinear optimization method (Džeroski and Todorovski, 2007), which by default implies difficulties in coping with multimodal and discontinuous error landscapes typical for nonlinear dynamics. Therefore, the success of the automated modeling process strongly dependents on the efficiency and effectiveness of the parameter identification method.

3 Optimization

Optimization is concerned with finding the "optimal" (or best possible) feasible solution for a given problem. Optimization problems are encountered in many domains: from science, through engineering, to business and management. Since reality is too complex for humans to cope with the problem of decision making, optimization is becoming an integral part of every-day life.

The roots of optimization can be traced back to the calculus of variations and the work of Euler and Lagrange. However, most of the progress in modern optimization theory and practice has been made in the last 60 years, initiated by the development of the well-known simplex algorithm for linear programming in the late 1940s (Dantzing, 1951). Optimization is also called *mathematical programming*, which from today's perspective is a confusing name, given that the term "programming" today refers almost exclusively to computer software. But, back then in the 1940s when the term was coined, the meaning was inclusive, with connotations of problem formulation, algorithm design and analysis.

The problem of identification of ODE model parameters from experimental data subject to the model dynamics and constraints, the main topic of this dissertation, is a typical optimization problem, that in general belongs to the class of nonlinear programming problems. Traditionally, nonlinear programming problems have been solved by using derivative-based methods, such as steepest-descent or Quasi-Newton methods, and so-called direct-search methods, such as the Nelder-Mead simplex algorithm (Nocedal and Wright, 1999). However, most of these methods fail to cope with complex and large-scale nonlinear problems, due to the the difficulties posed by the ruggedness of the search landscape. Modern optimization approaches, on the other hand, such as those from the emerging field of meta-heuristic optimization (Talbi, 2009), are designed to cope efficiently and effectively with different types of optimization problems.

The remainder of this chapter is organized as follows. A mathematical formulation of the optimization problem along with a brief overview of the existing types of optimization problems is presented in Section 3.1. Next, Section 3.2 discusses the problem features that make a given optimization problem a difficult problem. Further, Section 3.3 gives a short overview of existing methods for solving nonlinear programming problems. Finally, Section 3.4 gives a brief description of a representative local optimization method, used as a baseline algorithm in our empirical analysis.

3.1 Definition and classification

Mathematically formulated, optimization refers to the minimization or maximization of a function subject to constraints on its variables.

More precisely, an optimization problem can be defined by a pair $(\mathcal{S}, f)$, where $\mathcal{S} = \{(x_1, \ldots, x_n) \mid x_i \in D_i \wedge x_i \text{ satisfies constraints } \Omega, i = 1, \ldots, n\} \neq \emptyset$ represents the *feasible solution space* of the problem (or simply search space of the problem), defined by

- $X = (x_1, \ldots, x_n)$, a vector of *decision variables*[1]. The size of the vector defines the dimensionality of the search space;
- $D = D_1 \times \ldots \times D_n$, *domain of the decision variables*, which can be continuous or discrete;
- $\Omega = \{g_k \mid X \in D \ \wedge \ g_k(X) \leq 0, \ \ k = 1, \ldots, n_g\} \ \cup \ \{h_k \mid X \in D \ \wedge \ h_k(X) = 0, \ k = 1, \ldots, n_h\}$, a finite set of *constraints on the decision variables*. If specified, constraints can be linear or nonlinear and may or may not be given in an explicit form. If the optimization problem is not constrained then $\mathcal{S} = D$, meaning that the complete domain is a feasible search space;

Furthermore, f is a quality criterion known as *objective function* (also called *cost*, *utility*, or *fitness* function), defined as $f : \mathcal{S} \rightarrow \mathbb{R}$. The objective function formulates the goal to achieve. It assigns to every solution of the search space $s \in \mathcal{S}$ a real number indicating its quality. Therefore, it is very important to select a good objective function.

If the previous formulation holds, then solving an optimization problem consists of finding a solution $s* \in \mathcal{S}$ such that $\forall s \in \mathcal{S}$: $f(s^*) \leq f(s)$ is satisfied in case of minimization, or $f(s^*) \geq f(s)$ is satisfied in case of maximization. The solution s^* is called globally optimal solution or just *global optimum* of the optimization problem $(\mathcal{S}, f)$. If the problem has more than one global optimum, the corresponding set $\mathcal{S}^* \subseteq \mathcal{S}$ is called the set of globally optimal solutions. The main goal in solving an optimization problem is to find the values of the decision variables that yield globally optimal solutions.

In general, the shape of the objective function for a complex optimization problem introduces optimal solutions that are optimal only in their neighborhood, but are not optimal if the complete feasible search space is considered. Due to this, the optimization process may easily get trapped in the neighborhood of these sub-optimal solutions, which brings us to the important concept of locally optimal solutions, or simply local optima. A solution $\hat{s}$ is a *local optimum* of the optimization problem $(\mathcal{S}, f)$, if there exists a neighborhood of $\hat{s}$, $\mathcal{N}(\hat{s}) \subseteq \mathcal{S}$, such that $\forall s \in \mathcal{N}(\hat{s})$: $f(\hat{s}) \leq f(s)$ holds in case of minimization, or $f(\hat{s}) \geq f(s)$ holds in case of maximization. In addition, the solution $\hat{s}$ is called *strict local optimum* if $\forall s \in \mathcal{N}(\hat{s})$: $f(\hat{s}) < f(s)$ holds in case of minimization, or $f(\hat{s}) > f(s)$ holds in case of maximization.

Many different criteria can be used to classify existing optimization problems. Some classes of optimization problems, defined according to some of these criteria, are listed below.

Univariate *vs.* multivariate optimization problems: defined in terms of the dimensionality of the search space, *i.e.*, the number of decision variables that influence the objective function.

Discrete (integer) *vs.* continuous (real-valued) optimization problems: this definition is based on the type of the decision variables (solution space). If the solutions of the integer optimization problem are permutations of finite numbers (integer-valued variables), we refer to the problem as a combinatorial optimization problem. Combinatorial optimization problems are therefore characterized by a finite set of possible solutions, unlike the infinite solution space typical of continuous optimization problems. Problems

[1]We would like to remind the reader about the different meaning of the concepts "parameters" and "variables" in the context of mathematical modeling. Model parameters become variables, as these are varied in the context of the optimization process. Model variables, on the other hand, become part of the function that is optimized.

that have solutions composed from both integer-valued and real-valued decision variables are known as mixed integer optimization problems.

Constrained *vs.* unconstrained optimization problems: this classification is based on the existence of constraints over the search space. Constrained problems can be subject to one or more equality and inequality constraints. Problems with only boundary constraints are classified as unconstrained optimization problems.

Linear *vs.* nonlinear optimization problems: this distinction relies on the nature of the involved function expressions. We refer to problem as a linear optimization problem or linear programming, if the objective function and the constraints (if defined) are linear in the decision variables. If only a single constraint or objective function is nonlinear in the decision variables, the problem is referred to as nonlinear optimization problem or nonlinear programming.

Deterministic *vs.* stochastic optimization problems: this classification is based on the nature of the decision variables. The solutions of the stochastic optimization problems cannot be specified exactly, but only with some level of confidence as a result of the uncertain (stochastic) nature of the decision variables.

Separable *vs.* non-separable optimization problems: defined regarding the existence of dependencies between the decision variables. An n-dimensional optimization problem, represented by a function of n decision variables, is said to be separable if it can be optimized in a sequence of n independent one-dimensional optimization processes.

Single-objective *vs.* multi-objective optimization problems are distinguished, based on the number of objective functions simultaneously optimized.

Static *vs.* dynamic optimization problems: this classification depends on whether the problem characteristics (*e.g.*, objective function) change or remain static over time.

In the context of the outlined classifications, this dissertation approaches the parameter estimation problem in the considered ODE models as a minimization problem that is static, single-objective, deterministic, unconstrained, nonlinear, continuous, and multivariate.

3.2 Difficult optimization problems

Understanding the difficulty of the optimization problem being studied can helps us to choose the proper method, also called *optimization method*, to solve them. Furthermore, it can help us to improve the design of optimization methods. In this respect, the analysis of the landscape of the optimization problem and its difficulty is an important aspect of meta-heuristics design (Stadler, 1995, 1996; Fonlupt et al., 1999; Reeves, 1999).

Discriminating between "*easy*" and "*hard* or *difficult*" optimization problems is essentially related to the fundamental (in optimization) concept of *convexity*. A problem is named *convex* if it can be formulated as a minimization of a convex function (or a maximization of a concave function) where the feasible search region is a convex set as well. The fundamental results in convex analysis state that a locally optimal solution of a convex problem is also globally optimal (Horst et al., 1995). This is a very important

fact: it practically means that it is sufficient to find a local solution for convex problems. Figure 3.1.a) shows an example of a convex optimization problem in a one-dimensional unconstrained domain.

In contrast to the convex problems, *nonconvex* problems may have many different local optima. The field of applied mathematics that studies extremes of nonconvex functions subject to (possibly) nonconvex constraints is called *global optimization*. Choosing the best one among the local optima in a nonconvex problem can be an extremely hard task. Below we elaborate on the notion of difficulty of optimization problems in more detail.

We will discusse issues that influence the optimization process, that is, the properties of the optimization problem that pose difficulties in the optimization process. Knowing them will help us to understand the difficulties related to the (optimization) problem of parameter estimation in nonlinear dynamic models.

The representation of a given problem along with the definition of the solution neighborhood and the objective function define the topology of the *problem landscape* or *fitness landscape* (Stadler, 1995, 1996; Fonlupt et al., 1999; Reeves, 1999). Many indicators, such as the number of local optima, the distribution of local optima, the structure of the basins of attraction, and the presence of flat regions, may be used to identify the fitness landscape properties. Below we summarize some typical properties associated with difficult optimization problems (Finck et al., 2009; Weise et al., 2009).

Multimodality. A landscape with many local optima is called *multimodal* (see Figure 3.1.b)), as opposed to a landscape that has a single optimum and is called *unimodal* (see Figure 3.1.a)). Note that the existence of numerous global optima itself is not problematic and the discovery of a subset of them can be still considered a success in many cases. The occurrence of many local optima, however, can be a serious issue, *e.g.*, it can lead to premature convergence of the optimization process.

Neutrality. It defines the *flatness* of the landscape, that is the existence of regions (called neutral areas) with equal value of the objective function. The notation of a neutral area is illustrated in Figure 3.1.c).

Ruggedness. Different landscapes differ in their ruggedness. A continuous landscape with small (average) fitness differences between the neighboring solutions is called *smooth* and is usually characterized with just a few local optima. In contrast, an unsteady landscape with large (average) fitness differences is called *rugged* and usually has many local optima. Ruggedness can be simply defined as multi-modality with steep ascents and descents in the landscape (see Figure 3.1.d)).

Deceptiveness. It defines the presence of local optima that attract the optimization process away from the global optimum. More precisely, a deceptive problem is a class of multimodal problems in which the total size of the basins for local optima is much larger than the basin size of the global optimum. The gradient of the objective function can be misleading in this case (see Figure 3.1.e)).

Epistasis. Epistasis refers to the interactions among the decision variables. It defines the amount of nonlinearity in the optimization problem. Thus, a high epistasis is characterized by a high degree of interactions of the decision variables. Moreover, ruggedness and flatness of the landscape are often caused as a result of high epistasis.

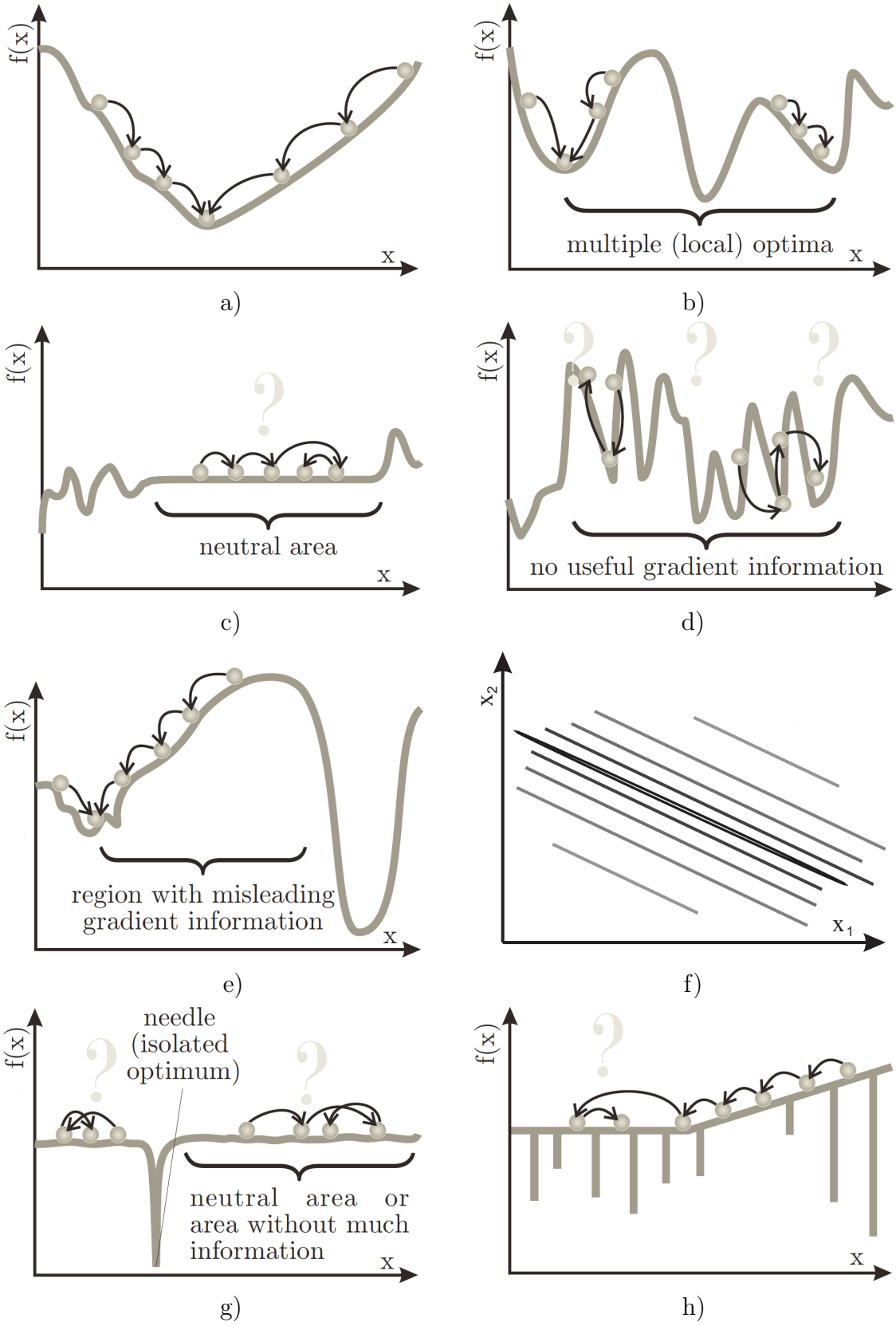


Figure 3.1: **Properties of optimization problems.** Graphs depicting the landscapes of different objective functions for a minimization problem with different properties: a) convex unimodal problem, the best case; b) multimodality; c) neutrality; d) ruggedness; e) deceptiveness; f) ill-conditioning; g) "needle-in-a-haystack"; and h) "nightmare". The graph corresponding to the ill-conditioned problem visualizes the contours of the objective function defined on a two-dimensional domain. The graphs are adapted, with modifications, from Weise et al. (2009).

Ill-conditioning. Conditioning of a problem can be defined as the range (over a level set[1]) of the maximum improvement of the objective function value in a ball of small radius centered on the given level set.

Conditioning of a function can be rigorously formalized in the case of convex quadratic functions: it is determined by the condition number of the Hessian matrix of the function, defined as the ratio between the largest and smallest eigenvalue of the Hessian. Since level sets associated to a convex quadratic function are ellipsoids, the condition number corresponds to the square root of the ratio between the largest and shortest axis lengths of the ellipsoid (see Figure 3.1.f)).

Ill-conditioned problems have a high condition number, typically larger than 10^5, that for real-world optimization problems can easily go up to 10^{10}.

High-dimensionality. The problem complexity does not scale linearly with the size of the problem. Thus, a search strategy (*e.g.*, exhaustive search) that is viable in search spaces of small dimensions might be inapplicable in moderate or large dimensional search spaces.

The example objective functions, as depicted in Figure 3.1, clearly show that the outlined properties can make the problem difficult even in a one-dimensional unconstrained search space. In real-world problems, these properties are rarely met individually. In fact, they usually appear combined leading to extremely hard problems, such as the "needle-in-a-haystack" problem (Figure 3.1.g)) and "nightmare" problem (Figure 3.1.h)).

3.3 Classification of optimization methods

As discussed in the previous sections, the existing optimization problems are of different size and have different properties and thus have different complexity. Therefore, different methods have been developed to solve them. According to the complexity of the optimization task, optimization methods can be generally classified as *exact* and *approximate* methods (Talbi, 2009).

Exact methods or complete methods are guaranteed to find an optimal solution for every finite size problem instance in bounded time. These methods are suitable for solving problems that belong to the so-called P class. Namely, according to computational complexity theory (Spiser, 2005) optimization problem (as decision problems) can be divided into two complexity classes, P and NP, with regard to the recourses, *i.e.*, memory and time, needed to solve them. Problems that can be solved by a deterministic machine in polynomial time belong the P complexity class, while the problems that can be solved by a nondeterministic machine in polynomial time belong to the NP complexity class.

Most of the real-world problems are large and complex. Furthermore, they are NP-hard and provably efficient algorithms to solve them do not exists. Assuming P $\neq$ NP, these problems need exponential computational time to be optimally solved.

Nevertheless, for many real-world optimization problems, it is not necessary to guarantee finding an optimal solution. Often it is sufficient to find a reasonably good (or approximate) solution using limited resources available. Thus, approximate methods have been introduced.

[1]Level set or contour of a given objective function $f : \mathcal{S} \to \mathbb{R}$ is set of all points in $f : \mathcal{S}$ for which f has some given constant value.

Approximate methods employ various strategies to find a high-quality solution in a reasonable time. They are designed to be practical for use. However, their practicality comes at the cost of no guarantee for global optimality of the provided solution.

The class of exact methods can be further divided into three main sub-classes: enumerative, relaxation and decomposition, and cutting plane/pricing methods.

- *Enumerative methods* include *branch and bound methods* (Lawler and Wood, 1966), *dynamic programming* (Bellman, 1957), A^* *search algorithms* (Dechter and Pearl, 1985), and other tree search algorithms, developed mainly in the operational research community or artificial intelligence community. The search is carried over the whole search space, and the problem is basically solved by subdividing the solution space into simpler subproblems, which are solved individually.

- *Relaxation and decomposition methods.* Relaxation methods are based on relaxation techniques (relax the strict requirements of the optimization problem, such as *Lagrangian relaxation* (Fisher, 1985)) that remove some of the problem constraints by incorporating them in the objective function. Decomposition methods, such as *Bender's decomposition* (Benders, 1962), fix the values of so-called complicated variables and solve the reduced problem iteratively.

- *Cutting plane and pricing methods* are based on polyhedral combinatorics and pruned search spaces. The use of *cutting planes* (Jeroslow, 1979) has greatly improved branch and bound algorithms, as for example in *branch and cut algorithms* (Caprara and Fishetti, 1997).

Furthermore, *constraint programming* (Apt, 2003) combines the concept of tree search with logical implications that is successfully used for tightly constrained combinatorial optimization problems.

In the class of approximate methods, two subclasses can be distinguished:

- *Approximation algorithms*, unlike heuristics, provide an approximate guarantee of the optimality of the obtained solutions and approximate run-time bounds (Apt, 2003). These algorithms give solid knowledge on the difficulty of the target problem, and have limited applicability since they are problem specific. Furthermore, in practice, it is difficult to obtain approximations close to the global optimal solutions, therefore the approximation algorithms are not very useful for many real-world applications.

- *Heuristics* arise from expert knowledge about the problem to be solved, and are not usually based on a formal analysis. They allow us to obtain acceptable performance at an acceptable cost over a wide range of problems. Originally, the heuristics developed in computer science were tailored to solve very specific problems and/or instances. These are today referred to as *problem-specific heuristics.* The research in the last 30-40 years resulted in more robust, general methods that may be used for solving various problems. These more general and improved heuristic methods are nowadays called *meta-heuristics* (Glover, 1986; Talbi, 2009).

Optimization methods can be classified according to different criteria, like randomness in the search process (deterministic/stochastic) or type of the optimization problem they are designed for (*e.g.*, constrained/unconstrained). Motivated by the problem of parameter estimation, that can be generalized as a nonconvex nonlinear optimization problem, we will give a brief overview of the methods for *nonlinear optimization* as opposed to

the methods for linear optimization. Methods for nonlinear optimization can be roughly classified as *local* and *global* optimization methods (Nocedal and Wright, 1999; Pintér, 2001).

Local optimization methods are traditional approaches to nonlinear continuous optimization. They converge fast to the optimum, provided that the search is started from a good initial point (in a close neighborhood of the optimum). However, they can only guarantee local convergence, as they do not have a mechanism to escape from a local optimum. These methods can be divided mainly into two subclasses, based on the use of the derivatives of the objective function, as follows:

- *Direct-search methods* are derivative-free methods that only exploit the information about the objective function in the sampled points (from the search space) to direct their further search (Kolda et al., 2003). They usually select a finite number of solutions in each step and check whether one of them is better then the current one. Direct-search methods are generally applicable and easy to use, but they become less efficient for high-dimensional problems. The two most representative members of this subclass are the *Hooke-Jeeves* algorithm (Hooke and Jeeves, 1961), a reliable but slow pattern search method, and the *Nelder-Mead* or *downhill simplex* algorithm (Nelder and Mead, 1965), a less reliable but more efficient method based on the idea of adaptive simplex.

- *Gradient-based or derivative-based methods* explicitly use the derivatives of the objective function being optimized with respect to the decision variables. The derivates used are the first-order derivative (or the *gradient*) and the second-order derivatives (or the *Hessian*) of the objective function. This means that the the structure of the function should be known, and moreover, the function should be smooth (continuous and differentiable). These methods can fail if the objective function is discontinuous (as well as for discontinuous derivatives of the objective function), nonsmooth, multimodal or ill-conditioned. The gradient-based methods include: i) general descent approaches, such as *steepest-descent* (uses the gradient information as it is based on the first-order Taylor approximation of the objective function, and has slow linear convergence), *Newton's method* (uses the gradient and Hessian information as it is based on the second-order Taylor approximation of the objective function, and has fast-quadratic convergence), and *Quasi-Newton* (uses an approximation of the Hessian matrix); ii) the cheap, fast, and popular Newton type methods for nonlinear least-squares estimation, *e.g.*, the *Gauss-Newton* method (neglects the second-order information in the Hessian matrix) and the *Levenberg-Marquard* method (a combination of Gauss-Newton and steepest-descent); and iii) methods for constrained problems, such as *sequential quadratic programming*. For proper references and a detailed overview, the reader is referred to Gill et al. (1981); Press et al. (1992); Nocedal and Wright (1999).

An intuitive way to deal with the problems of local optimality is the multi-start approach, that is, the local methods are restarted from different (random) starting points in the search space. However, this approach is ineffective, as there is no supervision in generating the starting points. In contrast to local optimization, global optimization is concerned with computation and characterization of global optima in nonlinear optimization problems. Global optimization problems are wide spread in the domain of mathematical modeling of real-world systems. Therefore, it is recommended to use global optimization approaches that are more robust than the local optimization methods regarding the dimensionality and characteristics of the objective function landscape.

Global optimization methods can be generally classified as *deterministic*, *stochastic* and *hybrid*.

- *Deterministic methods* can locate the global optima and assure their optimality, but there is no guarantee that they can solve any type of global optimization problem in finite time. Representative methods of this class were already discussed in the paragraph about exact methods and include branch and bound, interior-point, cutting planes, *etc.* For a detailed overview, the reader is referred to Horst et al. (1995); Horst and Tuy (1996).

- *Stochastic or probabilistic methods*, on the other hand, rely on probabilistic search rules to find good solutions (Törn et al., 1999). They can locate the neighborhood of the global optima relatively fast, but their efficiency comes at the cost of global optimality (which cannot be guaranteed) and computational effort. Nevertheless, these methods are easy to implement and can treat the objective function as a black box (*e.g.*, make no assumption on the properties or the structure of the function): the latter is a feature especially important for industrial applications, where the objective function is often simulated by a third-party software access to which is restricted. The simple *random search* methods are the oldest stochastic methods. Modifications of these include multi-start and adaptive versions. The most sophisticated stochastic methods are *stochastic meta-heuristics*. As already mentioned above, these general-purpose algorithms can find acceptable solutions in reasonable time in both complex and large search domains. Most meta-heuristics are inspired by natural processes such as evolution (*e.g.*, evolutionary algorithms (Fogel, 2000)), social behavior of biological organisms (*e.g.*, ant colony optimization (Dorigo and Stützle, 2004), particle swarm optimization (Kennedy and Eberhart, 1995)), and controlled cooling associated with physical processes (*e.g.*, simulated annealing (Kirkpatrick et al., 1983)).

- *Hybrid methods* are designed to obtain efficient and effective global optimization methods by combining different optimization approaches. They are inspired by the concept of *synergy*, that is, a mutually advantageous conjunction of distinct elements. As global optimization methods are known to be computationally expensive, especially for fine (intensive) search of the potential neighborhoods of the optimal solution, the majority of the existing hybrids are a combination of global optimization approaches with local solvers. The local solvers have a fast convergence if started in the vicinity of the optimal solution. One large class of such methods are *hybrid meta-heuristics* (Talbi, 2002), and particulary *memetic algorithms*, which are evolutionary algorithms refined through local search (Moscato, 1989).

Figure 3.2 depicts the above classification of optimization methods (Nelles, 2001; Pintér, 2001). As the parameter estimation problem is addressed by meta-heuristic optimization algorithms, the subclass of meta-heuristics will be discussed in more details in Chapter 4.

In the next section, we discuss a representative local optimization approach that we will use as a baseline.

3.4 Local optimization method: Algorithm 717

Our baseline method for optimization is a derivative-based local optimization method, Algorithm 717 (A717), essentially designed for nonlinear least-squares estimation. This

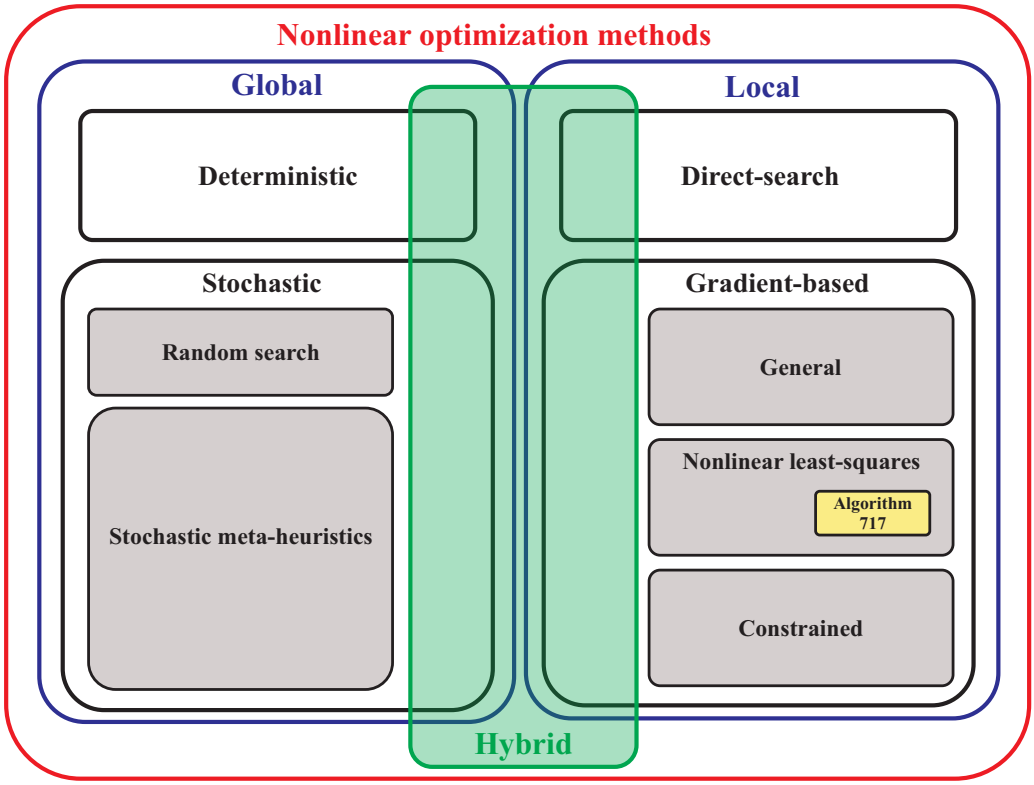


Figure 3.2: **Taxonomy of nonlinear optimization methods.** Boxes with different colors denote classification according to different criteria, while the yellow-filled box represents Algorithm 717, our baseline method for optimization.

particular method was also integrated in the automated modeling tool LAGRAMGE 2.0 (Todorovski and Džeroski, 2006) used in a previous study by Atanasova et al. (2006) to induce the model structure subjected to parameter estimation in this dissertation, as will be discussed in Chapter 7.

Algorithm 717 is a set of modules for solving the parameter estimation problem in nonlinear regression models like nonlinear least-squares, maximum likelihood and some robust fitting problems (Bunch et al., 1993). The basic method is a generalization of NL2SOL – an adaptive nonlinear least-squares algorithm (Dennis et al., 1981), which uses a model/trust-region technique for computing trial steps along with an adaptive choice for the Hessian model. In fact, NL2SOL is a variation of Newton's method (augmented Gauss-Newton method), in which a part of the Hessian is computed exactly and a part is approximated by a secant (Quasi-Newton) updating method. Thus, the algorithm sometimes reduces to the Gauss-Newton or the Levenberg-Marquardt method.

In order to promote convergence from poor starting guesses, the algorithm implements the idea of having a local quadratic model q_i of the objective function f at the current best solution c_i and an estimate of an ellipsoidal region centered at c_i in which q_i is trusted to represent f. So the next point, c_{i+1}, or the next trial step, is chosen to approximately minimize q_i on the ellipsoidal trust-region. The information obtained for f at c_{i+1} is used for model updating and also to resize and reshape the trust-region.

Among the modules, we can choose the ones for unconstrained optimization, or the ones that use simple bound constraints on the parameters. Furthermore, we can choose between modules that involve approximate computation of the needed derivatives by finite differences, and modules that expect the derivatives of the objective function to be provided by the routine that calls them.

In this work, we used the original implementation of A717 as available online[1]. Since A717 is not a global search method, we wrapped the original procedure in a loop of restarts with randomly chosen initial points, providing (in a way) a simple global search.

[1] `http://calgo.acm.org/717.gz`

4 Meta-heuristic optimization

Meta-heuristics are nowadays recognized as some of the most promising and successful optimization methods for solving hard and complex problems in science and engineering (Talbi, 2009). This family of approximate methods (according to the taxonomy represented in Chapter 3) essentially exploits the concept of combining basic heuristic strategies within a higher level framework aimed at efficiently and effectively exploring the search space of different optimization problems. While specific-problem-tailored heuristics tend to be fast, the solutions they find are not necessarily of high quality in general and may be difficult to expand to different problems. Meta-heuristics on the other hand are designed as general purpose methods that can provide satisfactory (near-optimal or globally optimal) solutions in reasonable time.

However, the robust design of meta-heuristics does not imply unconditional applicability. Meta-heuristics should be applied to problems for which satisfactory problem-specific exact algorithms or heuristics are not available, or when it is impractical to implement such a method. Related to the domain of continuous nonlinear optimization studied in this dissertation, meta-heuristics should be considered if: the derivative-based methods fail due to the ruggedness of the search landscape (*e.g.*, discontinuous, nonlinear, ill-conditioned, noisy, multi-modal, non-smooth, or nonseparable); the problem is of moderate or high dimensionality (considerably greater than three decision variables); there is no analytical formulation of the objective function (this situation is known as black-box optimization (Kargupta and Goldberg, 1996)); or the evaluation of the objective function is time-consuming. Indeed, the problem of parameter estimation within the framework of mathematical modeling of nonlinear dynamic systems, (*e.g.*, biological or ecological systems) considered in this dissertation, exhibits most of these properties.

The remainder of this chapter is organized as follows. Section 4.1 informally defines the meta-heuristic concept by summarizing the most important properties and elements of meta-heuristic design. Possible classifications of the existing meta-heuristic optimization approaches are outlined in Section 4.2. In addition, the basic characteristics of the two most successful population-based meta-heuristics paradigms, evolutionary and swarm intelligence algorithms, are presented. Section 4.3 introduces the representative methods used to perform parameter estimation in the considered nonlinear dynamic models of endocytosis and Lake Bled. Finally, Section 4.4 briefly outlines the problem of tuning meta-heuristics, *i.e.*, selecting the right values of the method's parameters that optimize its performance, and describes how it is approached in this study.

4.1 Definition

The term *meta-heuristics* was first proposed by Glover (1986). It is a compound of two words with Greek origin, that is *heuriskein*, which means "to find", and the suffix *meta*, which means "beyond, in an upper level". As introduced in Chapter 3, heuristics aim to provide reasonably good solutions to hard optimization problems in a short computational time. Meta-heuristics as heuristics share the same goal. However, unlike problem-specific

heuristics, meta-heuristics serve as a higher-level guiding strategy for designing heuristics to solve specific optimization problems.

While there exist many definitions of the meta-heuristics concept in the literature, *e.g.*, the overview paper by Blum and Roli (2003) quotes some of them, none of them has been widely accepted. In the absence of a commonly accepted definition, Blum and Roli (2003) analyzed the existing definitions in order to identify the fundamental properties of meta-heuristics. In this context, instead of a definition, we provide the nine fundamental properties of meta-heuristic algorithms, as summarized by Blum and Roli (2003):

- Meta-heuristics are strategies that "guide" the search process.
- Their goal is to efficiently explore the search space in order to find (near-) optimal solutions.
- Techniques which constitute meta-heuristic algorithms range from simple local search procedures to complex learning processes.
- Meta-heuristic algorithms are approximate and usually non-deterministic.
- They may incorporate mechanisms to avoid getting trapped in confined areas of the search space.
- The basic concepts of meta-heuristics permit an abstract level description.
- Meta-heuristics are not problem-specific.
- Meta-heuristics may make use of domain-specific knowledge in the form of heuristics that are controlled by the upper level strategy.
- Todays more advanced meta-heuristics use search experience (embodied in some form of memory) to guide the search.

As stated above, a meta-heuristic algorithm should be designed to efficiently and effectively explore the search space. This goal can be reached by integration of two essential concepts for guiding the search process, *intensification* and *diversification*, in the meta-heuristics design. The term *intensification* generally refers to exploitation of "promising" regions (neighborhood of high quality solutions found so far) in the search space, while *diversification* refers to exploration of search regions not visited before. While there exist a large number of meta-heuristic algorithms based on different philosophies and origins, the mechanisms they essentially employ to efficiently search the space of solutions are based on intensification and diversification. A "clever" combination of these two concepts in the search process has been acknowledged as the main force that drives a certain meta-heuristics design to a success. Related to this, Blum and Roli (2003) proposed a special framework, called *I&D frame*, for analysis of meta-heuristics as a first step towards systematic design of meta-heuristics. Within this framework, components of meta-heuristic algorithms can be characterized by the criteria they depend upon (objective function, non-objective oriented guiding information, such as search history, and randomization) and their effect on the search process. The analysis performed by the same authors, suggest that most of the basic meta-heuristic's components have both an intensification and a diversification effect. However, components mainly guided by the objective function have a dominant intensification effect, while components exploiting search history and randomness have a dominant diversification effect (Blum and Roli, 2003).

4.2 Classification

The value of research on meta-heuristics has been growing significantly over the last 20 years, as a result of their efficiency and effectiveness in solving large and complex problems across different domains. One can find a large number of successful meta-heuristic algorithms in the literature. Some of the best known and most widely applied meta-heuristics are simulated annealing (Kirkpatrick et al., 1983), tabu search (Glover, 1989, 1990), guided local search (Voudouris and Tsang, 2003), greedy randomized adaptive search procedure (Feo and Resende, 1995), iterated local search (Lourenço et al., 2003), evolutionary algorithms (Bäck et al., 1997), ant colony optimization (Dorigo and Stützle, 2004), particle swarm optimization (Engelbrecht, 2005), and scatter search (Glover et al., 2000).

A common characteristic of all meta-heuristics is that they try to avoid the generation of poor-quality solutions by using general mechanisms that extend problem-specific, single-run algorithms, such as greedy construction heuristics or local searches. What makes the existing meta-heuristics different from each other is the specific definition and integration of those mechanisms to avoid premature convergence to sub-optimal solutions. In this respect, several classifications of meta-heuristics are available in the literature, made according to different distinction criteria (Taillard et al., 2001; Blum and Roli, 2003; Dréo et al., 2007; Sirenko, 2009; Talbi, 2009). However, these classes are not mutually exclusive and many meta-heuristic algorithms combine ideas from different classes. Here, five relevant criteria are considered with a brief summary of the corresponding classifications. These are depicted in Figure 4.1 and described as follows.

Nature *vs.* non-nature inspired methods. There are many methods inspired by natural processes such as: biological evolution, *e.g.*, evolutionary algorithms; social behavior of biological organisms, *e.g.*, ant colony optimization and particle swarm optimization; and controlled cooling associated with physical processes, *e.g.*, simulated annealing. This classification is clearly motivated by the origin of the methods, but it is more intuitive than important, especially as there exist a number of meta-heuristics (*e.g.*, tabu search or the emerging class of hybrid meta-heuristics) that belong to both classes.

Constructive *vs.* iterative methods. Constructive or greedy meta-heuristics, such as ant colony optimization and greedy randomized adaptive search procedure, start to build a solution from scratch, assigning values to one decision variables at time, until a complete solution is constructed. Iterative meta-heuristics operate with complete solutions and transform them at each iteration using some search operators. Most of the meta-heuristics belong to the latter class.

Stochastic *vs.* deterministic methods. The relevant criterion for this classification is the nature of the decision making process which determines whether the decisions made within the search process, such as generation or selection of candidate solutions, involve random or deterministic rules. This furthermore determines the uniqueness of the obtained solution. More precisely, in deterministic meta-heuristics, such as basic local search and tabu search, the same initial solution will lead to the same final solution, whereas in stochastic meta-heuristics, such as evolutionary algorithms, different final solutions may be obtained from the same initial solution.

Memory use *vs.* memory-less methods. This classification is made with respect to the usage of the search history (memory) to direct the further search. Meta-heuristics like

tabu search, ant colony optimization or guided local search use information (*e.g.*, visited solutions, decisions or moves) extracted during the search to guide subsequent search. In contrast, methods like simulated annealing or the greedy randomized adaptive search procedure, are memory-less and make decisions based on the information associated with the current state of the search process. Today, the use of memory is acknowledged as a fundamental element in the meta-heuristic design.

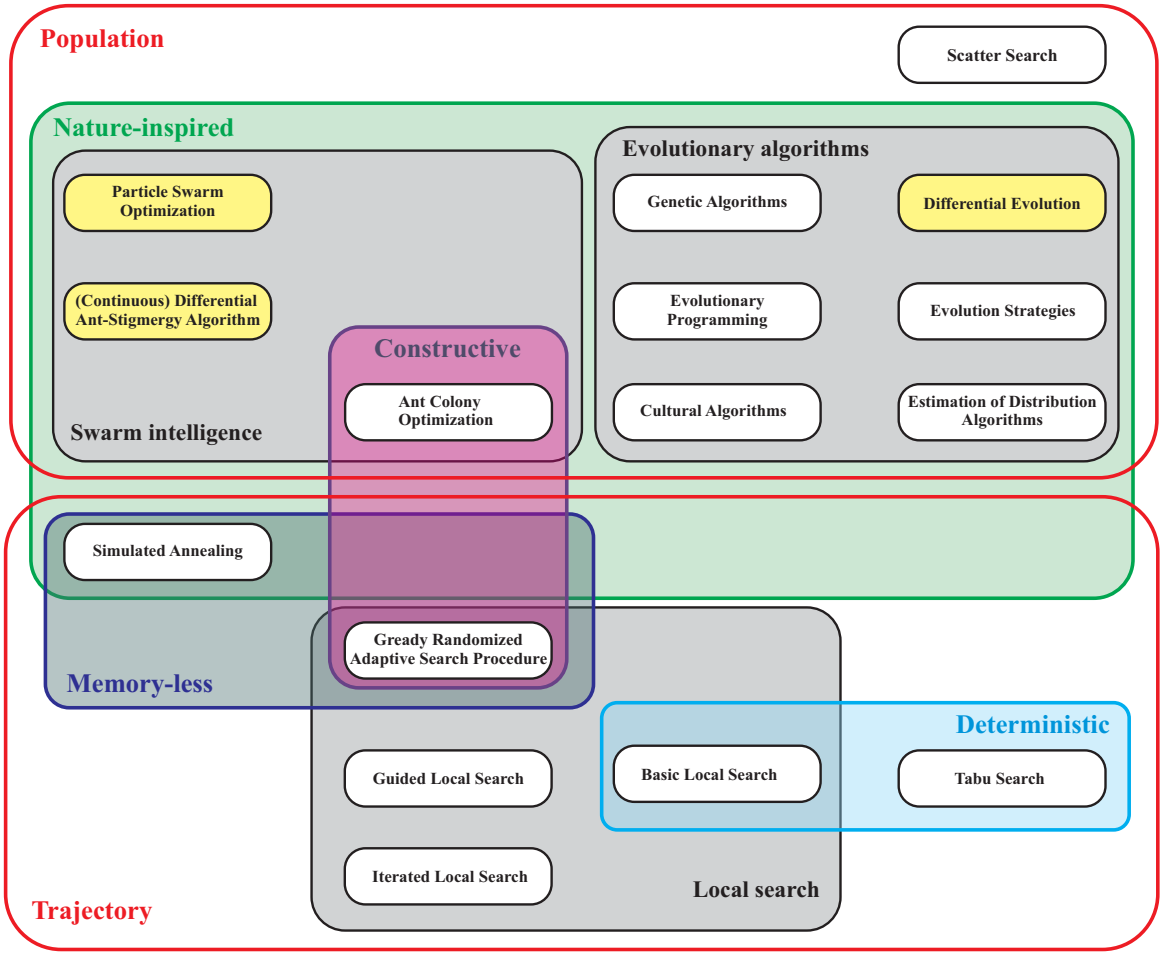


Figure 4.1: **Taxonomy of meta-heuristics.** Boxes with different colors denote classification according to different criteria. A single method is represented by a white-filled box with black edges, except for the ones depicted by yellow-filled boxes that represent the methods used for parameter estimation in our experimental analysis.

Single-solution *vs.* population-based methods. The criterion for this classification is the number of solutions that the method simultaneously manipulates and transforms at each search iteration. It describes the meta-heuristic algorithms more clearly than the other criteria: therefore, this is the primary classification considered for the introduction of the representative meta-heuristic methods.

Single-solution meta-heuristics use and manipulate a single solution at each iteration of the search, while the population-based meta-heuristics evolve a set of solutions during the search. The former are also knows as *trajectory methods*, because the search process is characterized by a trajectory in the search space, performed by iterative moving from the current solution to another one in the search space. In general, single-solution search methods are exploitation oriented, as they can intensify the search in local regions. In

contrast, population-based search methods are exploration oriented, as they allow better diversification across the whole search space.

A typical representative of the class of single-solution search methods is *local search* (LS) (also known as iterative improvement, hill climbing or descent search), the simplest and most probably the oldest meta-heuristics. The basic LS algorithm is an iterative procedure that replaces the current solution by a neighbor solution that improves the objective function. It requires an initial solution to be specified, therefore local search can be very sensitive to the choice of the initial solution and its main disadvantage is the convergence towards local optima. In order to improve LS, more sophisticated single-solution meta-heuristics have been developed (*e.g.*, simulated annealing, tabu search, iterated local search, greedy randomized adaptive search procedure, and guided local search), that make use of specific mechanisms to avoid being trapped in local optima. Some of these mechanisms involve restarting the search from different solutions as in iterated local search and the greedy randomized adaptive search procedure, changing the objective function as in guided local search, or accepting non-improving solutions (uphill moves) as in simulated annealing and tabu search.

Contemporary meta-heuristics mostly belong to the class of population-based search methods. An intuitive and logical explanation for this is that the population-based search framework provides a natural way for exploring the search space. However, it has to be used in combination with a proper population manipulation schema in order to perform successful search. Some representative population-based meta-heuristic paradigms are evolutionary algorithms, ant colony optimization, particle swarm optimization, and scatter search.

Below, we describe the principles behind the two most studied population-based meta-heuristics, *i.e.*, evolutionary algorithms and swarm intelligence algorithms.

4.2.1 Evolutionary algorithms

Algorithms inspired by the biological process of adaptation of species to their environment are known as *evolutionary algorithms* (Bäck et al., 1997; Eiben and Smith, 2003). They are basically developed upon the basis of the Darwinian theory of evolution (1859) that explains the adaptive change of the species by the principle of natural selection (also known as "survival of the fittest"), according to which those species that are best adapted to their environment will more likely survive and reproduce. The characteristics of the "fittest" species are inherited by the next generations and over time become dominant. Moreover, the theory states that random events may happen during reproduction that can change the characteristics of the existing species. Changes that result in better adaption of the species to the environmental conditions increase the chance for their survival.

Evolutionary algorithms (EAs) can be summarized as computational models of natural evolution. The basic EA model is a stochastic iterative procedure that maintains and evolves a population of individuals until some stopping condition is satisfied. A single individual in a population represents, directly or indirectly (through a decoding schema), one solution in the space of possible problem solutions. The "goodness" of a given solution (individual) is determined with the fitness value associated to it by the objective function that measures the quality of solutions. Starting from an initial (usually randomly initialized) population of individuals, the algorithm successively evolves the population towards better region of the search space by applying the randomized processes of *parent selection*, *reproduction*, and *replacement*. More precisely, at each iteration (also called generation), individuals are selected to be parents following the selection paradigm (individuals of higher quality are more likely to be selected). These parents reproduce using

the mechanisms of *recombination* and *mutation* to generate the offspring. The recombination mechanism, also called crossover, allows for mixing of the parental information while passing it to the children, while mutation serves as a generator of diversity in the population. The latter helps to avoid premature convergence of the search process that may happen as a result of high selection pressure. The replacement process, also called survival selection, happens at the end of each iteration, where according to a predefined rule it is decided which individuals, among the parents and offspring, will survive and form the new population.

A large number of different EAs have been developed within the *evolutionary computation* framework, where the differences depend on how population individuals are represented, the choice and particular implementation of the operators for population evolution (Bäck et al., 2000a,b). The field concerned with the development of problem-solving computational models of evolutionary processes is referred to as evolutionary computation. Historically, three main classes of algorithms have been developed within this field: *genetic algorithms*, *evolutionary strategies*, and *evolutionary programming*.

Genetic algorithms (GAs), probably the most widely known type of evolutionary algorithms, have been developed by Holland (1975) to understand the adaptive processes in natural systems. The classical GA model, originally developed for combinatorial optimization problems, includes a binary representation of individuals, proportional probabilistic parent selection, a crossover operator as the primary method for generating new individuals, a random bit-flipping mutation operator, and generational replacement (where parents are systematically replaced by the offspring). Modifications of the traditional GA include different representation schemes, as well as different operators for selection, crossover, and mutation. For details, the reader is referred to Bäck et al. (2000a,b).

Evolutionary strategies (ESs), introduced by Rechenberg (1973) and Schwefel (1977), use real-valued vector representations and are therefore perfectly suited for (and widely applied to) continuous optimization problems. ES uses completely deterministic selection operators, based on fitness ranking, and Gaussian perturbation as a mutation operator. Furthermore, they incorporate a self-adapted mutation step, meaning that the underlying algorithms are trying to optimize the evolution process itself, by optimizing the strategy parameters (that influence the search process) along with the decision variables.

Evolutionary programming (EP) was originally developed to evolve finite state machines and more generally learning machines (Fogel et al., 1966), where each state of the machine was represented by a bit string. Contemporary EP algorithms are mainly used for solving continuous optimization problems and include real-valued vector representations, deterministic parent selection, self-adapted mutation, no crossover and probabilistic replacement. It is very similar to the ES paradigm: it puts emphasis on mutation and includes self-adaptation of the strategy parameters.

Apart from the three main streams, other evolutionary paradigms, such as *differential evolution* (Storn and Price, 1997; Price et al., 2005), *estimation of distribution algorithms* (Larrañaga and Lozano, 2002), and *cultural algorithms* (Reynolds, 1994) have been recently developed. Differential evolution, which is described in Section 4.3 and is based on an arithmetic recombination, is one of the most successful approaches for continuous optimization. Estimation of distribution algorithms can be seen as descendants of GAs which use statistical information, extracted from the population, in the form of a probability

distribution from which the new individuals are sampled. Finally, cultural algorithms are computational models of cultural evolution based upon the principles of human social evolution.

4.2.2 Swarm intelligence algorithms

Algorithms inspired by the collective behavior of social organisms are known as swarm intelligence algorithms (Bonabeau et al., 1999; Engelbrecht, 2005). To be more precise, *swarm intelligence* refers to the complex problem-solving behavior of a group (*swarm*) of simple and unsophisticated agents emerging from the (direct or indirect) interaction with their local environment. Basically, the most important characteristic of a swarm system is the interaction (cooperation) of swarm members (agents): it shapes and dictates the swarm behavior, although there is no centralized control structure dictating how individual agents should behave.

The field concerned with developing algorithmic models of such behavior to solve complex problems, mainly optimization problems, is known as *computational swarm intelligence*. Studies of social animals (insects) have resulted in a number of computational models (algorithms) of swarm intelligence. Biological swarm systems that have inspired computational models include ant colonies, bee colonies, termite colonies, fish schools, and birds flocks. All of them are characterized by a collective behavior of the swarm that is far more complex than the individual behavior (ability) of a single swarm member and emerges as a result of the interaction of swarm members over time. Among the different ways of interaction in biological systems, social interaction is the most prominent one. It can happen directly (*e.g.*, by means of physical contact or visual/audio/chemical perception) or indirectly (by changes in the local environment): the indirect form of communication in swarm systems is generally known as *stigmergy* (Grassé, 1959). A well-known example of stigmergy is pheromonal communication observed in ant colonies (foraging behavior) and termite colonies (building nests).

The following paragraphs introduce the two most successful swarm intelligence optimization approaches, that is *ant colony optimization*, inspired by the foraging behavior of real ants, and *particle swarm optimization*, inspired by the flocking behavior of birds. For detailed description of ACO and PSO paradigm, including comprehensive overview and recent developments, the reader may consult the book by Engelbrecht (2005).

Ant colony optimization (ACO), introduced by Dorigo (1992), is a stochastic population-based optimization approach, initially developed for solving combinatorial optimization problems (Dorigo and Stützle, 2004). It uses a colony of artificial ants to construct the problem solution. The ants are guided by pheromone trails and heuristic information. Essentially, ACO mimics the principles of the simple pheromone trail-following behavior that real ants exhibit when searching for food. According to this behavior, ants mark the paths from their nest to the food sources by depositing a pheromone (chemical) trail on the ground (changes in the environment) which is, on the other hand, perceived by the other ants and used to guide their search: ants select with higher probability paths marked with stronger pheromone concentrations.

The basic ACO algorithm is an iterative constructive procedure that incrementally builds solutions in a probabilistic manner. It keeps adding solution components to a partial solution until a complete solution is constructed. This is performed by iterating mainly two processes, *solution construction* and *pheromone update*, until some stopping condition is satisfied.

The solution construction process manages a colony of ants that simultaneously and asynchronously perform randomized walks on a completely connected graph, called construction graph, whose vertices represent solution components. Moreover, a pheromone concentration is associated to every vertex, initially set to some value. Ants move by applying a stochastic local decision policy that makes use of the pheromone trails (which represent the acquired knowledge of the ant search process, an indirect form of memory of past performance) and heuristic information (which is problem-dependent, *e.g.*, may encode problem constraints).

By moving over the construction graph, the ants construct solutions. The constructed (partial or complete) solutions are evaluated and used to update the pheromone trails within the second process. The pheromone update process is especially important as it defines the balance between the exploration and exploitation of the search space. More precisely, it combines two rules: *evaporation*, *i.e.*, automatic decrease of pheromone concentrations in order to avoid premature convergence to sub-optimal search regions, and *reinforcement*, *i.e.*, increase of the pheromone concentrations on the paths associated with the generated solution in order to encourages the exploitation of the promising search areas.

In recent years, ACO algorithms have been extended to deal with continuous optimization problems (Dréo and Siarry, 2004; Korošec and Šilc, 2008; Socha and Dorigo, 2008). However, the direct application of ACO to continuous optimization problems is difficult, since the integration of the pheromone-laying model is not straightforward in this case. Therefore, most of these approaches do not follow the original ACO paradigm (Korošec and Šilc, 2008; Socha and Dorigo, 2008). The main idea of the proposed adaptations is to use continuous probabilistic distributions, *i.e.*, to use a continuous instead of a discrete pheromone model.

Particle swarm optimization (PSO), introduced by Kennedy and Eberhart (1995), is another stochastic population-based optimization approach developed initially for solving continuous optimization problems. It is based on a social-psychological model of social influence and social learning (Kennedy and Mendes, 2003). A typical PSO algorithm maintains a swarm of moving particles, where each particle is described by two properties: its *position* and *velocity* of moving in the search space. Each particle follows a very simple behavior: it moves in the search space according to its own experience and the social experience obtained by social interaction with the neighboring particles. As a result, the collective behavior of the particle swarm that emerges is that of all particles converging to the state that is optimal for all of them.

PSO evidently shares similarities with EA, inspired by the principles of biological evolution: both are inspired by natural phenomena and both maintain a population of candidate solutions and iteratively update (transform) the population using a variety of operators in order to find the optimal solution. However, PSO does not have selection, crossover or mutation operators: the main driving force of the swarm is the social interaction implicitly encoded in the social network structure. The social network structure is determined by the neighborhood of each particle, within which the particles can communicate by exchanging information about their success in the search space.

The basic PSO algorithm is described in the following section, along with the specific algorithm instance used to conduct the experimental analysis for this study.

4.3 Representative methods

This section describes the specific meta-heuristic optimization algorithms used to solve the nonlinear parameter estimation tasks considered in this dissertation. We address the task using a recently-developed swarm-based meta-heuristic differential ant-stigmergy algorithm (DASA), motivated by the fact that DASA has shown promising results in solving large scale continuous global optimization problems (Korošec et al., 2010, 2012), but has not been applied to the challenging task of parameter estimation in nonlinear ODE models. In addition, we use two well established meta-heuristics for global optimization, *i.e.*, particle swarm optimization and differential evolution. Below we provide a description of each of the three meta-heuristic algorithms. Note that we consider an additional algorithm for the task of parameter estimation in the Lake Bled ecosystem model, namely the continuous differential ant-stigmergy algorithm, which is conceptually similar to DASA.

4.3.1 The (continuous) differential ant-stigmergy algorithm

The differential ant-stigmergy algorithm (DASA) was initially proposed in 2006 by Korošec (2006) and further developed in (Korošec et al., 2012). It is an ACO-based meta-heuristic, designed to successfully cope with high-dimensional continuous optimization problems. The rationale behind the algorithm is in memorizing the "move" in the search space that improves the current best solution and using it in further search. The algorithm uses pheromones as a means of communication between ants (a case of stigmergy), combined with a graph representation of the search space. The DASA approach used in our experimental evaluation is described in detail by Korošec et al. (2012), where a reference to the available source code[1] is given. In principle, DASA relies on two distinctive characteristics, a differential graph and a continuous pheromone model. Therefore, these two are briefly discussed in the following. In addition, the main loop of the DASA search process is described. Finally, a version of DASA that does not use a differential graph is conceptually introduced.

First, DASA transforms the D-dimensional optimization problem into a graph-search problem. The differential graph used in DASA is a directed acyclic graph obtained by fine-grained discretization of the continuous parameters' differences (offsets). The graph has D layers with vertices, where each layer corresponds to a single parameter. Each vertex of the graph corresponds to a parameter offset value that defines a change from the current parameter value to the parameter value in the next search iteration. Furthermore, each vertex in a given layer is connected with all vertices in the next layer. The set of possible vertices (discretized offset values) for each parameter depends on the parameter's range, the discretization base b, and the maximal precision of the parameters ϵ, which defines the minimal possible offset value. Ants use these parameters' offsets to navigate trough the search space. At each search iteration, a single ant positioned at layer ℓ moves to a specific vertex in the next layer $\ell + 1$, according to the amount of pheromone deposited in the graph vertices belonging to the $(\ell + 1)$-th layer: the probability of a specific vertex to be chosen is proportional to the amount of pheromone deposited in the vertex.

Next, DASA performs pheromone-based search that involves best-solution-dependent pheromone distribution. The amount of pheromone is distributed over the graph vertices according to the probability density function (PDF) of Cauchy distribution (Press et al., 1992). DASA maintains a separate Cauchy PDF for each parameter. Initially, all Cauchy PDFs are identically defined by a location offset $l = 0$ and a scaling factor $s = 1$. As the search process progresses, the shape of the Cauchy PDFs changes: PDFs shrink as s

[1] `http://csd.ijs.si/korosec`

decreases and stretch as s increases, while the location offsets l move towards the offsets associated with the better solutions. The search strategy is guided by three user-defined real positive factors: the global scale increase factor s_+, the global scale decrease factor s_-, and the pheromone evaporation factor ρ. In general, these three factors define the balance between exploration and exploitation in the search space. They are used to calculate the values of the scaling factor, $s = f(s_+, s_-, \rho)$, and consequently influence the dispersion of the pheromone and the moves of the ants.

With two essential concepts of DASA defined, we now outline the search process performed by DASA. The main loop consists of iterative improvement of a temporary-best solution, performed by searching (constructing) appropriate offset paths in the differential graph. The search is carried out by a ants, all of which move simultaneously from a starting vertex to the ending vertex at the last level, resulting in a constructed paths. Based on the found paths, DASA generates and evaluates a new candidate solutions. The best among the a evaluated solutions is preserved as a current-best solution. If the current-best solution is better than the temporary-best solution, the later is replaced, while the pheromone amount is redistributed along the path corresponding to the temporary-best solution and the scale factor is accordingly modified to improve the convergence. If there is no improvement over the temporary-best solution, then the pheromone distributions stay centered along the path corresponding to the temporary-best solution, while their shape shrinks in order to enhance the exploitation of the search space. If for some predetermined number of tries (in this case D^2 for all ants) all the ants only find paths composed of zero-valued offsets, the search process is restarted by randomly selecting a new temporary-best solution and reinitializing the pheromone distributions. Related to this, DASA keeps information about a globally best solution, called global-best solution. This solution is the best over all restarted searches, while the temporary-best solution is the best solution found within one search (restart).

Finally, the continuous differential ant-stigmergy algorithm (CDASA) is an extended version of DASA based on the idea of continuous space exploration with probabilistic sampling (Korošec and Šilc, 2011). In contrast to DASA, CDASA uses arbitrary real offsets sampled from the inverse Cauchy probability function to navigate through the search space: the initial step of graph representation of the search space is omitted, consequently CDASA does not need the graph-related parameters (the discretization base and the maximal precision of the parameters). Like DASA, CDASA performs pheromone-based search that involves best-solution-dependent pheromone distribution. The amount of pheromone is distributed according to the Cauchy PDF as well: the only difference is that the pheromones are spread over a continuous domain of the parameter offsets. Regarding the search procedure, described in the previous paragraph, CDASA involves a small positive real user-defined threshold that determines the reset point: If for some predetermined number of tries (in this case D^2 for all ants) all the ants only find paths composed of offsets smaller than the specified threshold, the search process is restarted.

4.3.2 Particle swarm optimization

Particle swarm optimization, as introduced in the previous section, is a stochastic population-based optimization technique inspired by the concept of social behavior of biological organisms, *e.g.*, bird flocking or fish schooling (Reynolds, 1987). Essential for PSO is that the search process is performed by a swarm of particles, corresponding to a population of candidate solutions. Particles move (explore) in the search space, adjusting their position and velocity according to their own experience and to the social experience obtained by interaction with the neighboring particles. Here, we briefly describe the basic

computational model for PSO, and furthermore, outline the specifics of the particular PSO implementation used in our experimental evaluation.

The basic PSO method initializes the swarm with S uniformly-random positioned particles in the search space. The search for the optimal solution proceeds in iterations. In every iteration, the current position (at time t) of the particle $x(t)$ is incrementally updated with the new velocity $v(t+1)$ (*i.e.*, $x(t+1) = x(t) + v(t+1)$), which on the other hand is updated by using two sources of information. The first one, called cognitive component, reflects the experiential knowledge of the particle, which is its best position $x_{\mathrm{p}}(t)$ found so far. The second one, called social component, reflects the local knowledge of the search space obtained from the particle's neighborhood with size K and is represented by $x_{\mathrm{n}}(t)$, the best position found by the neighborhood of particles. The resulting formula for updating the velocity is then $v(t+1) = v(t) + c_1 r_1 (x_{\mathrm{p}}(t) - x(t)) + c_2 r_2 (x_{\mathrm{n}}(t) - x(t))$, where c_1 and c_2, called acceleration coefficients, are positive real values that balance the influence of the cognitive and the social component, while r_1 and r_2 are random factors uniformly sampled from the unit interval that introduce a stochastic component in the search.

The particular version of PSO used in our experimental evaluation is a standard variation of the basic PSO (the implementation is available online[1]), which includes only one acceleration coefficient c and an additional mechanism to control the exploration and exploitation in the search space via the parameter w, called inertia weight. The inertia weight basically controls the influence of the previous search direction on the new velocity. At each iteration, each particle chooses a few particles to be its informants, selects the best one from this set (neighborhood), and takes into account the information given by the chosen particle (the best informant). If there is no particle better than itself, then either the informant stays the same (default setting), or the informant is chosen randomly (optional setting). The velocity is updated according to the expression $v(t+1) = wv(t) + g(t) - x(t) + H(g(t), \|g(t) - x(t)\|)$, where the function H returns a random point inside the hypersphere with center of gravity $g(t)$ and radius $\|g(t) - x(t)\|$. The center of gravity is defined as $g(t) = \frac{1}{3}x(t) + \frac{1}{3}(x(t) + c(x_{\mathrm{p}}(t) - x(t))) + \frac{1}{3}(x(t) + c(x_{\mathrm{n}}(t) - x(t)))$.

4.3.3 Differential evolution

Differential evolution (DE) is a simple and efficient stochastic population-based metaheuristic for optimizing real-valued multi-modal functions, introduced by Storn and Price (1995, 1997). As it belongs to the class of EAs, it essentially maintains a population P of individuals (candidate solutions) that is iteratively evolved via the processes of selection, mutation, and crossover. However, the specific definition of these processes makes DE different from the typical EA schema. The latter is discussed in the following paragraphs, including the specific DE implementation used in our experimental evaluation.

The main difference between traditional EA and DE is in the reproduction step, where for every candidate individual x_{c} an offspring is created by using a mutated individual v. The latter is obtained by a simple arithmetic (differential) mutation operation over a set of parents (*e.g.*, x_1, x_2, x_3, such that $x_{\mathrm{c}} \neq x_1 \neq x_2 \neq x_3$) selected at random or by quality, based on one difference vector, *i.e.*, $v = x_1 + F \cdot (x_2 - x_3)$. The rate at which the population evolves can be controlled by a scale (mutation) factor F, a user-defined real number from the interval $[0, 2]$. To complement the differential mutation strategy, DE employs uniform crossover (also known as discrete recombination) over the candidate and mutated individual in order to generate the offspring. A user-specified crossover factor

[1] `http://www.particleswarm.info/standard_pso_2011_c.zip`

$CR \in [0, 1]$ is used to control the fraction of parameter values copied from the mutated individual to the offspring. Finally, the offspring is evaluated and if its fitness (objective function) is better, it replaces the corresponding candidate individual in the population.

Depending on the specific mutation and crossover procedure, one can chose among several DE strategies identified using the name format "DE/x/y/z". In the name format, x represents a string denoting the solution to be perturbed: i) a solution randomly chosen from the population (x ="rand"); ii) the current best solution (x = "best"); or iii) a solution based on the candidate solution combined with a difference vector towards the current best individual (x = "rand-to-best"), *i.e.*, $x_c + F \cdot (x_{best} - x_c)$. Further, y represents the number of difference vectors considered for perturbation, while z stands for the type of crossover being used that can be exponential (z = "exp") or binomial (z = "bin").

The implementation used in our experimental evaluation is based on the implementation of the DE algorithm described in the technical report by Storn and Price (1995), available online[1]. It includes 10 search strategies, STR, enumerated from one to 10 in the following order: 1 - "DE/best/1/exp", 2 - "DE/rand/1/exp", 3 - "DE/rand-to-best/1/exp", 4 - "DE/best/2/exp", 5 - "DE/rand/2/exp", 6 - "DE/best/1/bin", 7 - "DE/rand/1/bin", 8 - "DE/rand-to-best/1/bin", 9 - "DE/best/2/bin", and 10 - "DE/rand/2/bin". As the original code does not check whether the newly generated solutions are feasible, *i.e.*, lie within the prescribed parameter ranges, we slightly modified the code: if the new solution is outside the specified bounds, it is set to the closest range limit.

4.4 Tuning meta-heuristics

As evident from the material presented in the previous sections, meta-heuristic methods have many parameters that guide the search process. These "control" parameters determine the operational behavior of the methods and consequently influence the methods' performance. Choosing reliable and robust values for these parameters, *i.e.*, a default parameter setting, is a major challenge for the algorithm designers: in general, all these methods come with some recommend standard setting found by testing them on benchmark problems. These can always be considered as a good initial setup for the problem we want to optimize, but this does not mean that by default they will perform best for all kinds of problems.

To obtain the best possible performance on a given problem, one should consider task specific tuning of the parameters for the optimization method used (see, *e.g.*, the study by Dräger et al. (2009) in the domain of systems biology). Determining the optimal parameter setting is an optimization task in itself, which is extremely computationally expensive. There are two common approaches for choosing parameters values (Eiben and Smit, 2011): *parameter tuning* and *parameter control*. The first approach selects the parameter settings before running the optimization method (and they remain fixed while performing the optimization). The second approach optimizes the method's parameters along with the problem's parameters, i.e., decision variables of the original optimization problem[2].

A detailed discussion and survey of parameter tuning methods is given by Eiben and Smit (2011). According to this survey, one way to approach parameter tuning is by

[1] `http://http.icsi.berkeley.edu/~storn/DeWin.zip`

[2] Note the difference between "method's parameters", determining the behavior of the optimization method, and "model parameters", determining the behavior of the mathematical model. In the case of parameter estimation, "model parameters" are the decision variables we would like to optimally estimate. While, in the case of parameter (tuning) control, "method's parameters" extend the original set of decision variable and become subject to optimal estimation as well.

sampling methods. Sampling methods reduce the search effort by decreasing the number of investigated parameter settings as compared to the full factorial design: the basic full factorial design investigates 2^k parameter settings, subject to k parameters, each of which has 2 possible values; in the more general case, parameters can have an arbitrary number of values; moreover, an increase in the number of investigated parameters means an exponential increase in the number of parameter settings to be tested. Two widely used sampling methods are *Latin-squares* and *Taguchi orthogonal arrays* (appropriate references are given by Eiben and Smit (2011)). However, these are not the most robust sampling techniques, *e.g.*, Latin-squares or Latin hypercube sampling is good in the case where one of the parameters dominates the method's performance, while it should be used with care if there are interactions among the parameters.

Ultimately, we would like to find a sampling schema that will be able to detect the interactions among the parameters, will be independent from user-specified information regarding the particular parameter values to be considered (typical for factorial design), and will deliver a small but representative sample of the parameter search space. The first two requirement are satisfied by pure random sampling, but the last is not, as random sampling does not guarantee that the sampled values are evenly spread across the entire domain. The so-called *low-discrepancy sequences* were specifically designed to fulfill all three requirements. Therefore, we consider a representative variation of low-discrepancy sequences, called *Sobol' sequences*, for sampling the parameter space of the considered meta-heuristics in the case of parameter identification in the ODE model of endosome maturation.

Sobol' sequences were introduced by Sobol' in 1967 (Press et al., 1992). These low-discrepancy sequences of sampling points have very desirable uniformity properties. Sobol sequences, sampled from the D-dimensional unit search space, are quasi-random sequences of D-tuples that are more uniformly distributed than uncorrelated random sequences of D-tuples. These sequences are neither random nor pseudo-random, as they are cleverly generated not to be serially uncorrelated, but instead to take into account which tuples in the search space have already been sampled, as depicted in Figure 4.2. For a detailed explanation and overview of the schemas for generating Sobol' sequences, we refer to Press et al. (1992). The particular implementation of Sobol' sampling used in this paper is based on the Gray code order (Joe and Kuo, 2008) and is available online[1].

The results gathered by the parameter tuning process are most often subjected to ordinal data analysis, which includes ranking of the different sampled parameter sets according to some calculated statistics, *e.g.*, best, average, or median performance of the method in some predefined number of method executions (Talbi, 2009). The performance of the method is usually expressed in terms of the quality of the solution, that is, objective function value, but could be also measured in terms of convergence speed, number of successful executions, or some other relevant criteria. In general, the final rank can be calculated by aggregating the individual rankings assigned to the different parameter sets according to the considered criteria. With respect to our parameter tuning task, we performed ordinal data analysis subject to a single ranking criterion, *i.e.*, the median performance of the method. Details and results are provided in Chapter 6.

[1] `http://web.maths.unsw.edu.au/~fkuo/sobol/index.html`

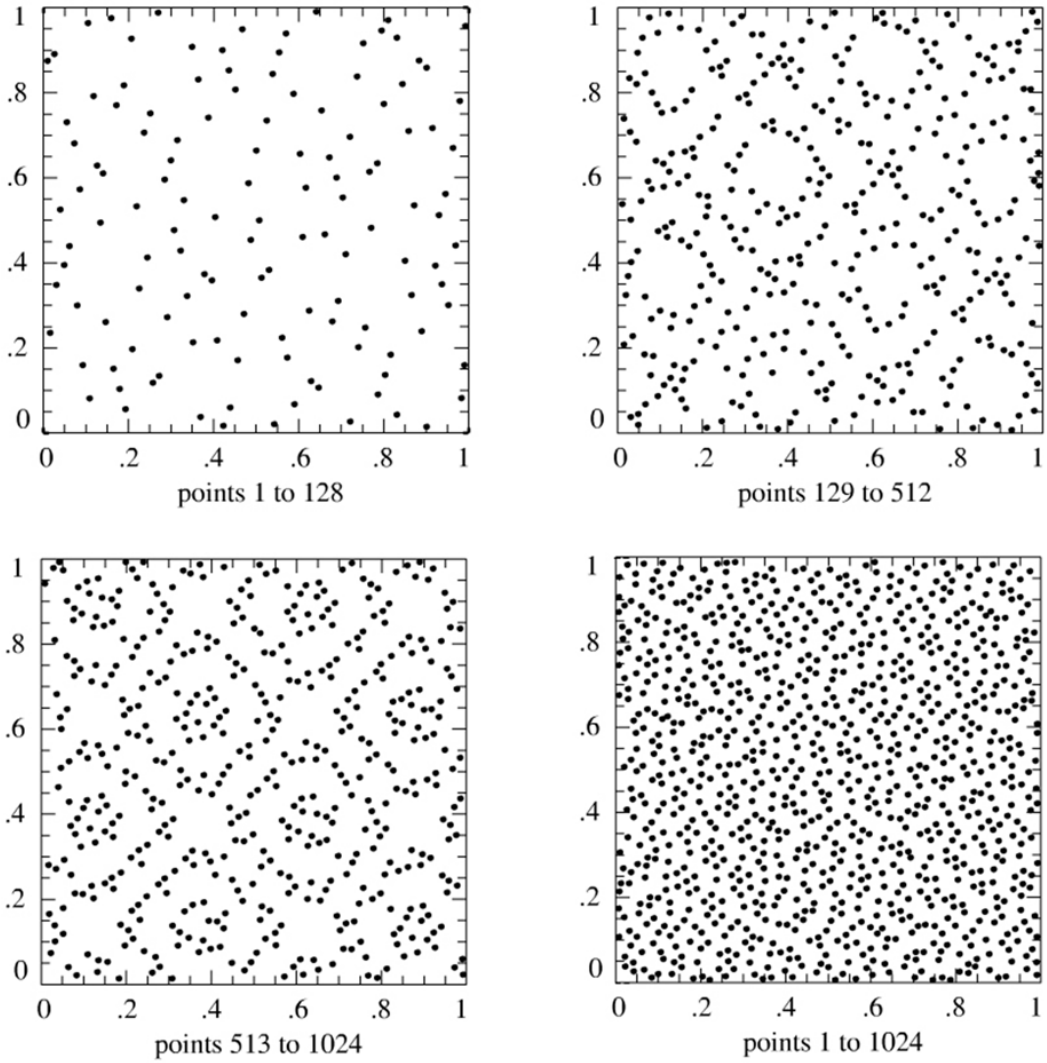


Figure 4.2: **Scatter plot of the first 1024 points of a two-dimensional Sobol' sequence.** The points are evenly spread over the two-dimensional space. Rather than random, the successive points at each iteration are generated to fill the gaps in the previously generated distribution. The above graphs have been adapted from Press et al. (1992).

5 Experimental methodology

The contribution part of this dissertation includes two studies of parameter estimation in two specific models from two different domains: the first model captures the dynamics of a representative cell process in systems biology, *i.e.*, endosome maturation in endocytosis, while the second captures the population dynamics of phytoplankton and zooplankton in a representative aquatic ecosystem, *i.e.*, Lake Bled. Both are treated in similar way, and the common experimental methodology is therefore outlined in this chapter. However, details, such as the setup of the optimization methods, that are specific for each estimation task are covered in the chapters devoted to the particular task. If not specified otherwise, the methodology described here is valid in the following.

The remainder of this chapter is organized in two sections. Section 5.1 outlines the details of the Monte Carlo-based approach for practical parameter identifiability, while Section 5.2 provides details on the methodology for performance comparison of the chosen optimization methods on the considered parameter estimation tasks.

5.1 Practical parameter identifiability

For the specific tasks of modeling endocytosis and Lake Bled, we generated 1000 and 400 datasets, respectively, by simulating the models with their reference parameter values and adding Gaussian noise. Given a percentage of noise, relative noise level s, we calculated the noisy value as $Y_{\text{noisy}} = Y \cdot (1 + s \cdot N(0,1))$, considering a noise level of 20%. We estimated the parameters of the model using these datasets, and collected the estimates for outlier examination and further statistical analysis. We followed the same procedure as described by Joshi et al. (2006), according to which outliers are data points that do not belong to the interval $[Q1 - 1.5(Q3 - Q1),\ Q3 + 1.5(Q3 - Q1)]$, where $Q1$ and $Q3$ represent the 25^{th} and 75^{th} percentiles of the sample, respectively. The detected outliers are removed: More precisely, the new (reduced) sample includes only the estimates obtained from those datasets that did not produce any outlier over all parameters. The distributions of the parameters are presented with histograms, including the corresponding 95% confidence intervals (CIs): these are calculated as the length of the interval between the 2.5^{th} and 97.5^{th} percentile of the sample. The width of the bins h for a single histogram is calculated according to the Freedman-Diaconis rule (Freedman and Diaconis, 1981) as

$$h = \frac{2\text{IQR}}{\sqrt[3]{N_{\text{s}}}}, \tag{5.1}$$

where IQR is the interquartile range of the sample and N_{s} is the sample size.

Based on the outlier-free samples of parameter estimates, the correlation of two model parameters c_i and c_j in terms of a linear dependence is calculated based on the Pearson correlation coefficient (Rice, 2007) according to the formula

$$R(c_i, c_j) \approx R(\tilde{c}_i, \tilde{c}_j) = \frac{\sum_{k=1}^{N_{\text{s}}} \tilde{c}_{ik}\tilde{c}_{jk} - N_{\text{s}}\mu_{\tilde{c}_i}\mu_{\tilde{c}_j}}{N_{\text{s}}\sigma_{\tilde{c}_i}\sigma_{\tilde{c}_j}}, \tag{5.2}$$

where $\mu_{\tilde{c}_i}$ and $\sigma_{\tilde{c}_i}$ are the mean and the standard deviation of the vector $\tilde{c}_i$ with estimates of the parameter c_i, while $\mu_{\tilde{c}_j}$ and $\sigma_{\tilde{c}_j}$ are the mean and the standard deviation of the vector $\tilde{c}_j$ with estimates of the parameter c_j. A correlation of 1 (or -1) means a perfect positive (or negative) linear relationship between the two parameters.

5.2 Comparison methodology

To guarantee a fair comparison of the selected optimization methods, we ran each method 25 times allowing half a million of evaluations of the objective function per single run. We used a number of performance evaluation metrics to compare the utility of the optimization methods for parameter estimation; the reported method performance is the average/median performance over all 25 runs. While the first quality measure is about the convergence rate of the optimization methods, the others focus on the quality of the obtained models.

Convergence curves are commonly used for visualizing the convergence rates of optimization methods. They show the change of the value of the objective function with the increasing number of objective function evaluations. Each curve in our paper depicts the change of the objective function value averaged over 25 runs: The convergence curves are plotted on log-log plots, with a logarithmic scale for both axes in order to be able to capture the convergence trend over a wide range of values.

Root mean squared error (RMSE) measures the difference between output values predicted by the model $\hat{Y}$ and the observed values of the output variables Y.

$$\mathrm{RMSE} = \mathrm{RMSE}(Y, \hat{Y}) = \sqrt{\frac{1}{N}\mathrm{SSE}(Y, \hat{Y})} \tag{5.3}$$

The division by the number of data points and the square root in the definition of RMSE make its measurement units and scale comparable to the ones of the observed output variables. This is in contrast with the SSE measure defined with Eq. (2.5). The specific RMSE formulation in the two applications, will depend on the specific version of the SSE metric used. For parameter estimation in the specific endocytosis model it is defined by Eq. (2.7), while for the specific Lake Bled model by Eq. (2.6). Finally, note that better models have smaller values of RMSE.

As defined above, the RMSE quality metric measures the degree-of-fit between simulated model output and observed system output. However, reconstruction of system dynamics goes beyond reconstructing output; ultimately, modeling is about capturing the complete (also unobserved) system dynamics. To measure this aspect of reconstruction quality, we have to measure the degree-of-fit between simulated and observed values of the system variables. Although this is impossible in real cases where system variables cannot be directly observed, experiments with artificial data allow us to measure this aspect of model quality. In this context, we use an additional model quality metric when comparing the methods in the case of parameter estimation in the endocytosis model from artificial data.

The *root mean squared error of the completely simulated model* (RMSEm) is defined as

$$\mathrm{RMSEm} = \sqrt{\frac{1}{N}\sum_{i=1}^{4}\sum_{j=1}^{N}\left(S_i[j] - \hat{S}_i[j]\right)^2}, \tag{5.4}$$

where $S_i[j]$ and $\hat{S}_i[j]$ are the values of the system variables from the reference model and the estimated (predicted) model, respectively. This metric allow us to test whether the

good quality of the model in terms of outputs is related to the ability of the model to capture the unobserved system dynamics.

To represent the distributions of the quality measure values over the 25 runs of a specific optimization method regarding a specific experimental scenario, we used *box-and-whisker diagrams (boxplots)*. They not only provide a convenient graphical representation of the dispersion, skewness, and outliers in a given data sample, but also enable a visual comparison of different data samples. The top and bottom edge of the box in a boxplot represent the 25^{th} and 75^{th} percentiles of the sample, respectively; in consequence, the box height corresponds to the *interquartile range* (IQR). The line in the middle of the box corresponds to the sample median. The sample mean is represented with a diamond. The "whiskers", *i.e.*, the two lines extending above and below the box, represent the sample range. The maximal length of the whiskers is set to $1.5 \cdot \text{IQR}$. Data points above and below the whiskers' end points correspond to the sample outliers and are represented with "+" markers. Finally, the notches, *i.e.*, triangular markers, of the boxes represent the variability of the median in the sample. The width of a notch is computed so that boxplots whose notches do not overlap have medians that are significantly different at the 0.05 significance level, assuming a normal data distribution. The boxplots presented in this dissertation were generated using the MATLAB statistical toolbox[1].

Statistical significance testing was performed in order to assess the obtained differences in performance between the compared optimization methods. We used the post-hoc multiple comparison Holm test (Holm, 1979), according to which we first rank the compared methods based on their performances averaged over all test problems and assign a score r_i for $i = 0, \ldots, N_{\text{m}} - 1$, where N_{m} is the number of methods being compared and i is the appropriate rank ($i = 0$ corresponds to the best ranked method, while $i = N_{\text{m}} - 1$ to the worst ranked method). Second, we select the method with the best score (lowest rank), r_0 and calculate the values

$$z_i = \frac{r_0 - r_i}{\sqrt{\frac{N_{\text{m}}(N_{\text{m}}+1)}{(6N_{\text{tp}})}}}, \tag{5.5}$$

where N_{tp} is the number of considered test problems for each method. Finally, we calculate the cumulative normal distribution values p_i corresponding to z_i and compare them with the corresponding α/i values where α is the significance level, set to 0.05 in our case. We report the z_i, p_i, and α/i values in a table, where each row corresponds to one of the methods. The best ranking method, used as reference method, is excluded from the table. Based on these values, we can make a decision about the null-hypothesis that "there is no difference in performance between this and the best ranking method".

[1] `http://www.mathworks.com/help/toolbox/stats/boxplot.html`

6 Modeling the dynamics of endocytosis

In this chapter, we address the task of estimating the parameters of a nonlinear ODE model of endocytosis, more specifically of the maturation of endosomes, which are membrane-bound intracellular compartments used to transport and disintegrate external cargo. The model focuses on a key endocytotic regulatory system that switches from cargo transport in early endosomes to cargo disintegration in mature endosomes (Zerial and McBride, 2001; Rink et al., 2005). The regulatory system is based on the process of conversion of Rab5 domain proteins to Rab7 domain proteins.

Using both a theoretical and an experimental approach to model this process, Del Conte-Zerial et al. (2008) show that a cut-out switch ODE model provides the best fit to the biological observations, which show a rapid transition from the state with high Rab5 and low Rab7 concentrations in early endosomes to the inverse state of low Rab5 and high Rab7 concentrations in mature endosomes. The modeling task at hand is representative and challenging due to the characteristics of the modeled dynamics and the measurements process. Most notably, we have to cope with the limited measurability of the concentrations of Rab5 and Rab7 domain proteins in the endosome, since the different, *i.e.*, active and passive, states of these proteins can not be distinguished in the measurement process.

More precisely, we study the effect of this kind of limited observability of the system dynamics on the complexity of the parameter estimation task, as well as the applicability and performance of four different optimization methods in this context. In order to do so, we define four different observation scenarios and generate artificial (pseudo-experimental) data for each of them. The scenarios cover a wide range of situations, from the simplest one of complete observability, where the concentrations of all protein states are assumed to be directly measurable, to the most complex (and realistic) scenario, where the observations give the total concentrations of each protein in all different states.

We test the performance of several optimization methods in the different observation scenarios and compare the ability of the different methods to cope with them. The compared methods are the differential ant-stigmergy algorithm (DASA), particle swarm optimization (PSO), differential evolution (DE) and Algorithm 717 (A717). A final set of experiments, based on real-experimental data, are performed in order to check the validity of the results obtained on artificial data. More specifically, we test the methods' performance on measured data obtained through real-world biological experiments that corresponds to the most complex observation scenario described above.

The remainder of this chapter is organized as follows. Section 6.1 describes and mathematically formulates the specific endocytotic process being modeled, i.e., Rab5-to-Rab7 conversion. Section 6.2 formulates the problem of parameter estimation, specifying the objective function, parameter ranges, the data used, and the considered observation scenarios. Section 6.3 outlines the specific experimental design related to the particular setup of the optimization methods. Section 6.4 presents and discusses the experimental results of parameter estimation in the Rab5-to-Rab7 conversion model. Finally, Section 6.5 summarizes the results and outlines the conclusions of the experimental study. The material presented in this chapter has already been published in the form of conference and journal papers (Tashkova et al., 2009, 2010a,b, 2011a).

6.1 Model

This work addresses the task of parameter estimation in a practically relevant model of endocytosis, *i.e.*, describing one part of the life-cycle of endosomes. Endosomes are membrane-bound intracellular components that typically encapsulate, transport, and disintegrate external cargo within cells. The model at hand focuses on the process of endosome maturation, representing it by a cut-out switch between the concentrations of Rab5 and Rab7 domain proteins (Zerial and McBride, 2001; Rink et al., 2005). The theoretical and experimental approach (Del Conte-Zerial et al., 2008) undertaken to model the endocytosis rely on the mutual exclusiveness of the Rab5 and Rab7 domains. Early endosomes have with high Rab5 and low Rab7 concentrations, while late or mature endosomes have low Rab5 and high Rab7 concentrations. The transition from the early to the mature state is rapid.

To model the Rab5-to-Rab7 conversion, we distinguish between active and inactive (passive) states of the Rab5 and Rab7 domain proteins. Thus, the ODE model involves four system (endogenous) variables corresponding to the concentrations of Rab5 domain proteins in inactive (r_5) and active state (R_5) and Rab7 domain proteins in inactive (r_7) and active state (R_7), measured in mol/l. These four species (chemical compounds) are involved in ten different biochemical reactions $v_1, \ldots, v_{10}$ parameterized with eighteen constant parameters $c_1, \ldots, c_{18}$ corresponding to the kinetic rates of the reactions, leading to the following structure of the model ODEs:

$$\begin{aligned}
\frac{d}{dt} r_5 &= v_1 + v_7 + v_9 - v_2 - v_3, \\
\frac{d}{dt} R_5 &= v_2 - v_7 - v_9, \\
\frac{d}{dt} r_7 &= v_4 + v_{10} - v_5 - v_6 - v_8, \\
\frac{d}{dt} R_7 &= v_5 + v_6 - v_{10}.
\end{aligned} \tag{6.1}$$

Here, $v_1, \ldots, v_{10}$ denote the kinetic models of the corresponding biochemical reactions, given below:

$$\begin{aligned}
v_1 &= c_1, \\
v_2 &= \frac{c_2 r_5}{1 + e^{(c_3 - R_5)c_4}} \frac{t}{100 + t}, \\
v_3 &= c_5 r_5, \\
v_4 &= c_6, \\
v_5 &= \frac{c_7 r_7 R_7^{c_8}}{c_9 + R_7^{c_8}}, \\
v_6 &= \frac{c_{10} r_7}{1 + e^{(c_{11} - R_5)c_{12}}}, \\
v_7 &= \frac{c_{13} R_5}{1 + e^{(c_{14} - R_7)c_{15}}}, \\
v_8 &= c_{16} r_7, \\
v_9 &= c_{17} R_5, \\
v_{10} &= c_{18} R_7.
\end{aligned} \tag{6.2}$$

Note that all variables in the model are system variables, *i.e.*, $S = \{r_5, R_5, r_7, R_7\}$ and there are no exogenous variables, *i.e.*, $E = \emptyset$. Figure 6.1 depicts the simulated dynamics

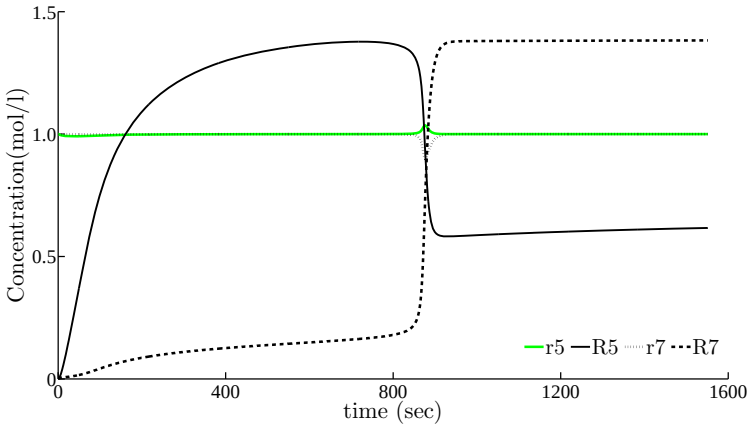


Figure 6.1: **Simulated dynamics of the Rab5-to-Rab7 conversion model.** Simulation of the cut-out switch model, capturing the conversion of Rab5 domain proteins to Rab7 domain proteins in the regulatory system of endocytosis, as proposed by Del Conte-Zerial et al. (2008).

of the four system variables of the model in the time interval $[0, 1551]$ seconds, with the parameters values set to

$$\begin{array}{lll} c_1 = 1, & c_2 = 0.3, & c_3 = 0.1, \\ c_4 = 2.5, & c_5 = 1, & c_6 = 0.483, \\ c_7 = 0.21, & c_8 = 3, & c_9 = 0.1, \\ c_{10} = 0.021, & c_{11} = 1, & c_{12} = 3, \\ c_{13} = 0.31, & c_{14} = 0.3, & c_{15} = 3, \\ c_{16} = 0.483, & c_{17} = 0.06, & c_{18} = 0.15, \end{array} \tag{6.3}$$

and initial values of the state variables, at the time point $t_0 = 0$ seconds, set to

$$\begin{aligned} r_5(t_0) &= r_5(0) = 1 \text{ mol/l}, \\ R_5(t_0) &= R_5(0) = 0.001 \text{ mol/l}, \\ r_7(t_0) &= r_7(0) = 1 \text{ mol/l}, \\ R_7(t_0) &= R_5(0) = 0.001 \text{ mol/l}, \end{aligned} \tag{6.4}$$

as proposed by Del Conte-Zerial et al. (2008). Note that the behavior of the concentrations of the active-state proteins R_5 and R_7 follow the expected (rapid) cut-out switch from high Rab5 and low Rab7 to low Rab5 and high Rab7 concentrations, while the concentrations of the passive-state proteins r_5 and r_7 remain almost constant throughout the whole process, with a small but notable change at the transition point.

6.2 Problem statement

In sum, the task of parameter estimation in the Rab5-to-Rab7 cut-out switch model leads to a 22-dimensional continuous minimization problem with 18 dimensions corresponding to model parameters and four dimensions corresponding to the initial values of the four system variables. The objective function, *sum of squared errors* (SSE, as defined in Eq. (2.7)) between the observed and predicted values of the system output, is minimized with respect to the given data, subject to the structure of the ODEs of the Rab5-to-Rab7 conversion model (described by Eqs. (6.1) and (6.2)) and the following bound constraints on the values of the constant parameters and protein concentrations: $c_i \in (0, 4]$ for

$1 \leq i \leq 18$, $c_i \in (0, 2]$ for $19 \leq i \leq 22$, $r_5(t) \geq 0$, $R_5(t) \geq 0$, $r_7(t) \geq 0$, and $R_7(t) \geq 0$. To calculate the objective function, we perform ODE integration. Despite the fact that we use advanced adaptive-step integrators, for some parameter sets the ODE integration can fail, due to the discontinuities in the model dynamics: in that case or in the case of violation of the basic constraint about non-negative values for the simulated protein concentration, we simply discard the respective solution (by giving the objective function a very high real value).

In order to evaluate the performance of different parameter optimization methods on this task, we conducted experiments with artificial data, obtained by simulating the Rab5-to-Rab7 conversion model, and with real data from experimental measurements.

6.2.1 Data

Artificial (pseudo-experimental) data. We generated the artificial data by simulating the ODE model from Eqs. (6.1–6.4) at 2781 equally spaced time points inside the interval $[0, 1551]$ seconds. To obtain more realistic artificial data, we added a normal Gaussian noise $N(0, 1)$ to the noise-free simulated data. Given the percentage of a relative noise level s, we calculate the noisy data as $Y_{\text{noisy}}(t) = Y(t) \cdot (1 + s \cdot N(0, 1))$. In our experiments, we use two noise levels of 5% and 20%. Note that the noise-free data correspond to the noise level of 0%.

Measured (real-experimental) data. In the second set of experiments, we used the real time-course measurements from Del Conte-Zerial et al. (2008). The measurements are taken following a complex procedure, where a number of endosomes were followed in three independent experiments for Rab5 (23 endosomes) and one for Rab7 (15 endosomes). Experimental data for different endosomes were manually aligned around the conversion point, scaled, and averaged at 10571 time points in the interval of $[-5, 330]$ seconds, where time point 0 corresponds to the conversion point. Note however, that due to physical limitation of the measurement experiments, only the total concentrations of the active and inactive Rab5 and Rab7 domain proteins could be measured, leading to a complex observation scenario with the following two output variables $Y_1(t) = r_5(t) + R_5(t)$ and $Y_2(t) = r_7(t) + R_7(t)$.

6.2.2 Observation scenarios

The limited measurability of the system variables in the real-world measurement scenario, described above, represents one of the most challenging properties of the parameter estimation task addressed in this study. To evaluate the impact that the limited observability has on the difficulty of the optimization task (and consequently on the performance of different optimization methods), we define here four observation scenarios, ranging from the simplest one that assumes that all the system variables can be directly measured to the most complex one that corresponds to the limitations of the real measurement process described in the previous paragraph.

Complete observation (CO). In this scenario, we assume that all the system variables are directly observed, meaning that the measurement process can identify the four concentrations of active and inactive states of the Rab5 and Rab7 proteins at each time point, *i.e.*, $Y_1(t) = r_5(t)$, $Y_2(t) = R_5(t)$, $Y_3(t) = r_7(t)$, and $Y_4(t) = R_7(t)$.

Active-state protein concentration observation (AO). Here, we assume that only concentrations of the active-state proteins can be observed, *i.e.*, $Y_1(t) = R_5(t)$ and $Y_2(t) = R_7(t)$. This scenario is simpler than the real one. Since we can measure total (active-state and passive-state) protein concentration and the passive-state protein concentrations are expected to be constant most of the time (see Figure 6.1), this scenario is based on a reasonable assumption.

Total protein concentration observation (TO). This scenario represents the real measurement process outlined above, where $Y_1(t) = r_5(t)+R_5(t)$ and $Y_2(t) = r_7(t)+R_7(t)$.

Neglecting passive-state protein concentration observation (NPO). This is the scenario based on how the measurements are (visually) matched against model simulations by Del Conte-Zerial et al. (2008). In this case, we observe the total protein concentrations, *i.e.*, $Y_1(t) = r_5(t) + R_5(t)$ and $Y_2(t) = r_7(t) + R_7(t)$, but we match them against concentrations of the active-state proteins predicted by the model, *i.e.*, $\hat{Y}_1(t) = R_5(t)$ and $\hat{Y}_2(t) = R_7(t)$. The rationale for this scenario is the same as for the second one (AO) and it is included here to match the procedure used by Del Conte-Zerial et al. (2008).

6.3 Experimental setup

Subject to four optimization methods, four observation scenarios, and four datasets, that is, three artificial and one real, we performed 64 experiments. In addition to the experimental methodology described in Chapter 5, in this section we outline the specific setup of the optimization methods used to perform the experiments for parameter estimation of the ODE model represented by Eqs. (6.1–6.4). The parameter settings for the meta-heuristic algorithms used in the evaluation task at hand were chosen based on the Sobol'-sampling-based parameter tuning, as follows.

The standard DE algorithm has only four parameters, while DASA and PSO have more: consequently, we chose only four parameters per single method to be tuned. For DASA, we chose the three real-valued parameters that directly influence the search heuristic (s_+, s_-, and ρ) and the number of ants (a), while for PSO, we chose the size of the swarm (S), the size of the neighborhood (K), the inertia weight (w), and the acceleration coefficient (c). The number of sampled parameter settings (4-tuples) per method was 2000. Due to the stochastic nature of the methods, every parameter setting was used for optimization of the endocytosis model in a multiple-run experimental evaluation that included half a million objective function evaluations per run. The number of runs was set to eight. The optimal performing parameter setting was chosen based on the median best performance over all runs. A common approach is to use the mean best performance, but we took the median in order to avoid the problems that the mean has when observing large variance in the objective function values across the runs.

The parameters of the three meta-heuristic methods chosen for Sobol'-sampling-based parameter tuning and their ranges are summarized in Table 6.1. In the same table, we report the resulting (tuned) parameter settings for the three optimization methods. Note that the Sobol' sampling approach generates a number on the unit interval: in order to obtain the parameter settings, we had to map that value on the predefined range of parameter values. For the integer-valued parameters, the mapped value was rounded to the closest integer value. An additional note concerns the upper bound of the parameter K, denoting the size of the neighborhood in the PSO method, which can not be larger than the value $S - 1$. Its value was mapped according to the chosen value of the size of

Table 6.1: **Setup and results of Sobol'-sampling-based parameter tuning of optimization methods.** The table includes the search ranges (their lower and upper bound) for each of the four parameters of each of the three different meta-heuristic optimization methods that were tuned. We used the Sobol' sampling procedure with the number of sampling points set to 2000. The resulting vector of method's parameters was chosen as the one that showed best median performances (according to the SSE metric) in the multiple-run experiments among the 2000 sampled parameter settings. A single experiment included eight runs, each performed with half a million of objective function evaluations. The parameter tuning was performed on the complete observation scenario using noise-free artificial data.

Method	Parameter	Lower	Upper	Tuned
DASA	a	4	200	144
	ρ	0	1	0.036
	s_+	0	1	0.573
	s_-	0	1	0.01
PSO	S	4	200	155
	K	1	$S-1$	89
	w	0	1	0.762
	c	1	4	1.037
DE	P	6	200	81
	STR	1	10	8
	F	0	2	0.942
	CR	0	1	0.915

the swarm S. In a similar way, the upper bound of the global scale decrease factor s_- is limited by the value of the evaporation factor ρ in the DASA method. Finally, note that the parameter tuning was performed on the complete observation scenario using noise-free artificial data. The same parameter settings were then used across all scenarios and all datasets.

The parameter settings used in our experimental evaluation are given as follows.

DASA. The discretization base is set to 10, the maximum parameter precision is set to 10^{-15}, the number of ants is set to 144, the global scale increase factor to 0.575, the global scale decrease factor to 0.01, and the pheromone evaporation factor to 0.036.

PSO. A variable random topology was chosen, the particle swarm size was set to 155, the neighborhood size to 89, the inertia weight to 0.762, and the acceleration coefficient to 1.037. In addition, default settings were used for the remaining parameters related to advanced options not included in the standard PSO method.

DE. The chosen strategy was "DE/rand-to-best/1/bin", the population size was set to 81, the weight factor to 0.915, and the crossover factor to 0.942.

A717 setup. Since A717 is not a global search method, we wrapped the original procedure in a loop of restarts with randomly chosen initial points, providing in a way a simple global search. The procedure was restarted as many times as needed to achieve a comparable number of function evaluations to the other four methods. We used the module for bound constraint optimization with exact calculation of the derivatives.

6.4 Results and discussion

6.4.1 Parameter estimation from artificial data

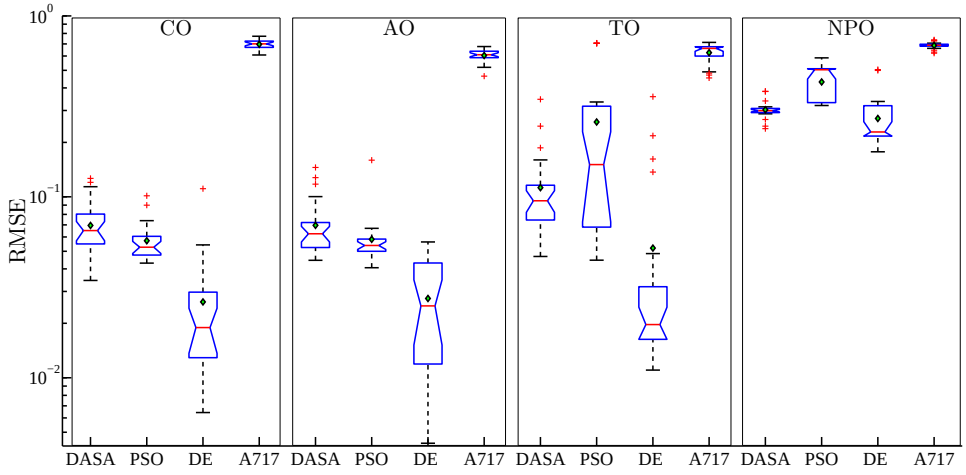
Given the artificial data described above (obtained using the reference values of the constant parameters from Eq. (6.3), we can calculate the value of the objective function at the reference point for each noise level and observation scenario: these are reported in Table 2. Note that the value of the objective function at the reference point, when considering noise-free data, is zero, while in the case of noisy data it increases and becomes greater than zero. The exception to this rule is the NPO observation scenario used in (Del Conte-Zerial et al., 2008), where the authors assume that the concentrations of passive-state proteins can be neglected when fitting the data. This assumption is obviously implausible, since it leads to large values of the objective function, even in the case of noise-free data.

Let us now consider the RMSE performance of the four parameter estimation methods (DASA, PSO, DE, and A717) on the artificial datasets with three levels of noise (0%, 5%, and 20%) under the four observation scenarios (CO, AO, TO, and NPO). Figure 6.2 summarizes the RMSE performance with boxplots over the 25 runs of each methods. The 12 graphs on the figure correspond to the four observation scenarios (in columns) and three artificial datasets (in rows), where each graph depicts the performance comparison of the four parameter estimation methods. The graphs show that the median performance of A717 is significantly worse than the performance of the three meta-heuristic methods. The comparison among the latter indicate that the median RMSE performance of DE is significantly better than the performance of DASA and PSO. These findings hold in all observation scenarios and at all noise levels. The performance comparison among different levels of noise shows a systematic decrease of the RMSE performance with the increasing noise level. The noise in the data affects the performance of all methods in all observation scenarios, but the magnitude of the effect differs. While we observe very large and remarkable differences in performance of the meta-heuristic methods in the noise-free case, there is much less difference in performance on the noisy datasets.

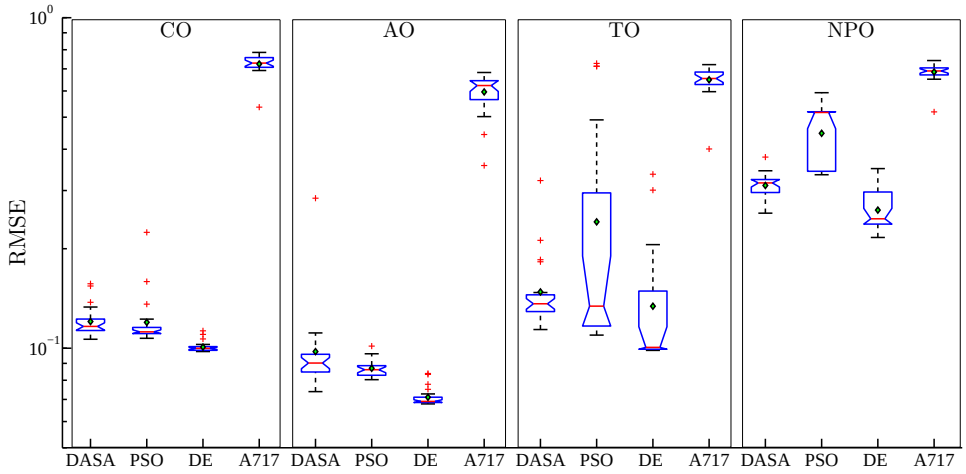
Table 6.2: **Values of the quality metrics for the reference model.** The reference model was used for generation of the artificial data.

Noise	Scenario	SSE	RMSE
0%	CO	0	0
	AO	0	0
	TO	0	0
	NPO	5549.839	1.413
5%	CO	2.653	0.031
	AO	1.289	0.022
	TO	2.591	0.031
	NPO	5556.486	1.414
20%	CO	42.447	0.124
	AO	20.627	0.086
	TO	41.452	0.122
	NPO	5607.516	1.420

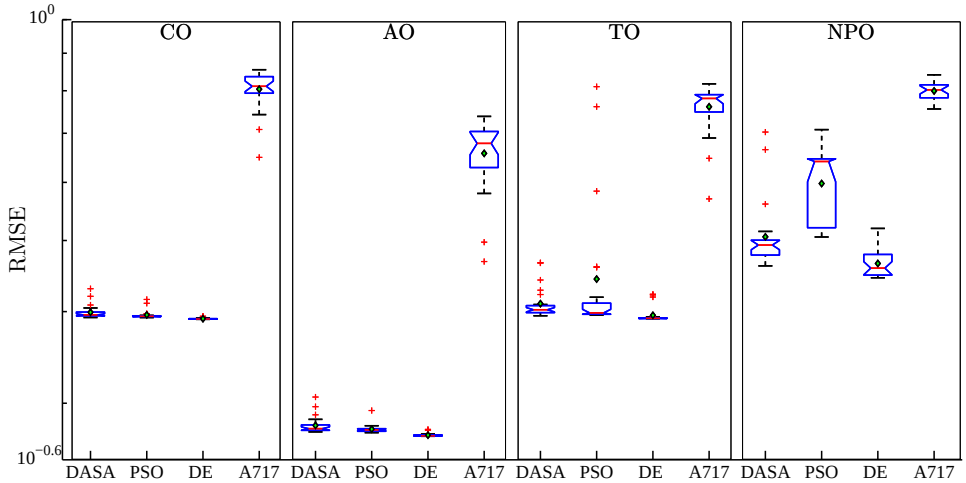
The comparison among observation scenarios shows that the CO and AO scenarios are very similar: they induce an identical ranking of the optimization methods in terms of



a)



b)



c)

Figure 6.2: **RMSE performance of the models obtained by parameter estimation from artificial data.** Boxplots of the performance distributions of the four optimization methods (DASA, PSO, DE, and A717) in terms of the quality of the reconstructed output (RMSE), when considering four different observation scenarios (columns CO, AO, TO, and NPO) and three artificial datasets (rows): a) noise-free, $s = 0\%$; b) noisy data, $s = 5\%$; and c) noisy data, $s = 20\%$. Due to the large differences in the order of magnitude, the RMSE values are plotted on a logarithmic scale.

performance at all noise levels. The rankings are slightly different (but still very similar) in the case of TO, and quite different in the implausible scenario (see the discussion above) of NPO. As the noise level increases, the AO scenario seems to become an easier task than the CO scenario leading to much better optimum values of RMSE, while the CO scenario becomes very similar to the TO scenario. In the NPO case, all four optimization methods overfit the observed output, leading to values of the objective function that are smaller than the value at the reference point from Table 6.2. Note also that the PSO and DE methods lead to a higher variance of RMSE across the different runs in the TO and NPO scenarios. Overall, the RMSE performance metric does not provide a clear and unified conclusion about the relative difficulty of the parameter estimation task under different observation scenarios.

However, comparing the RMSEm performance, *i.e.*, the quality of the complete model reconstruction, leads to much clearer conclusions about the relative difficulty of the four observation scenarios. Figure 6.3 summarizes the RMSEm performance. As one would expect, the easiest optimization tasks stem from the CO scenario, since in this scenario all the system variables are directly observed. However, note that the TO scenario, corresponding to the real biochemical measurement process, although complex (the observed outputs are linear combinations of the system variables) compares favorably to the other two scenarios in terms of complete system dynamics reconstruction. This can be explained by the fact that the observed outputs in the TO scenario carry more information about the system (they include both active and passive states of proteins) than the observed outputs (active-state proteins only) in the AO or NPO scenarios. The higher variance and evident outliers in the RMSEm values associated with the AO and NPO scenarios confirm that incomplete and/or misinterpreted measurements lead to more difficult optimization tasks.

In the CO and TO scenarios, the performance in terms of RMSEm of the four optimization methods follows a similar pattern as the one for the output reconstruction (RMSE) reported above: A717 is clearly and significantly inferior to the other methods, while DE is better than DASA and slightly better than PSO. In the other two scenarios (AO and NPO), there is no significant difference in performance between all four methods: PSO is handling the AO scenario slightly better than the other methods, while A717 performs better compared to the other methods (significantly better than DE) in the case of the NPO scenario when considering data with 5% noise.

The convergence curves in Figure 6.4 further confirm that DE is the most suitable method for parameter estimation in the endocytosis model for the given amount (half a million) of function evaluations. DE has faster convergence than DASA and PSO over all scenarios when considering noise-free data: in the CO and the TO case this is clear after 10 thousand evaluations, while in the AO and NPO case DE outperforms the others after one hundred thousand evaluations. The convergence rate of DE and the other methods is notably influenced by the noise level: regardless of the observation scenario, at 20% noise level there is no difference in the convergence rates of DASA, PSO, and DE, and it is clear that all methods (not only A717) have extremely slow convergence and seem to be trapped in local optima. Moreover, when comparing DASA and PSO, the convergence plots show that DASA is better in the TO and NPO scenario, while PSO has better convergence in the CO and AO scenario. A717 is clearly showing the poorest convergence, which is very little affected by the different observation and noise scenarios.

In order to assess the statistical significance of the differences in performance across all scenarios, two Holm tests were conducted using first the median values of RMSE and then the median values of RMSEm. The corresponding median values are given in Table 6.3, while the results of the Holm tests regarding both metrics are reported in

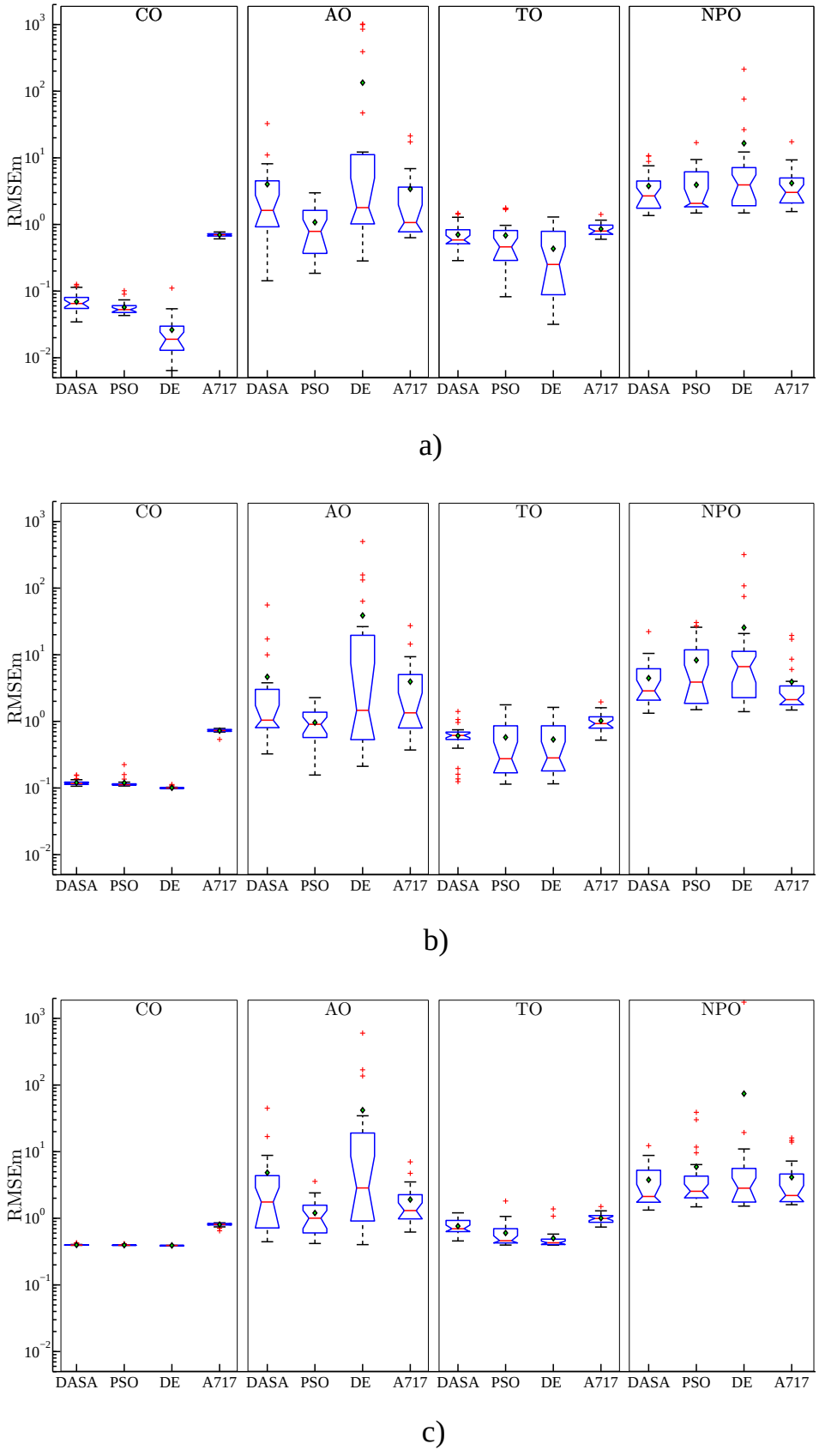


Figure 6.3: **RMSEm performance of the models obtained by parameter estimation from artificial data.** Boxplots of the performance distributions of the four optimization methods (DASA, PSO, DE, and A717) in terms of the quality of the complete model reconstruction (RMSEm), when considering four different observation scenarios (columns CO, AO, TO, and NPO) and three artificial datasets (rows): a) noise-free, $s = 0\%$; b) noisy data, $s = 5\%$; and c) noisy data, $s = 20\%$. Due to the large differences in the order of magnitude, the RMSEm values are plotted on a logarithmic scale.

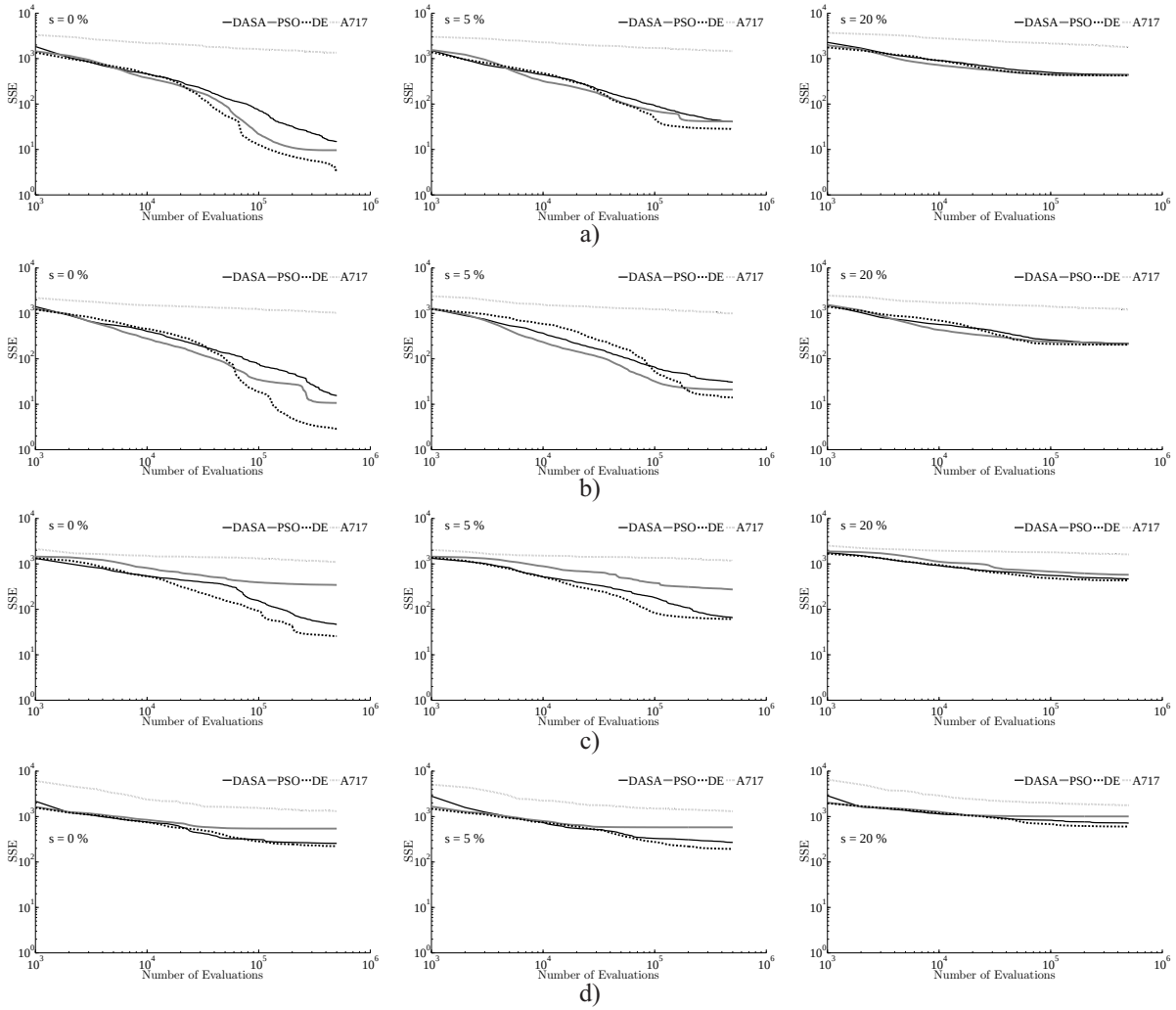


Figure 6.4: **Convergence of the optimization methods on the task of parameter estimation from artificial data.** Convergence curves of the four parameter estimation methods (DASA, PSO, DE, and A717) applied to three artificial datasets (columns) and four observation scenarios (rows): a) CO; b) AO; c) TO; and d) NPO. Graphs in the left column correspond to the noise-free dataset, while the graphs in the middle and right column correspond to the noisy datasets with 5% and 20% relative noise, respectively. In order to capture the convergence trend over a wide range of values, the convergence curves are plotted using logarithmic scales for both axes.

Table 6.3: **Results on RMSE and RMSEm of the models estimated from artificial data.** The table presents the median values of RMSE and RMSEm (over the 25 runs) of the models reconstructed with the parameters' estimates obtained by the four optimization methods from artificial data. The best values for both metrics are given in bold.

		RMSE				RMSEm			
Noise	Scenario	DASA	PSO	DE	A717	DASA	PSO	DE	A717
0%	CO	0.0651	0.0527	**0.0189**	0.7005	0.0651	0.0527	**0.0189**	0.7005
	AO	0.0625	0.0539	**0.0250**	0.6099	1.6272	**0.7866**	1.7876	1.0684
	TO	0.0951	0.1507	**0.0197**	0.6612	0.5857	0.4606	**0.2511**	0.7960
	NPO	0.2993	0.5040	**0.2282**	0.6881	2.6840	**2.0717**	3.9246	3.0273
5%	CO	0.1164	0.1121	**0.0999**	0.7287	0.1164	0.1121	**0.0999**	0.7287
	AO	0.0902	0.0861	**0.0690**	0.6232	1.0437	**0.9043**	1.4639	1.3442
	TO	0.1363	0.1341	**0.1006**	0.6546	0.6162	**0.2750**	0.2831	0.9265
	NPO	0.3162	0.5166	**0.2463**	0.6897	2.8668	3.8831	6.6315	**2.1172**
20%	CO	0.3958	0.3941	**0.3907**	0.8113	0.3958	0.3941	**0.3907**	0.8113
	AO	0.2770	0.2760	**0.2707**	0.6782	1.7547	**1.0050**	2.8513	1.3052
	TO	0.4023	0.3983	**0.3917**	0.7810	0.6967	0.4606	**0.4289**	0.9952
	NPO	0.4929	0.6407	**0.4585**	0.8023	**2.1250**	2.5423	2.8333	2.1999

Table 6.4. In terms of output reconstruction (RMSE), DE is the best ranked method that significantly outperforms the other three methods at the 0.05 significance level. In terms of complete system dynamics reconstruction (RMSEm), PSO is the best ranked method that significantly outperforms A717 at the same significance level, while the advantage over DASA and DE is not statistically significant.

Finally, we check whether good output reconstruction is related to good overall system dynamics reconstruction, which is of central interest to the modeler. In Table 6.5, we check the validity of the conjecture that "best according to the output reconstruction is best according to the complete model reconstruction". For each model, simulated with the best parameter estimates obtained by a single method in a single case (a single observation scenario with a single dataset), we report the corresponding RMSE and RMSEm values and expect that the optimization method that led to the best RMSE (the figure printed in bold in each row) would also lead to the best RMSEm (the figure printed in italic). This is trivially true in the CO scenario, where RMSE equals RMSEm, hence at all three noise levels DE leads to best RMSE and RMSEm. In the NPO scenario, this never happens, since DASA and PSO estimates lead to best RMSEm at different noise levels. In the other two scenarios, AO and TO, only at one out of three noise levels, DE leads to best RMSE and RMSEm; at the other two noise levels, the other meta-heuristic methods lead to best RMSEm. In sum, only in two out of nine non-trivial cases, models that perform best with respect to RMSE lead also to the best RMSEm performance.

6.4.2 Parameter estimation from measured data

Table 6.6 summarizes the results of parameter estimation in the Rab5-to-Rab7 conversion model using measurements obtained through real-world experiments. The rows Best, Median, and Worst give the RMSE value corresponding to the best, median and worst solution found by different optimization methods. The remaining two rows report the average RMSE performance (Average) and its standard deviation (Std). We consider only the observation scenarios TO and NPO, which are applicable given that the total protein

Table 6.4: **Results of the Holm test for a significance level of $\alpha = 0.05$.** The table summarizes the outcome of the Holm test performed on the median values of the RMSE and RMSEm results obtained by parameter estimation with the four optimization methods from artificial data. In the first test based on median values of the RMSE measure, DE is the reference method with rank $i = 0$. In the second test based on median values of the RMSEm measure, PSO is the reference method with rank $i = 0$. The hypothesis "there is no difference in performance between DE and the i-th ranked method" is rejected if the statement $p_i < \alpha/i$ holds.

			RMSE		
i	α/i	Method	z_i	p_i	Hypothesis
3	0.017	A717	5.69	$1.25 \cdot 10^{-8}$	Rejected
2	0.025	DASA	3.16	$1.57 \cdot 10^{-3}$	Rejected
1	0.050	PSO	2.53	$1.14 \cdot 10^{-2}$	Rejected

			RMSEm		
i	α/i	Method	z_i	p_i	Hypothesis
3	0.017	A717	2.53	$1.14 \cdot 10^{-2}$	Rejected
2	0.025	DASA	1.58	$1.14 \cdot 10^{-1}$	Accepted
1	0.050	DE	1.58	$1.14 \cdot 10^{-1}$	Accepted

concentrations are measured: the other two scenarios (CO and AO) are not applicable in this case. The first two graphs in Figure 6.5.a) visually summarize these results. The remaining four graphs, corresponding to the artificial noisy data (omitting the noise-free data as less likely in practice), are given as a reference for comparison.

The results on measured data confirm the findings of the experiments performed on artificial data. DE consistently leads to models with smallest RMSE (best performance), regardless of whether we consider the best, median, worst, or average RMSE (over the 25 runs). The boxplots clearly show the statistical significance of the performance differences between the four methods. DASA is the second best method, PSO is ranked third and A717 is ranked as the worst performing method. The observation about the higher variance of the RMSE values obtained by the PSO method in the TO and NPO scenarios with artificial data is confirmed in the experiments with measurement data. In the case of measured data, there is a very similar error distribution (range of values) in both scenarios (which is less expected given the definition of the scenarios), while in case the of artificial data, the NPO scenario is characterized with higher RMSE errors than the TO scenario. The error distribution in the measured data case is closer to the error distributions generated when considering artificial data with a noise level of 5%. Similarly, the convergence curves in Figure 6.6 resemble the ones for artificial data, i.e, the DE method converges faster to better solutions than the other three methods.

As a final test of the quality of the obtained models, we can visually compare the observed outputs with the outputs predicted by the models. In this context, Figure 6.10 visualizes the simulated output *vs.* the measured output (graphs on the left-hand side) and the complete dynamic behavior of all the system variables (graphs on the right-hand side) for the two models corresponding to the best parameters estimated by DASA and DE for the TO scenario. Figures 6.10–6.13 depict the predicted dynamics of the best model *vs.* the real-experimental behavior of the Rab5-to-Rab7 conversion for the TO and NPO scenarios found by DASA, PSO, DE, and A717, respectively. Since the time scale in the measured data was shifted for the sake of synchronization of the conversion

Table 6.5: **The RMSE and RMSEm values for the best model estimated from artificial data.** The table presents the RMSE and corresponding RMSEm values for the model simulated with the best parameters obtained by parameter estimation with the four optimization methods from artificial data. The best values for RMSE are marked in bold, while the best values for RMSEm are marked in italic.

		DASA		PSO	
Noise	Scenario	RMSE	RMSEm	RMSE	RMSEm
0%	CO	0.0345	0.0345	0.0430	0.0430
	AO	0.0446	6.0913	0.0406	*0.7693*
	TO	0.0468	1.0964	0.0447	*0.0877*
	NPO	0.2382	2.5430	0.3198	*1.9977*
5%	CO	0.1064	0.1064	0.1072	0.1072
	AO	0.0739	0.3343	0.0803	1.8387
	TO	0.1139	*0.1246*	0.1096	0.1639
	NPO	0.2562	3.0161	0.3349	*1.4970*
20%	AO	0.2742	1.3904	0.2735	*0.4750*
	TO	0.3948	0.4568	0.3955	0.4368
	NPO	0.4616	*1.8011*	0.5055	2.6218

		DE		A717	
Noise	Scenario	RMSE	RMSEm	RMSE	RMSEm
0%	CO	**0.0064**	*0.0064*	0.6080	0.6080
	AO	**0.0043**	1.1807	0.4644	21.3690
	TO	**0.0110**	0.1074	0.4542	0.6150
	NPO	**0.1774**	12.2287	0.6220	3.0273
5%	CO	**0.0977**	*0.0977*	0.5363	0.5362
	AO	**0.0678**	*0.2424*	0.3570	0.3723
	TO	**0.0985**	0.5058	0.4007	0.9028
	NPO	**0.2163**	318.415	0.5189	1.9670
20%	AO	**0.2704**	1.6916	0.4680	0.6220
	TO	**0.3913**	*0.3954*	0.5698	1.2933
	NPO	**0.4448**	5.4207	0.7556	7.2268

Table 6.6: **Results on RMSE of the models estimated from measured data.** The table presents the RMSE values associated with the predicted models (over 25 runs) obtained by parameter estimation with the four optimization methods from measured data. The best values regarding all statistics are given in bold.

Scenario		DASA	PSO	DE	A717
TO	Best	0.0661	0.0752	**0.0599**	0.2482
	Median	0.0744	0.2032	**0.0643**	0.2782
	Worst	0.1530	0.2045	**0.0682**	0.2898
	Average	0.0782	0.1494	**0.0647**	0.2749
	Std	0.0163	0.0627	**0.0029**	0.0124
NPO	Best	0.0665	0.0825	**0.0623**	0.2453
	Median	0.0799	0.1942	**0.0649**	0.3964
	Worst	0.1788	0.2338	**0.0698**	0.4920
	Average	0.0924	0.1680	**0.0654**	0.3857
	Std	0.0305	0.0471	**0.0019**	0.0724

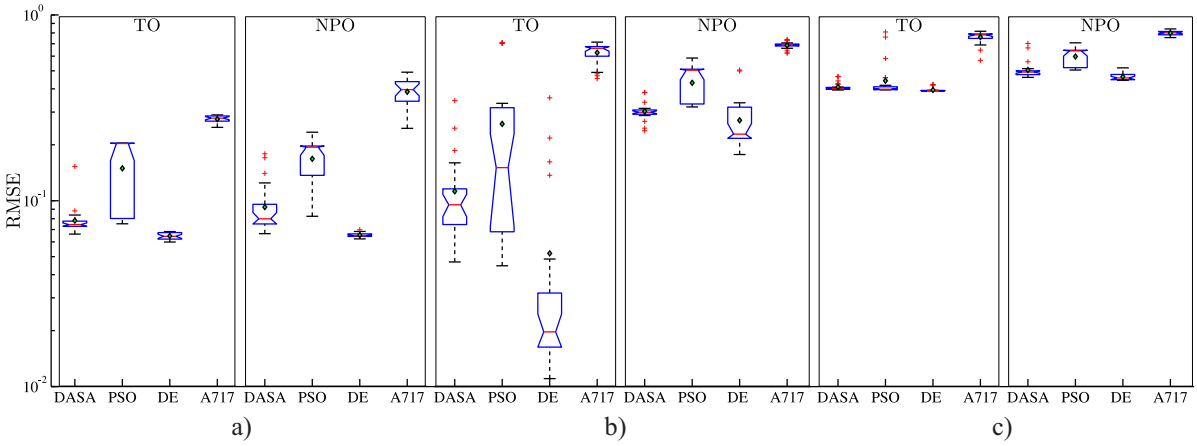


Figure 6.5: **RMSE performance of the models obtained by parameter estimation from measured data.** Boxplots of the performance distributions of the four optimization methods (DASA, PSO, DE, and A717) in terms of the reconstructed output (RMSE), when considering two different observability scenarios (columns TO and NPO) and three datasets: a) measured data, b) artificial data with $s = 5\%$ relative noise; and c) artificial data with $s = 20\%$ relative noise. Graphs b) and c) are the same as the corresponding graphs from Figure 6.2. Due to the large differences in the order of magnitude, the RMSEm values are plotted on a logarithmic scale.

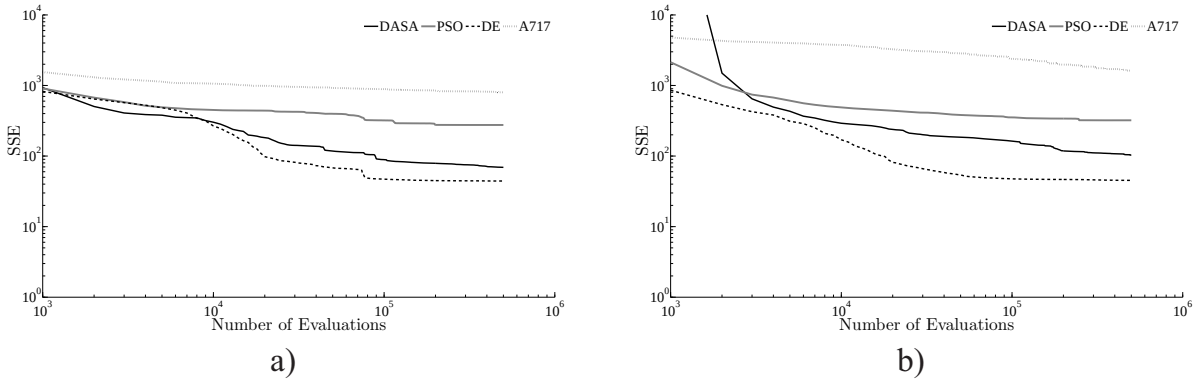


Figure 6.6: **Convergence of the optimization methods on the task of parameter estimation from measured data.** Convergence curves of the four parameter estimation methods (DASA, PSO, DE, and A717) when considering two observation scenarios: a) TO and b) NPO. In order to capture the convergence trend over a wide range of values the convergence curves are plotted using logarithmic scales for both axis.

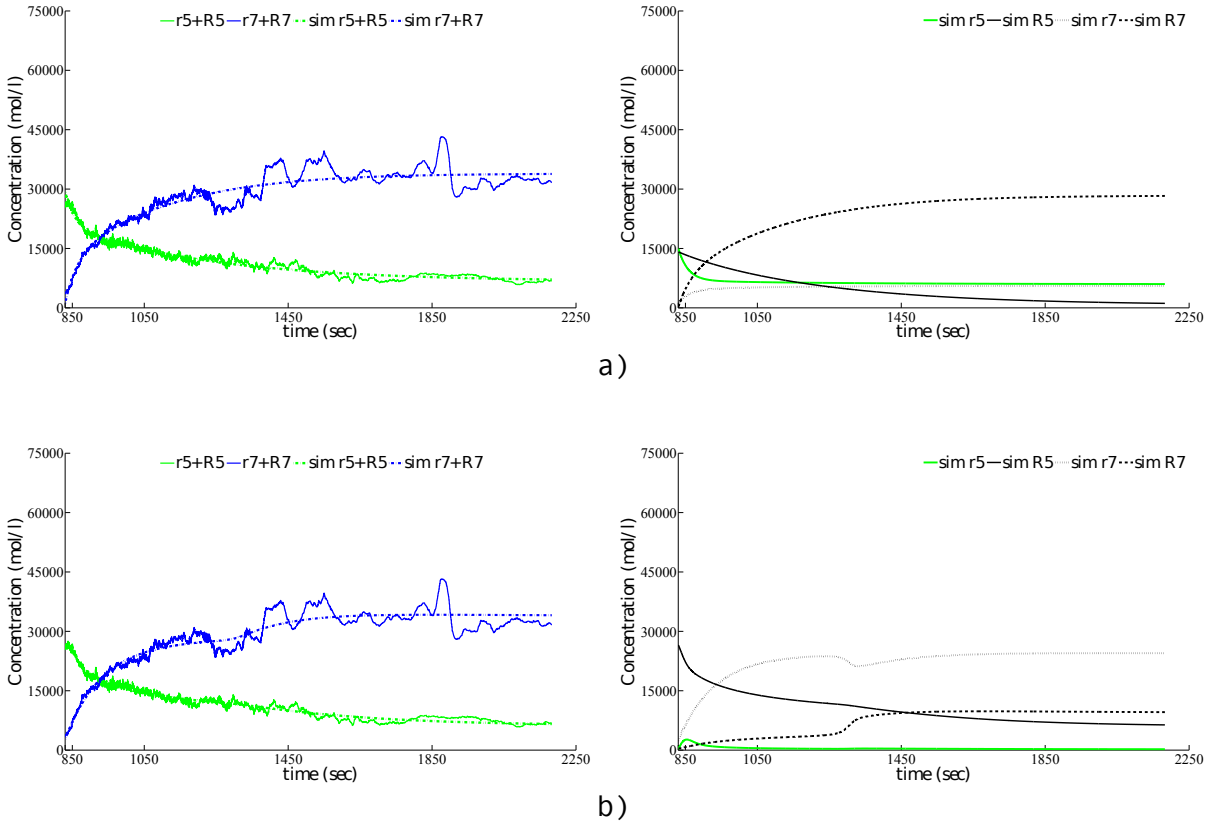


Figure 6.7: **Simulated dynamics of the best models obtained by parameter estimation from measured data in the TO observation scenario.** Experimental (observed) behavior *vs.* predicted (simulated) behavior shown in terms of the reconstructed output (left-hand side) and the reconstructed model dynamics (right-hand side) of the model with the best parameters' estimates obtained from measured data in the TO observation scenario using: a) DASA and b) DE.

events of Rab5 and Rab7 domain proteins, we rescaled it in order to conduct a direct comparison of the numeric simulation (reference model) with the measured data: we used the transformation $t \leftarrow 4\,t + 850$.

The left-hand side graphs in Figure 6.13 show that A717 fails to reconstruct the Rab5-to-Rab7 dynamics. In both observation scenarios, A717 fails to find a model with the cut-out switch behavior observed in Figure 6.1. On the other hand, DASA, PSO, and DE, when following the TO scenario, are able to almost perfectly reconstruct the real-experimental data (see left-hand side graph in: Figure 6.10.a), and Figures 6.11.a) and 6.12.b)), complete with the time point when the switch in the total (active- and passive-state) protein concentrations occurs. In the NPO scenario, DASA, PSO, and DE are also able to almost perfectly match the 'assumed' model output (active-state protein concentrations) against the measured system output (total protein concentration), but if we compare the model outputs, we can see that only DE is able to reconstruct the cut-out switch (even though slightly shifted on both scales: see graph on the left-hand side of Figure 6.12.b)), while DASA and PSO are only able to fit one protein domain, due to the completely different dynamics of the simulated passive-state protein concentration (see graphs on the left-hand side of Figures 6.10.b) and 6.11.b)).

Finally, the analysis of reconstructed model dynamics (complete simulation of all the system variables) of the obtained models reveals further details about their quality. Note first that the simulated behavior in Figure 6.1 shows that the passive-state protein concentrations are almost constant all the time, leading to the conclusion that the cut-out switch appears at the very same time point for both total and active-state protein concentrations. For the TO scenario, only the model obtained with DASA has these two properties of the original model behavior (see its complete simulation on the right-hand side of Figure 6.12.a)), while in the NPO scenario this holds for the model obtained by DE (see its complete simulation on the right-hand side of Figure 6.12.b)). All the other models do not match one or both properties: the complete simulation graphs show that the concentrations of passive-state proteins vary and/or the switch of the active-state concentrations takes place at a different time point (or at all). Further comparison of the simulated behaviors with the original cut-out switch model behavior from Figure 6.1 also shows that the ratio between the active- and passive-state protein concentrations in the simulation of the model obtained with DASA in the TO case is closer to the same ratio in the original model.

Overall, the results on measured data show that all three meta-heuristic methods are far better than A717. Among the meta-heuristic methods, DE has a clear advantage over the other two methods, both in terms of the convergence rate and in terms of the reconstruction of model output. In terms of other relevant qualitative aspects of the behavior of the obtained models, *i.e.*, the time point of the switch and the ratio between active- and passive-state protein concentrations, one of the other two methods (DASA) performs better then DE. However, note that these qualitative aspects have not been included in the objective function (SSE) used by the optimization methods: thus, we cannot objectively and fairly compare the methods along this dimension.

6.4.3 Parameter values and practical parameter identifiability

Table 6.7 compares the reference parameter values from Eq. (6.3) with the best parameter values obtained using each of the four parameter estimation methods for the CO and TO scenarios on artificial data with 20% relative noise. Tables 6.8–6.11, present the same comparison in terms of relative error of the estimated parameters with respect to all four observation scenarios on artificial data with 0%, 5%, and 20% noise for DASA, PSO, DE,

and A717, respectively, while Table 6.12 presents the best parameter values obtained by the four parameter estimation methods on measured data, for the TO and NPO observation scenarios. Despite the fact that DE finds parameter values that lead to low values of the objective function, the obtained parameter values differ quite substantially from the reference parameter values. Results on artificial data show this same pattern of large differences between the estimated and reference parameter values for all four parameter estimation methods in all scenarios; except for the parameters c_4, c_8, c_{12}, c_{15}, $r_5(0)$, and $r_7(0)$, the relative error of the other estimated parameters is over 100%. On measured data, we do not have reference values, but the comparison (Table 6.12) with the parameter values proposed by Del Conte-Zerial et al. (2008) reveals the same pattern of large differences.

Table 6.7: **Best estimated values of the model parameters obtained from artificial data with 20% relative noise.** Best parameter values as estimated by the four optimization methods (DASA, PSO, DE, and A717) from artificial data with 20% relative noise in the CO and TO observation scenarios.

		CO				TO			
c	c^*	DASA	PSO	DE	A717	DASA	PSO	DE	A717
c_1	1	4.0000	1.4644	0.2226	1.6393	3.1593	1.5627	1.8293	0.2974
c_2	0.3	3.7099	2.0786	1.1132	2.5748	3.7499	2.7324	4.0000	2.5086
c_3	0.1	0.1977	1.2612	0.1974	0.0229	3.3526	0.2064	0.2682	0.5823
c_4	2.5	3.5412	0.4192	3.1208	3.3226	0.2007	1.5837	3.7871	0.1709
c_5	1	3.9940	1.4613	0.2217	1.5411	3.1287	1.4623	1.7688	0.5845
c_6	0.483	0.5165	3.0640	3.6860	1.3897	0.4074	1.8952	0.4713	1.9140
c_7	0.21	3.9471	2.6526	0.1503	1.5383	1.7030	2.9345	3.4951	2.0874
c_8	3	3.1843	1.5314	3.4591	1.6254	1.5254	1.6742	3.1784	3.5895
c_9	0.1	0.1563	1.5057	0.0524	3.1257	1.6444	1.5321	0.9762	1.9652
c_{10}	0.021	2.0757	1.8316	0.0645	2.4551	0.2091	2.1490	1.4581	1.1780
c_{11}	1	1.8340	2.9039	2.2013	2.3769	2.5222	3.2725	1.9195	2.6830
c_{12}	3	3.1572	2.2358	1.7009	2.8349	1.2553	1.5096	0.1557	1.2227
c_{13}	0.31	4.0000	1.7187	0.9381	0.6126	4.0000	2.9568	3.4364	3.2812
c_{14}	0.3	1.0661	1.3179	0.3833	2.2955	1.7539	1.1975	0.8110	2.4085
c_{15}	3	2.3525	1.7764	3.9800	3.6281	2.1599	2.1684	3.5535	3.3994
c_{16}	0.483	0.5178	3.0728	3.6981	1.2091	0.4224	2.0041	0.7261	2.7693
c_{17}	0.06	1.8635	0.4696	0.4159	2.0548	0.8316	1.3081	1.7687	0.2836
c_{18}	0.15	2.7213	1.0043	0.1087	0.4984	0.5992	1.0744	1.3643	0.6677
$r_5(0)$	1.0	0.8750	0.9116	0.9122	0.9535	0.9957	0.4830	0.9239	1.4011
$R_5(0)$	0.001	4.0E-07	0.0358	0.1194	1.0854	3.3E-07	0.2313	0.0000	1.3742
$r_7(0)$	1.0	0.8096	1.3352	0.7978	0.1557	1.0153	0.4473	0.7310	1.0320
$R_7(0)$	0.001	1.2E-10	0.2444	3.4E-04	0.8451	0.0139	0.1696	0.2515	0.9143

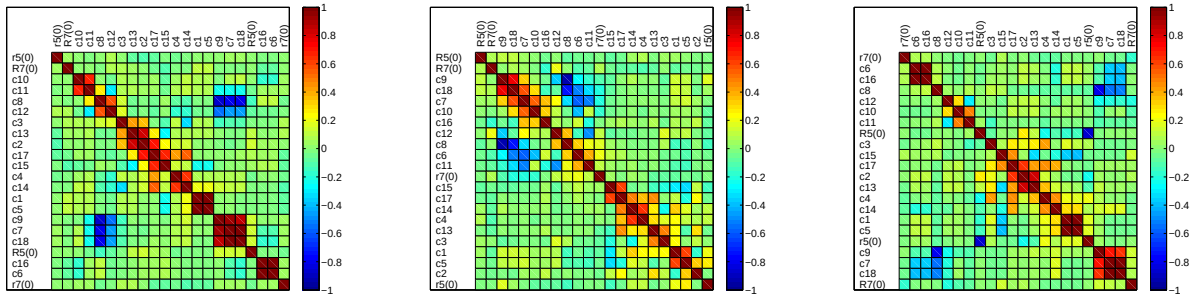
Evidently, many quite different sets of parameter values produce behaviors that resemble the reference model behavior, suggesting that the endocytosis modeling task, as many others in systems biology, has parameter identifiability problems. Indeed, a systematic study of seventeen systems biology models (Gutenkunst et al., 2007) has found parameter identifiability issues in each of them. To empirically confirm this conjecture about our model, we performed a practical parameter identifiability test using the Monte Carlo-based approach and DE as a parameter estimation method. We considered this test in three observations scenarios, CO, AO, and TO, using data with 20% noise. The results of the test in the different observation scenarios confirm that the considered parameter

estimation task has identifiability issues. The results reveal high relative errors; the mean value of the estimated parameters are overall far from the reference ones. Except for the parameters c_4, c_8, c_{12}, c_{14}, c_{15}, $r_5(0)$, and $r_7(0)$, the relative error of the other estimated parameters is over 100%. This observation additionally re-confirms the statements in the previous paragraph obtained from the results regarding all optimization methods. In extreme cases, the relative errors are over 1000%, as it is the case with the parameters: c_{10} (in all scenarios), c_{17} (in CO and TO scenarios), $R_5(0)$ (in CO and TO scenarios), and $R_7(0)$ (in TO scenario). Furthermore, the calculated uncertainties (95% confidence interval) of the parameters are large, especially for c_7, $R_5(0)$, and $R_7(0)$ over all scenarios; see Tables 6.13–6.15 that summarize the results of the Monte Carlo-based approach for the CO, AO, and TO scenario, respectively.

Furthermore, the estimated values for many model parameters are evenly distributed across the parameter ranges; see the histograms of the distributions of the estimated parameter values in Figures 6.14–6.16. If we take a look at the histograms for the CO scenario, we can see that parameters like c_1, c_2, c_5, c_8, c_{11}, and c_{13} have very similar (almost) uniform distributions with higher concentration of the estimates on the bounds of the allowed range. We observe similar distributions for most of these parameters in the AO scenario (including $r_5(0)$, $r_7(0)$, c_6, and c_{16}) and in the TO scenario (including c_6, c_{15}, and c_{16}) as well. For some parameters (like c_3, c_{12}, and c_{13} in the CO and AO scenario) the confidence interval does not include the reference value of the parameter, emphasizing the complexity of the optimization problem and the objective function. A closer look at the histograms reveals that some pairs of parameters have very similar (or almost identical) distributions of the estimates: This is in general the case with the (c_1, c_5) and (c_6, c_{16}) pairs of parameters. Note also how the distributions of the initial values of the system variables $r_5(0)$ and $r_7(0)$ differ among scenarios. In the case of complete observability, their values follow (almost) a Gaussian distribution around the reference value. In the TO scenario, most of the estimated initial values are in the neighborhood of the reference values even though the relative errors are higher than in the CO scenario. However, in the AO scenario, the distribution does not resemble a Gaussian; the values are spread all over the corresponding ranges, with higher concentrations at the ranges' limits and far from the reference values. Evidently, the lack of information on the concentration of passive-state proteins (r_5 and r_7) in the data worsens the problems related to parameter identifiability.

The correlation matrices for the estimated parameter values, presented in Figure 6.8, re-confirm the practical identifiability problems, by emphasizing several pairs of correlated parameters. In the CO scenario, there are seven pairs of highly correlated parameters: A correlation $R > 0.9$ is evident for the (c_6, c_{16}), (c_1, c_5), (c_7, c_{18}), and (c_8, c_9) pairs of parameters, while the pairs (c_8, c_9), (c_8, c_{18}), and (c_2, c_{13}) have correlations in the range $0.84 < |R| < 0.9$. In the AO scenario, the most correlated are c_8 and c_9, while in the TO scenario there are six such pairs: the pairs (c_6, c_{16}), (c_1, c_5), and (c_7, c_{18}) have almost prefect linear correlation $R > 0.99$, while the $(r_5(0), R_5(0))$, (c_2, c_{13}), and (c_8, c_9) pairs have a correlation in the range $0.81 < |R| < 0.85$. In the last case, the correlation between the initial values of passive-state and active-state protein concentrations is expected, since we only observe their sum in the TO scenario.

The high pairwise correlations can be observed visually in Figure 6.9, where the scatter plots of the obtained solutions are combined with the contour plots of the objective function landscape for selected pairs of parameters. For example, observe the long diagonal valley with many (almost) equally good solutions for the (c_1, c_5) and (c_6, c_{16}) pairs of parameters in Figure 6.9.a) (left-hand side) and Figure 6.9.c) (left-hand side). We observe a similar pattern of behavior for these pairs of parameters in both the CO and



a) b) c)

Figure 6.8: **Correlation matrices for the model parameters generated with a Monte Carlo-based approach (estimated by DE from artificial data with 20% relative noise).** Colored matrix cells visualize the correlation R for parameter pairs based on a scale-to-color mapping. The cells on the main diagonal represent the self-correlations of the parameters (they are equal to 1). The most correlated pairs of parameters per observations scenarios are: a) $R(c_6, c_{16}) = 0.99997$, $R(c_1, c_5) = 0.99997$, $R(c_7, c_{18}) = 0.9862$, $R(c_7, c_9) = 0.9093$, $R(c_8, c_9) = -0.8749$, $R(c_9, c_{18}) = 0.8568$, $R(c_2, c_{13}) = 0.8413$ in the case of CO; b) $R(c_8, c_9) = -0.9097$ and $R(c_9, c_{18}) = -0.7789$ in the case of AO; and c) $R(c_6, c_{16}) = 0.9980$, $R(c_1, c_5) = 0.9934$, $R(c_7, c_{18}) = 0.9901$, $R(r_5(0), R_5(0) = -0.8509$, $R(c_2, c_{13}) = 0.8343$, and $R(c_8, c_9) = -0.8105$ in the case of TO. Smallest values for R are obtained for the following pairs of parameters: a) $R(c_1, c_{18}) = -0.00196$ in the case of CO; b) $R(c_3, c_5) = 0.000032$ in the case of AO; and c) $R(c_4, c_6) = -0.0011$ in the case of TO.

TO scenarios; see Figures 6.17 and 6.18. Figure 6.9.b), corresponding to the AO scenario, confirms that the objective function is characterized with elongated elliptical contours for the parameter pairs (c_8, c_9) as well. While the above-mentioned examples of correlated parameters are related to the lack of practical identifiability, the plot for the c_7 and c_{18} parameters on the right-hand side in Figure 6.9.a) (see Figure 6.18 for the TO case as well) indicates structural non-identifiability of c_7 in the considered search interval; we observe a very large flat region in the part of the space $0.5 < c_{18} < 4$, where c_7 can take any value and does not influence the objective function. A similar observation holds for the c_9 and c_{18} parameters in Figure 6.9.b) (see Figure 6.17 for the CO case as well), in which case the c_9 parameter seems to be structurally non-identifiable. Finally, the right-hand side plot in Figure 6.9.c) re-confirms the correlation of the initial conditions, $r_5(0)$ and $R_5(0)$.

6.5 Summary

The focus of this chapter was the use of meta-heuristic optimization methods for parameter estimation in dynamic system models typical of systems biology. More specifically, we addressed the task of parameter estimation in a practically relevant model of endocytosis that captures the nonlinear dynamics of endosome maturation reflected in a cut-out switch transition between the Rab5 and Rab7 domain protein concentrations. The model is nonlinear and has many, potentially unidentifiable, parameters which have to be estimated from partial observations.

We applied three global-search meta-heuristic methods for numerical optimization, i.e., the differential ant-stigmergy algorithm (DASA), particle-swarm optimization (PSO), and differential evolution (DE), as well as Algorithm 717 (A717), a derivative-based local search method, to the task of estimating the parameters in the ODE model of the Rab5-to-Rab7 conversion process. We evaluated their performance on the considered representative task along a number of metrics, including the quality of reconstructing the

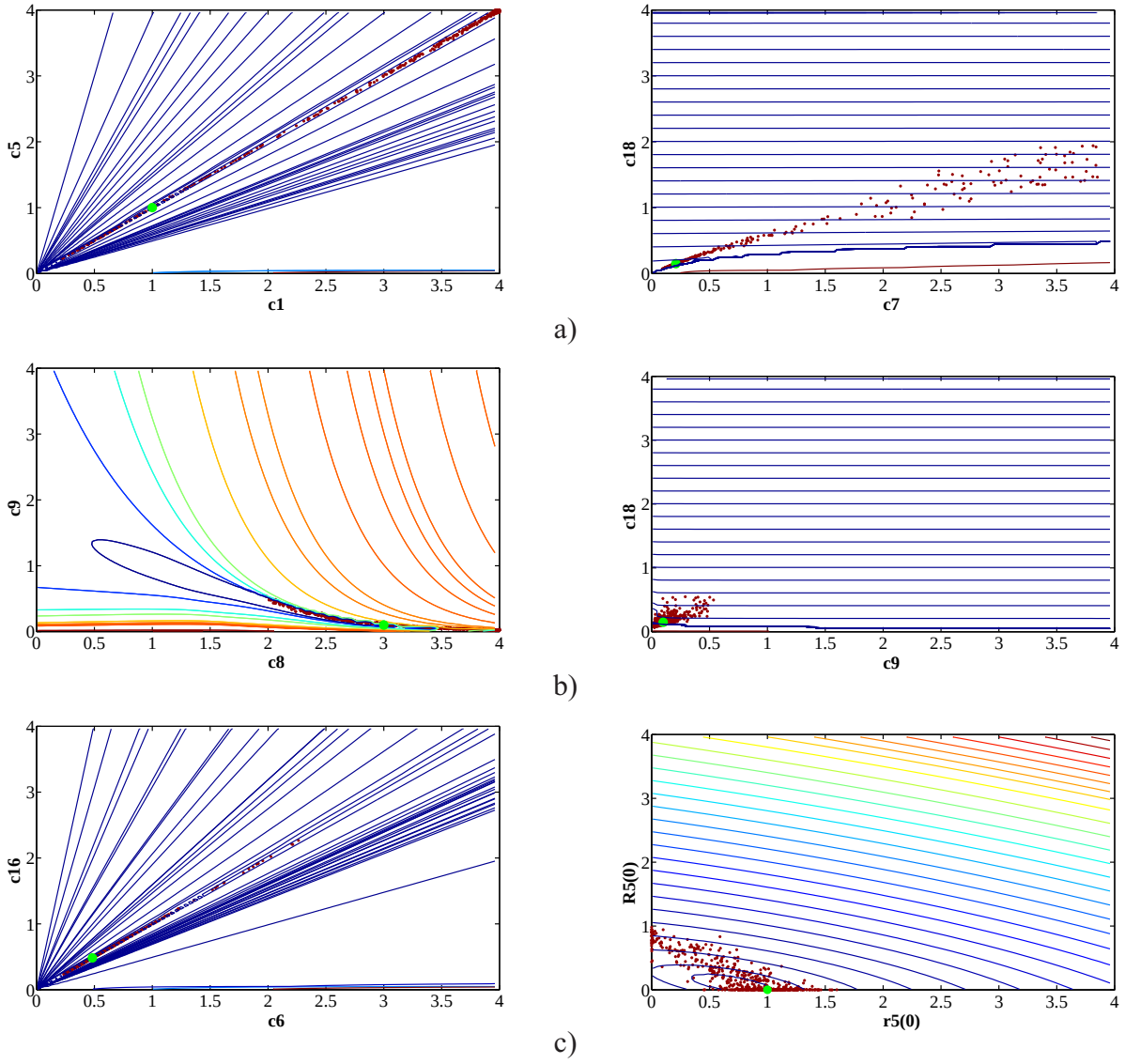


Figure 6.9: **Contour plots of the objective function with scatter plots of the parameters' estimates generated with a Monte Carlo-based approach (estimated by DE from artificial data with 20% relative noise).** The plots correspond to two representative pairs of correlated parameters in the observation scenarios: a) CO; b) AO; and c) TO. Note that one pair of correlated parameters in the TO observation scenario corresponds to the initial values of the Rab5 protein. The green dot represents the reference value of the parameter from Eqs. (6.3) and (6.3). The red dots are the parameters' estimates obtained by DE from datasets generated in the Monte Carlo-based fashion.

system output and the complete dynamics, as well as the speed of convergence, both on real-experimental data and on artificial pseudo-experimental data with varying amounts of noise. We compared the four optimization methods under a range of observation scenarios, where data of different completeness and accuracy of interpretation were given as input.

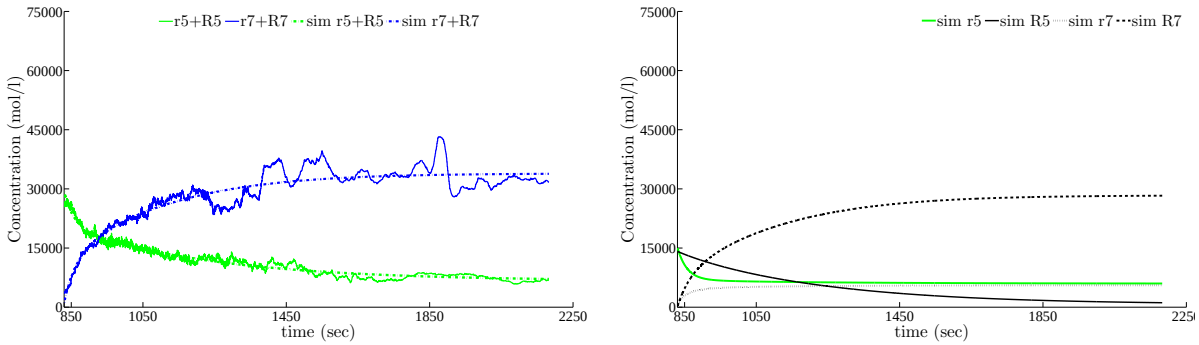
Overall, the global meta-heuristic methods (DASA, PSO, and DE) clearly and significantly outperform the local derivative-based method (A717). Among the three meta-heuristics, differential evolution (DE) performs best in terms of the objective function, i.e., reconstructing the output, and in terms of convergence. These results hold for both real and artificial data, for all observability scenarios considered, and for all amounts of noise added to the artificial data. In sum, the meta-heuristic methods considered are suitable for estimating the parameters in the ODE model of the dynamics of endocytosis under a range of conditions: With the model and conditions being representative of parameter estimation tasks in ODE models of biochemical systems, our results clearly highlight the promise of bio-inspired meta-heuristic methods for parameter estimation in dynamic system models within systems biology.

6.6 Additional figures and tables

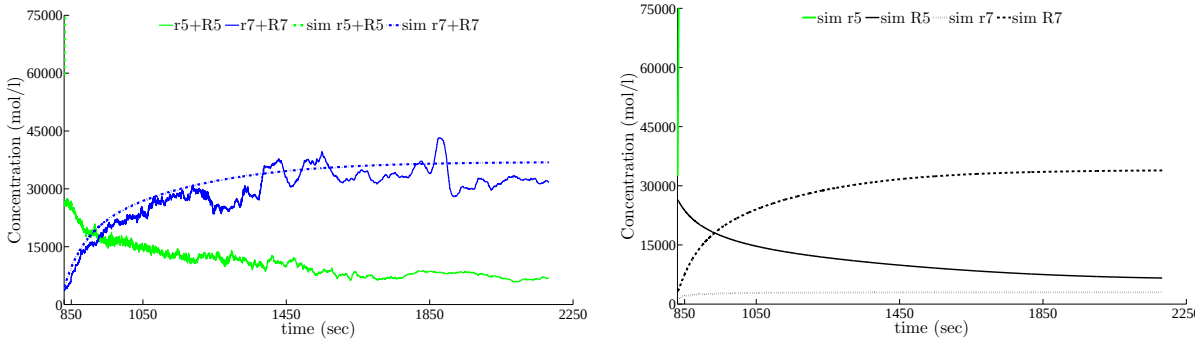
This section contains Figures 6.10– 6.18, and Tables 6.8–6.15 with additional results regarding the task of parameter estimation in the Rab-to-Rab7 conversion model. The corresponding figures and tables are properly referenced in Section 6.4, which discusses all results obtained from the experiments regarding the parameter estimation task at hand.

Four figures visualize the experimental behavior vs. simulated behavior of the reconstructed output and reconstructed model dynamics with the best parameter values estimated from measured data by the four optimization methods: DASA (Figure 6.10), PSO (Figure 6.11), DE (Figure 6.12), and A717 (Figure 6.13). Three figures depict the distributions of the estimated parameter values generated with a Monte Carlo-based approach, using DE as the baseline estimation method and artificial data with 20% relative noise, each corresponding to one observation scenario: CO (Figure 6.14), AO (Figure 6.15), and TO (Figure 6.16). The last two figures combine the scatter plots of the Monte Carlo-based DE parameter estimates with the contour plots of the objective function when considering artificial data with 20% noise for the most correlated pairs of parameters in the CO (Figure 6.17) and TO (Figure 6.18) observation scenarios.

Four tables present the relative errors of the best parameter values estimated from artificial data by the four optimization methods: DASA (Table 6.8), PSO (Table 6.9), DE (Table 6.10), and A717 (Table 6.11). Table 6.12 presents the parameter values associated with the best solutions estimated from measured data. Finally, the summary statistics for the estimated parameter values generated with a Monte Carlo-based approach, using DE as the baseline estimation method and artificial data with 20% relative noise are provided in three tables, each corresponding to one observation scenario: CO (Table 6.13), AO (Table 6.14), and TO (Table 6.15).



a)



b)

Figure 6.10: **Simulated dynamics of the best models obtained by DASA from measured data.** Experimental (observed) behavior *vs.* predicted (simulated) behavior shown in terms of the reconstructed output (left-hand side) and the reconstructed model dynamics (right-hand side) of the model with the best parameters' estimates obtained by DASA from measured data in the observation scenarios: a) TO and b) NPO. Note that in the case of NPO scenario, the simulated concentration of the passive-state Rab5 protein ("sim r5", green solid line) is instantly and rapidly increasing towards higher values until it reaches some stable level (approx. 200000 mol/l), therefore it is invisible on the right-hand graph. Related to this, the predicted total Rab5 concentration ("sim r5+R5", green dashed-dotted line) is almost invisible on the left-hand graph as well.

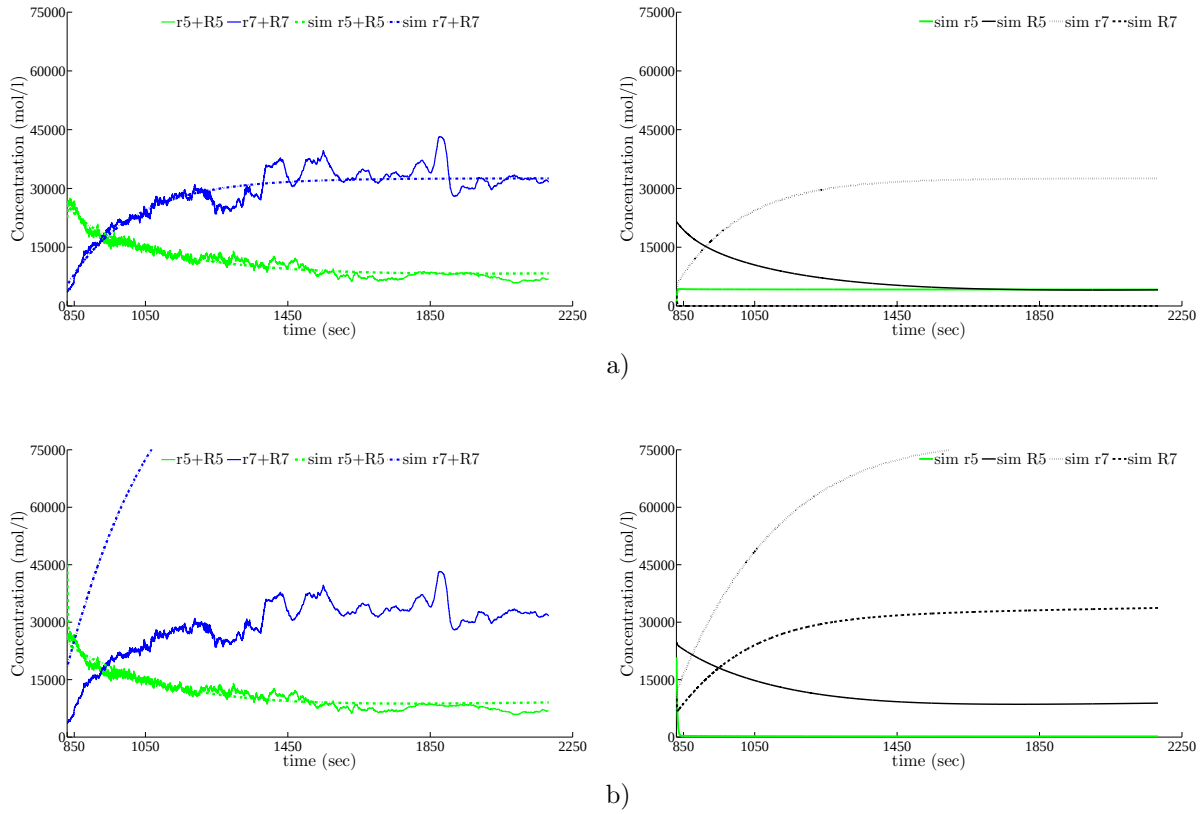
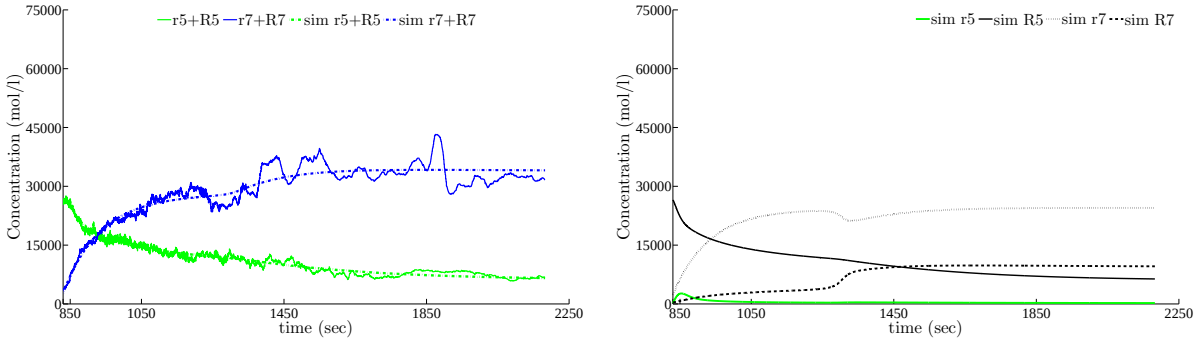
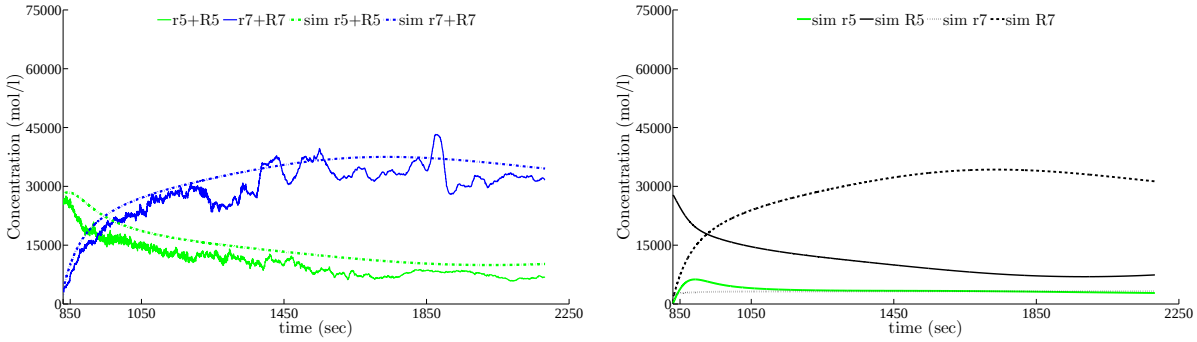


Figure 6.11: **Simulated dynamics of the best models obtained by PSO from measured data.** Experimental (observed) behavior *vs.* predicted (simulated) behavior shown in terms of the reconstructed output (left-hand side) and the reconstructed model dynamics (right-hand side) of the model with the best parameters estimated by PSO from measured data in the observation scenarios: a) TO and b) NPO. Note that in the case of NPO scenario, the simulated concentration of the passive-state Rab5 protein ("sim r5", green solid line) is instantly and rapidly increasing towards higher values until it reaches some stable level (approx. 200000 mol/l), therefore it is invisible on the right-hand graph. Related to this, the predicted total Rab5 concentration ("sim r5+R5", green dashed-dotted line) is almost invisible on the left-hand graph as well.

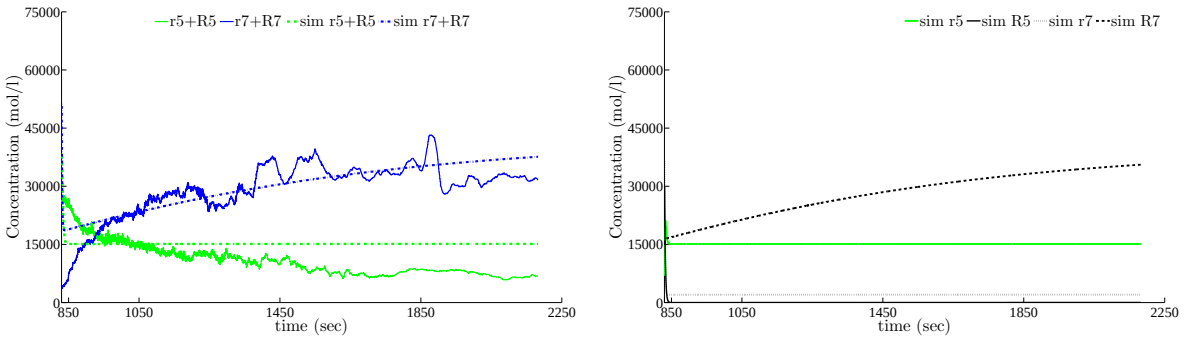


a)

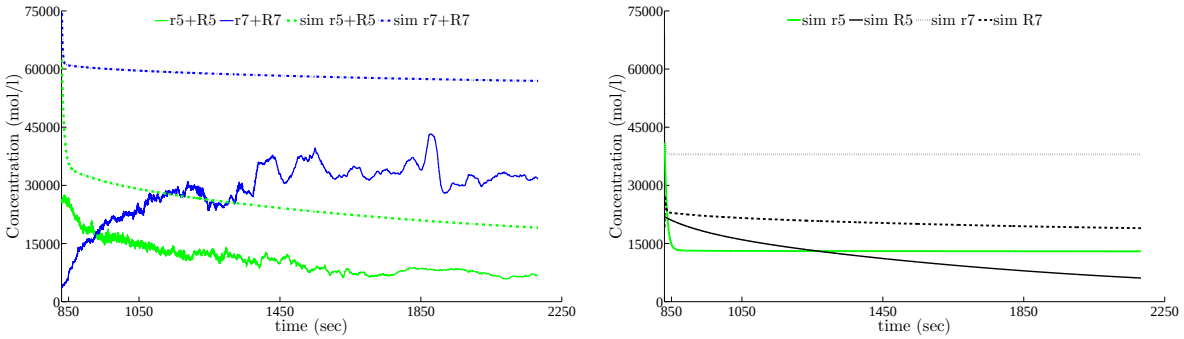


b)

Figure 6.12: **Simulated dynamics of the best models obtained by DE from measured data.** Experimental (observed) behavior *vs.* predicted (simulated) behavior shown in terms of the reconstructed output (left-hand side) and the reconstructed model dynamics (right-hand side) of the model with the best parameters estimated by DE from measured data in the observation scenarios: a) TO and b) NPO. Note that in the case of NPO scenario, the simulated concentration of the passive-state Rab5 protein ("sim r5", green solid line) is instantly and rapidly increasing towards higher values until it reaches some stable level (approx. 200000 mol/l), therefore it is invisible on the right-hand graph. Related to this, the predicted total Rab5 concentration ("sim r5+R5", green dashed-dotted line) is almost invisible on the left-hand graph as well.



a)



b)

Figure 6.13: **Simulated dynamics of the best models obtained by A717 from measured data.** Experimental (observed) behavior *vs.* predicted (simulated) behavior shown in terms of the reconstructed output (left-hand side) and the reconstructed model dynamics (right-hand side) of the model with the best parameters estimated by A717 from measured data in the observation scenarios: a) TO and b) NPO. Note that in the case of NPO scenario, the simulated concentration of the passive-state Rab5 protein ("sim r5", green solid line) is instantly and rapidly increasing towards higher values until it reaches some stable level (approx. 200000 mol/l), therefore it is invisible on the right-hand graph. Related to this, the predicted total Rab5 concentration ("sim r5+R5", green dashed-dotted line) is almost invisible on the left-hand graph as well.

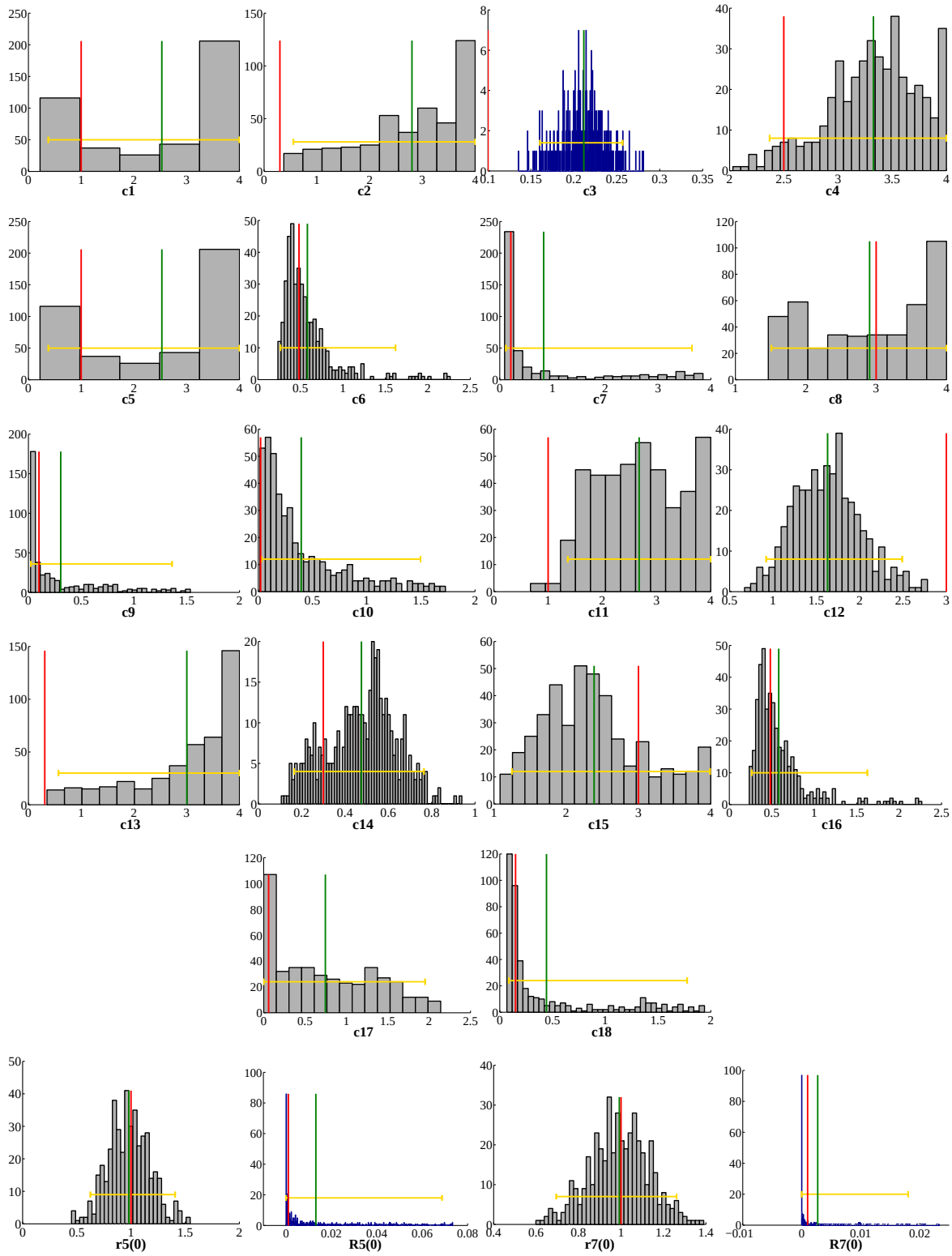


Figure 6.14: **Histograms of the parameter values generated with a Monte Carlo-based approach (estimated by DE using the CO observation scenario and artificial data with 20% relative noise).** The red vertical line represents the "true" value of the estimated parameter; the green vertical line represents the mean value of the sample μ; the yellow horizontal line visualizes the uncertainty of the estimated parameter, *i.e.*, its 95% confidence interval. The width of the bins h for a single histogram is calculated according to the Freedman-Diaconis rule $h = \frac{2\mathrm{IQR}}{\sqrt[3]{N}}$, where IQR is the interquartile range of the sample and N is the size of the filtered sample (it includes the estimates obtained by those datasets that did not produce any outlier over all parameters), here $N = 428$.

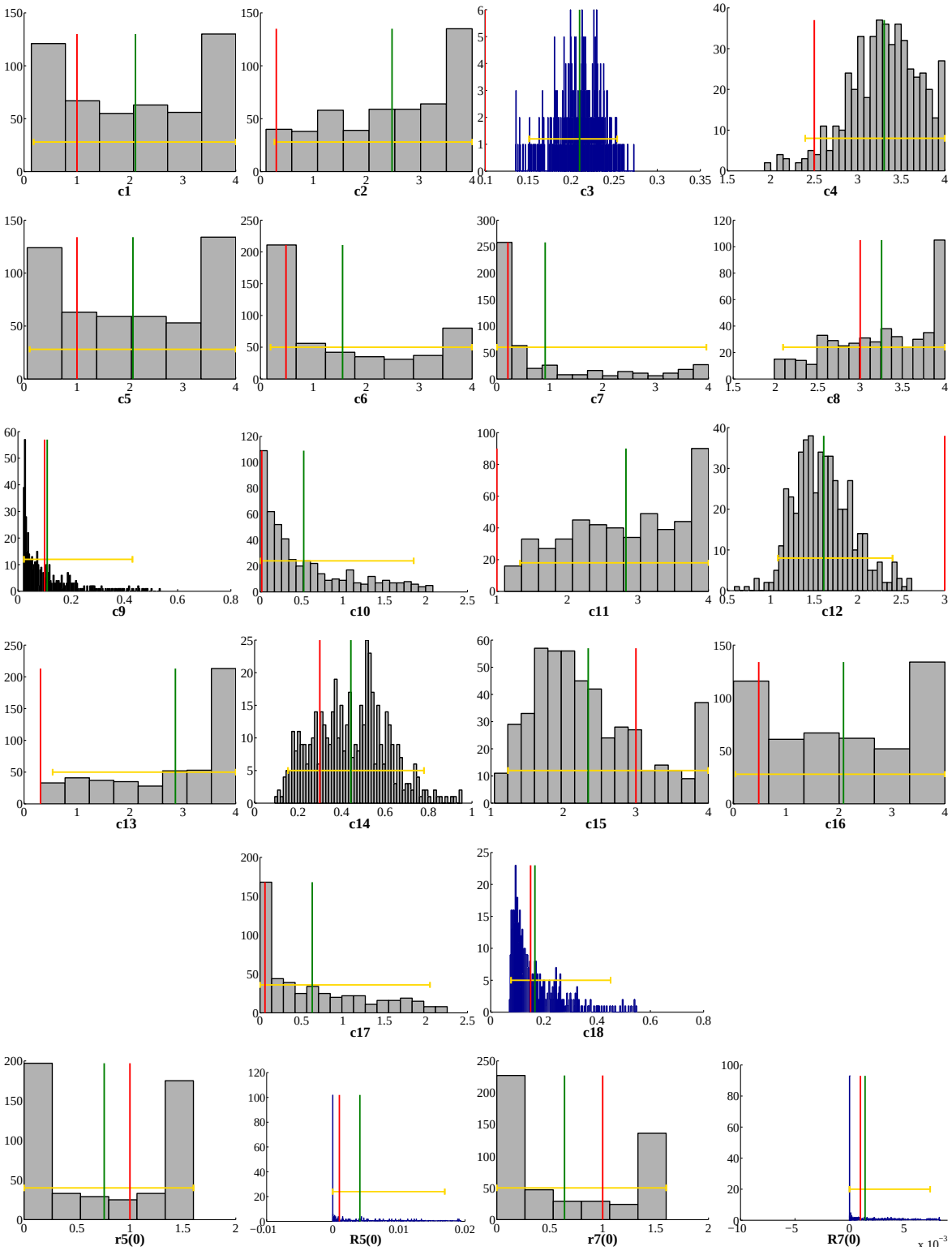


Figure 6.15: **Histograms of the parameter values generated with a Monte Carlo-based approach (estimated by DE using the AO observation scenario and artificial data with 20% relative noise).** The red vertical line represents the "true" value of the estimated parameter; the green vertical line represents the mean value of the sample μ; the yellow horizontal line visualizes the uncertainty of the estimated parameter, *i.e.*, its 95% confidence interval. The width of the bins h for a single histogram is calculated according to the Freedman-Diaconis rule $h = \frac{2\text{IQR}}{\sqrt[3]{N}}$, where IQR is the interquartile range of the sample and N is the size of the filtered sample (it includes the estimates obtained by those datasets that did not produce any outlier over all parameters), here $N = 492$.

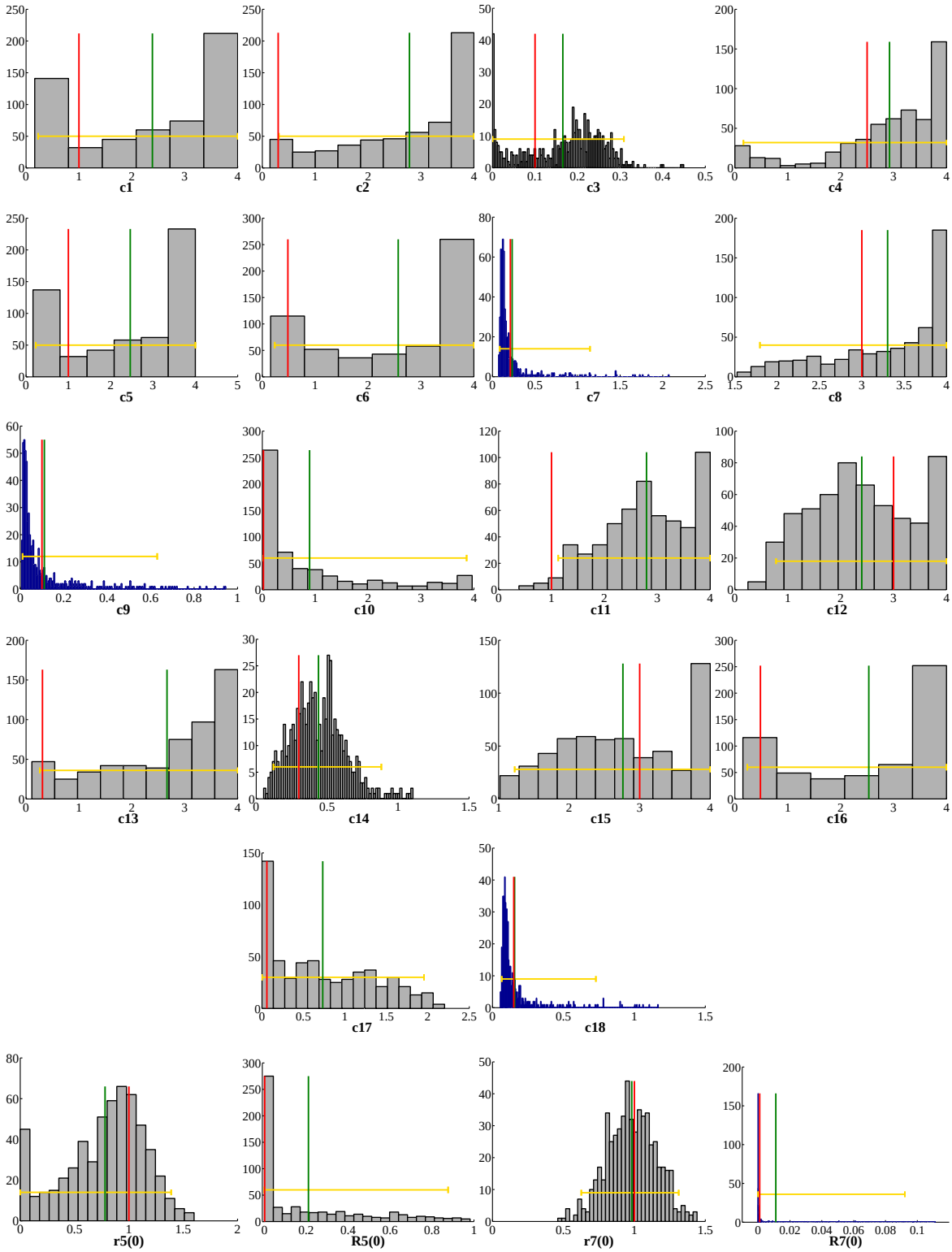


Figure 6.16: **Histograms of the parameter values generated with a Monte Carlo-based approach (estimated by DE using the TO observation scenario and artificial data with 20% relative noise).** The red vertical line represents the "true" value of the estimated parameter; the green vertical line represents the mean value of the sample μ; the yellow horizontal line visualizes the uncertainty of the estimated parameter, *i.e.*, 95% confidence interval. The width of the bins h for a single histogram is calculated according to the Freedman-Diaconis rule $h = \frac{2\text{IQR}}{\sqrt[3]{N}}$, where IQR is the interquartile range of the sample and N is the size of the filtered sample (it includes the estimates obtained by those datasets that did not produce any outlier over all parameters), here $N = 564$.

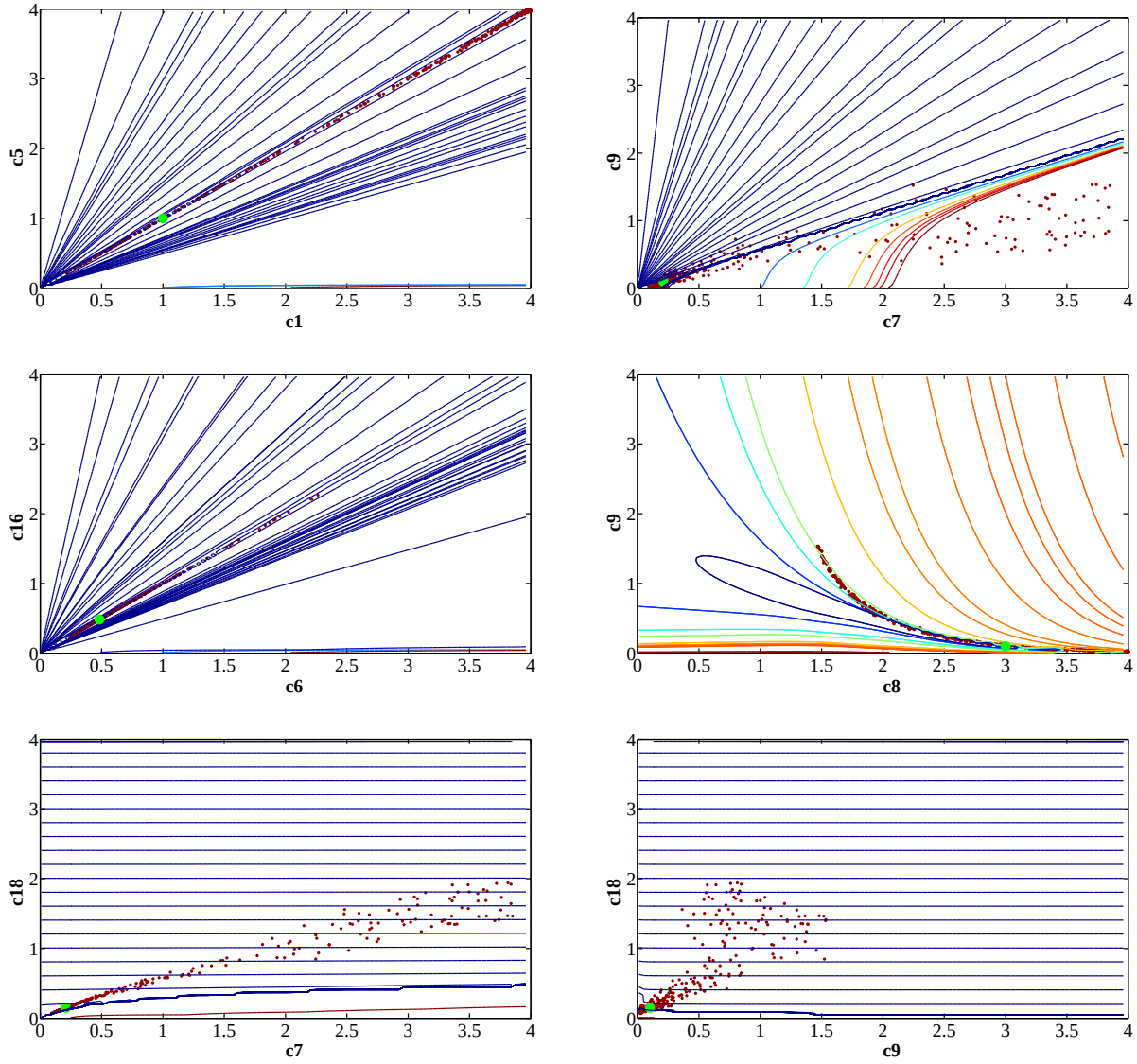


Figure 6.17: **Contour plots of the objective function with scatter plots of the parameters' estimates generated with a Monte Carlo-based approach (estimated by DE using the CO observation scenario and artificial data with 20% relative noise).** The plots correspond to the six pairs of the most correlated parameters. The green dot represents the "true" parameter solution (the one used for the simulation generating the artificial data). The red dots are the parameters' estimates obtained by DE from datasets generated in the Monte Carlo-based fashion.

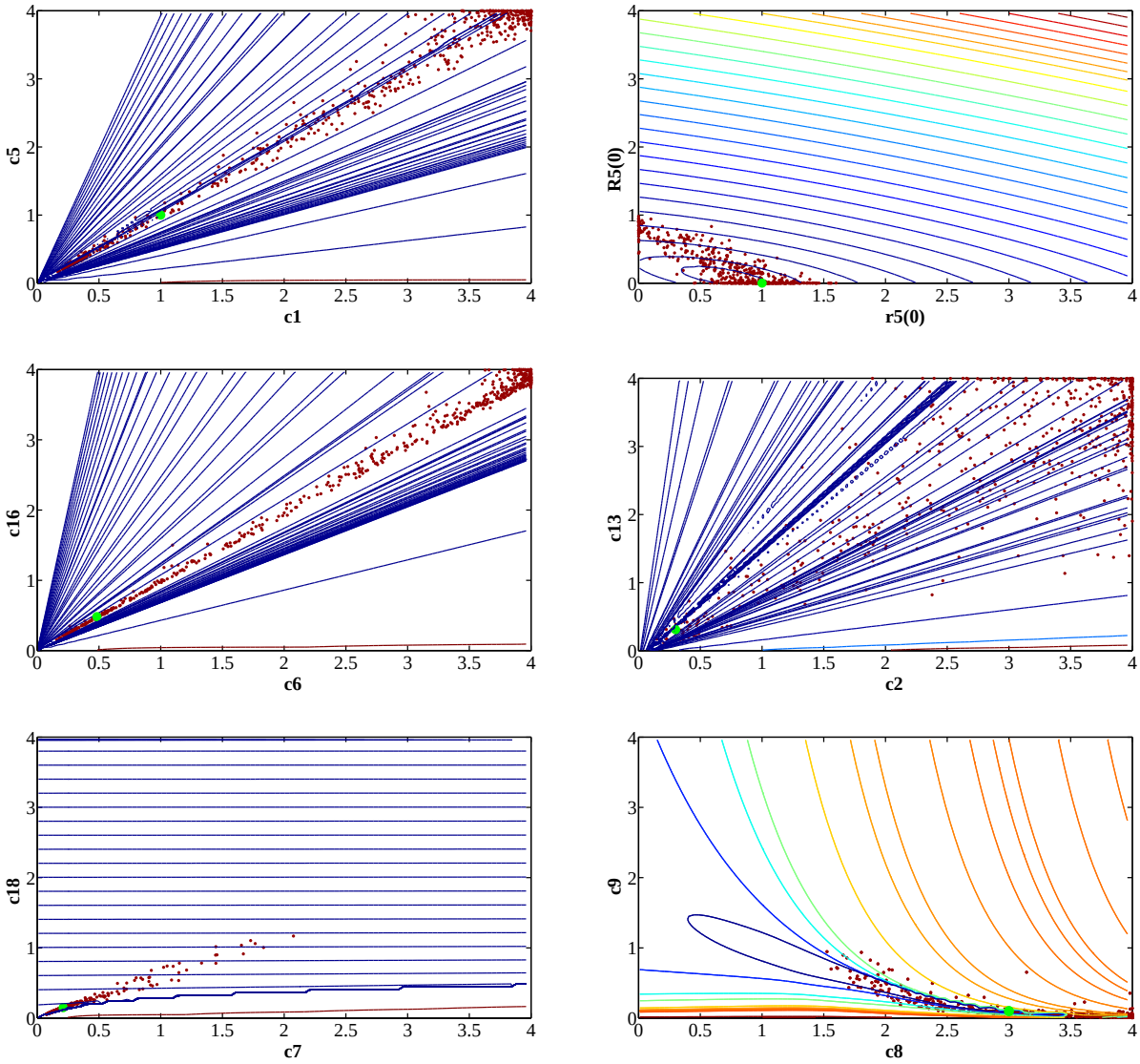


Figure 6.18: **Contour plots of the objective function with scatter plots of the parameters' estimates generated with a Monte Carlo-based approach (estimated by DE using the TO observation scenario and artificial data with 20% relative noise).** The plots correspond to the six pairs of the most correlated parameters. The green dot represents the "true" parameter solution (the one used for the simulation generating the artificial data). The red dots are the parameters' estimates obtained by DE from datasets generated in the Monte Carlo-based fashion.

Table 6.8: **Relative errors of the best parameter values estimated by DASA from artificial data.** The relative errors of the estimated parameters c are calculated with regard to the reference value c^* according to the formula $\frac{|c^*-c|}{c^*}$. Note that the error values are given in percent.

		CO			AO			TO			NPO		
c	c^*	$s=0\%$	$s=5\%$	$s=20\%$	$s=0\%$	$s=5\%$	$s=20\%$	$s=0\%$	$s=5\%$	$s=20\%$	$s=0\%$	$s=5\%$	$s=20\%$
c_1	1	300	216	17	220	289	177	60	250	31	300	300	300
c_2	0.3	435	1150	803	691	479	1150	482	1176	658	1173	1199	1137
c_3	0.1	922	3253	156	100	100	100	3900	3890	13	224	88	98
c_4	2.5	91	92	6	59	60	57	91	89	24	18	12	42
c_5	1	281	213	3	31	4	42	94	300	70	300	300	299
c_6	0.483	9	16	432	433	621	683	60	43	499	13	46	7
c_7	0.21	1798	711	1799	205	1466	1776	1805	1207	816	672	560	1780
c_8	3	31	49	26	10	9	19	30	36	13	18	17	6
c_9	0.1	3891	1544	605	3899	3900	3900	31	562	8	254	307	56
c_{10}	0.021	12828	896	7765	1472	7928	10212	5531	5635	1679	1527	9160	9784
c_{11}	1	73	152	292	119	153	104	49	150	37	52	228	83
c_{12}	3	98	58	82	98	99	96	33	34	29	26	51	5
c_{13}	0.31	307	1190	651	813	1094	1179	1190	997	1175	1018	1190	1190
c_{14}	0.3	548	485	171	181	156	156	172	563	51	58	298	255
c_{15}	3	3	28	33	33	33	33	31	65	18	33	24	22
c_{16}	0.483	396	13	474	102	683	353	619	56	728	13	46	7
c_{17}	0.06	759	1286	2886	795	28	296	4042	308	3158	1653	3728	3006
c_{18}	0.15	220	299	1275	413	811	2055	327	1178	556	579	481	1714
$r_5(0)$	1	2	0.4	29	59	60	49	100	67	97	6	3	13
$R_5(0)$	0.001	19	100	203	139349	156945	159590	92	100	402	96	51	100
$r_7(0)$	1	96	2	93	59	50	93	65	93	91	1	1	19
$R_7(0)$	0.001	85883	1292	36882	50110	23812	48395	80	521	985	54	95	100

Table 6.9: **Relative errors of the best parameter values estimated by PSO from artificial data.** The relative errors of the estimated parameters c are calculated with regard to the reference value c^* according to the formula $\frac{|c^*-c|}{c^*}$. Note that the error values are given in percent.

		CO			AO			TO			NPO		
c	c^*	$s=0\%$	$s=5\%$	$s=20\%$	$s=0\%$	$s=5\%$	$s=20\%$	$s=0\%$	$s=5\%$	$s=20\%$	$s=0\%$	$s=5\%$	$s=20\%$
c_1	1	221	83	300	100	100	300	54	57	75	25	22	78
c_2	0.3	272	1233	600	1115	68	260	1233	786	398	1233	873	271
c_3	0.1	90	168	168	97	3754	100	119	117	107	116	134	97
c_4	2.5	9	51	38	70	100	60	28	31	7	37	27	25
c_5	1	246	77	296	100	100	140	38	60	90	25	22	78
c_6	0.483	417	2	142	725	720	191	728	250	251	10	19	663
c_7	0.21	8	1564	1078	99	100	97	72	49	45	19	44	28
c_8	3	11	6	44	33	33	12	19	33	17	14	5	15
c_9	0.1	37	876	1084	1944	1363	1574	58	75	52	43	37	48
c_{10}	0.021	258	6843	970	104	100	100	404	3166	2252	250	195	207
c_{11}	1	140	92	127	300	100	278	269	291	226	118	66	120
c_{12}	3	42	95	80	29	33	21	57	45	49	43	37	43
c_{13}	0.31	182	1009	776	715	86	54	1139	796	1188	1111	1151	203
c_{14}	0.3	28	170	12	501	508	493	20	41	16	66	19	28
c_{15}	3	5	18	43	33	33	33	36	13	10	10	42	33
c_{16}	0.483	408	50	153	37	98	53	285	190	559	11	19	666
c_{17}	0.06	438	2848	94	475	99	894	17	1272	909	2243	100	593
c_{18}	0.15	7	810	593	93	95	89	37	36	21	17	41	28
$r_5(0)$	1	10	8	12	58	6	23	99	92	100	0.2	4	9
$R_5(0)$	0.001	11550	100	86	79314	123582	147319	333	59	99	47	3807	11839
$r_7(0)$	1	0.5	27	24	55	80	32	59	100	10	0.1	3	20
$R_7(0)$	0.001	57	25050	50	113333	157207	107106	100	100	100	73	51	66

Table 6.10: **Relative errors of the best parameter values estimated by DE from artificial data.** The relative errors of the estimated parameters c are calculated with regard to the reference value c^* according to the formula $\frac{|c^*-c|}{c^*}$. Note that the error values are given in percent.

		CO			AO			TO			NPO		
c	c^*	$s=0\%$	$s=5\%$	$s=20\%$	$s=0\%$	$s=5\%$	$s=20\%$	$s=0\%$	$s=5\%$	$s=20\%$	$s=0\%$	$s=5\%$	$s=20\%$
c_1	1	115	56	208	140	115	255	158	211	90	209	264	46
c_2	0.3	816	811	919	797	1082	774	763	562	570	740	1174	593
c_3	0.1	59	106	242	45	71	83	185	1246	132	1043	111	1161
c_4	2.5	4	37	4	35	19	42	42	78	15	81	1	83
c_5	1	124	46	173	61	110	56	51	16	143	210	264	46
c_6	0.483	256	292	501	649	398	590	237	440	441	520	553	534
c_7	0.21	1226	1297	1107	730	1585	895	1349	1251	1025	328	694	1163
c_8	3	55	44	28	6	8	7	49	53	42	44	42	49
c_9	0.1	1943	1432	585	3654	3642	3439	1395	1808	912	834	834	1406
c_{10}	0.021	12792	10133	14304	11686	17748	17352	10473	3563	14414	11948	17065	8622
c_{11}	1	188	227	136	179	200	188	157	165	156	186	161	190
c_{12}	3	3	50	15	30	23	29	3	26	0.4	3	13	25
c_{13}	0.31	757	854	592	551	541	875	1025	697	506	362	727	454
c_{14}	0.3	212	299	150	208	161	218	179	168	237	201	126	339
c_{15}	3	35	28	19	28	12	17	11	18	46	25	12	41
c_{16}	0.483	245	315	521	227	204	150	161	211	372	521	555	536
c_{17}	0.06	1709	2080	3018	42	11	429	3629	1985	844	908	3044	683
c_{18}	0.15	500	616	784	813	1298	1241	898	983	734	183	425	570
$r_5(0)$	1	3	52	56	91	65	48	26	14	98	7	32	9
$R_5(0)$	0.001	96663	23031	21794	21315	68210	79688	53029	45843	13858	37554	82844	3481
$r_7(0)$	1	84	55	57	53	70	23	26	44	74	8	29	34
$R_7(0)$	0.001	38304	16858	54415	46707	23580	5703	23531	50995	55932	18971	21861	24338

Table 6.11: **Relative errors of the best parameter values estimated by A717 from artificial data.** The relative errors of the estimated parameters c are calculated with regard to the reference value c^* according to the formula $\frac{|c^*-c|}{c^*}$. Note that the error values are given in percent.

		CO			AO			TO			NPO		
c	c^*	$s=0\%$	$s=5\%$	$s=20\%$	$s=0\%$	$s=5\%$	$s=20\%$	$s=0\%$	$s=5\%$	$s=20\%$	$s=0\%$	$s=5\%$	$s=20\%$
c_1	1	200	70	166	281	260	176	31	264	165	220	228	64
c_2	0.3	301	736	906	738	768	866	765	892	862	809	1226	758
c_3	0.1	1327	482	295	100	503	2284	1408	383	2566	334	165	77
c_4	2.5	98	93	84	38	66	99	75	6	86	66	43	33
c_5	1	136	42	284	6	87	63	82	296	256	196	237	54
c_6	0.483	136	296	29	705	397	525	645	676	164	456	403	188
c_7	0.21	1231	894	611	6	1253	1213	22	1597	1369	226	1112	633
c_8	3	18	20	44	80	29	9	40	20	13	6	15	46
c_9	0.1	1183	1865	2528	2815	2792	2588	574	741	10	2193	443	3026
c_{10}	0.021	13058	5510	5534	9537	12926	11291	11765	15748	10253	14053	16837	11591
c_{11}	1	6	168	155	221	155	209	199	75	77	53	272	138
c_{12}	3	12	59	1	36	23	40	18	11	8	15	75	6
c_{13}	0.31	163	958	750	963	435	1000	653	962	121	583	157	98
c_{14}	0.3	735	703	1069	171	31	19	464	784	55	744	263	665
c_{15}	3	82	13	71	33	10	2	11	80	22	22	75	21
c_{16}	0.483	170	473	500	453	126	69	67	677	130	469	599	150
c_{17}	0.06	872	373	692	1368	251	3186	376	1306	649	2784	3636	3325
c_{18}	0.15	1027	345	67	47	1743	2312	1367	1379	2060	290	848	232
$r_5(0)$	1	51	40	7	0.1	98	23	72	34	26	45	8	5
$R_5(0)$	0.001	18553	137320	131935	31923	57318	39264	63014	141434	29665	38180	113730	108441
$r_7(0)$	1	68	3	17	56	49	26	13	60	92	20	67	84
$R_7(0)$	0.001	58815	91328	90558	52936	154684	83854	84335	142195	67290	125217	136966	84406

Table 6.12: **Best estimated values of the model parameters from measured data.** Best parameter values as estimated by the four optimization methods (DASA, PSO, DE, and A717) from real-experimental data in the case of the TO and NPO observation scenario. Note that the initial values of the protein concentrations are scaled to simplify the comparison with the reference solution c^* in the artificial case.

		DASA		PSO		DE		A717	
c	c^*	TO	NPO	TO	NPO	TO	NPO	TO	NPO
c_1	1	0.0430	3.9990	0.7322	0.0169	0.0024	0.0124	2.3811	0.3542
c_2	0.3	3.9933	3.9998	2.2210	2.2310	3.9942	0.7024	2.4955	1.4670
c_3	0.1	2.0998	2.6714	1.3881	0.1878	0.0285	0.5391	3.6488	2.1391
c_4	2.5	3.8081	3.3911	3.7274	0.4415	1.4343	4.0000	1.8106	3.3485
c_5	1	0.1554	0.5537	3.7698	1.9905	0.2493	0.0908	3.4111	0.5914
c_6	0.483	0.0525	0.0717	0.0180	0.0425	0.0268	0.1561	0.1594	3.6601
c_7	0.21	3.8982	0.9565	1.3462	0.0848	0.1528	1.1262	1.1800	0.3755
c_8	3	2.4040	4.0000	1.4918	1.5146	3.9626	3.7725	0.6442	0.4293
c_9	0.1	0.4421	0.4149	1.6716	2.1492	0.0004	0.6144	3.9632	3.1055
c_{10}	0.021	2.4798	1.7236	2.8433	1.8045	0.0676	0.7280	3.0716	3.6792
c_{11}	1	0.0665	1.7365	2.7186	0.0003	0.4900	0.2363	2.9422	0.4156
c_{12}	3	0.0082	0.0149	3.7046	2.5726	3.9985	1.7607	3.6026	0.6001
c_{13}	0.31	0.0401	0.7091	3.7703	0.9808	1.7516	0.2506	1.2693	1.6567
c_{14}	0.3	2.9980	3.0230	2.8314	3.1975	1.1150	1.3669	3.5432	3.8190
c_{15}	3	3.5816	2.1751	2.1126	1.9293	3.8053	2.5061	1.5805	3.4274
c_{16}	0.483	0.3749	0.9534	0.0219	0.0216	0.0435	1.9111	3.0877	3.8228
c_{17}	0.06	0.0106	0.0184	0.0097	0.0075	0.0053	0.0137	1.7533	0.0078
c_{18}	0.15	0.6259	0.1223	3.0210	3.1291	0.4060	0.0895	0.0122	3.6839
$r_5(0)$	1	0.6917	1.5009	0.0231	0.9616	0.0100	0.0100	0.3196	1.8947
$R_5(0)$	0.001	0.6532	1.2222	0.9929	1.1451	1.2256	1.2863	1.3786	1.0128
$r_7(0)$	1	0.0212	0.0154	0.0688	0.2109	0.0611	0.0100	0.9175	1.7340
$R_7(0)$	0.001	0.0100	0.0835	0.0662	0.2533	0.0113	0.0515	0.3739	0.4843

Table 6.13: **Summary statistics for the parameter values generated with a Monte Carlo-based approach (estimated by DE using the CO observation scenario and artificial data with 20% relative noise)**. The column μ represents the mean values of estimated parameters; σ represents the standard deviation of the estimated parameters; $\frac{|c^*-\mu|}{c^*}$ is the relative error; c^{l} is the 2.5 percentile of the sample values, the lower bound of the 95% confidence interval; c^{m} is the 50 percentile of the sample values, the median; c^{u} is the 97.5 percentile of the sample values, the upper bound of the 95% confidence interval; $\mathrm{CI} = c^{\mathrm{u}} - c^{\mathrm{l}}$ is the length of the 95% confidence interval; $\frac{\mathrm{CI}}{\mu}$ is the confidence interval with respect to the mean; the last column represents the number of outliers. The complete sample size is 1000. For calculating the statistics we used reduced sample with size 428, without outliers.

c	c^*	μ	σ	$\frac{\lvert c^*-\mu\rvert}{c^*}$[%]	c^{l}	c^{m}	c^{u}	CI	$\frac{\mathrm{CI}}{\mu}$[%]	outliers
c_1	1	2.529	1.419	152.86	0.377	3.143	3.998	3.621	143.21	38
c_2	0.3	2.802	1.027	834.12	0.556	3.023	4.000	3.444	122.91	0
c_3	0.1	0.212	0.024	111.62	0.160	0.214	0.257	0.096	45.52	140
c_4	2.5	3.327	0.427	33.07	2.369	3.359	3.999	1.630	49.01	96
c_5	1	2.529	1.419	152.91	0.378	3.154	4.000	3.622	143.20	0
c_6	0.483	0.582	0.322	20.49	0.265	0.487	1.620	1.356	232.95	206
c_7	0.21	0.832	1.109	296.43	0.117	0.233	3.643	3.526	423.58	65
c_8	3	2.907	0.856	3.08	1.512	3.003	4.000	2.487	85.54	0
c_9	0.1	0.306	0.381	206.29	0.024	0.110	1.361	1.337	436.62	55
c_{10}	0.021	0.397	0.393	1788.55	0.032	0.247	1.497	1.465	369.39	98
c_{11}	1	2.680	0.795	167.99	1.363	2.683	3.998	2.636	98.36	0
c_{12}	3	1.630	0.388	45.66	0.922	1.616	2.491	1.569	96.24	40
c_{13}	0.31	3.002	1.009	868.38	0.571	3.312	4.000	3.429	114.23	0
c_{14}	0.3	0.476	0.162	58.60	0.165	0.495	0.764	0.599	125.90	52
c_{15}	3	2.386	0.713	20.45	1.250	2.298	3.991	2.741	114.86	0
c_{16}	0.483	0.582	0.322	20.49	0.264	0.488	1.626	1.362	233.99	194
c_{17}	0.06	0.746	0.612	1142.63	2.99E-04	0.638	1.951	1.951	261.67	4
c_{18}	0.15	0.443	0.515	195.34	0.087	0.168	1.776	1.689	381.30	26
$r_5(0)$	1	0.980	0.193	1.95	0.626	0.976	1.407	0.782	79.73	31
$R_5(0)$	0.001	0.013	0.019	1210.78	0.000	0.003	0.069	0.069	524.15	107
$r_7(0)$	1	0.992	0.140	0.79	0.695	0.993	1.262	0.567	57.16	55
$R_7(0)$	0.001	0.003	0.005	171.37	0.000	3.90E-04	0.018	0.018	665.52	161

Table 6.14: **Summary statistics for the parameter values generated with a Monte Carlo-based approach (estimated by DE using the AO observation scenario and artificial data with 20% relative noise)**. The column μ represents the mean values of estimated parameters; σ represents the standard deviation of the estimated parameters; $\frac{|c^*-\mu|}{c^*}$ is the relative error; c^{l} is the 2.5 percentile of the sample values, the lower bound of the 95% confidence interval; c^{m} is the 50 percentile of the sample values, the median; c^{u} is the 97.5 percentile of the sample values, the upper bound of the 95% confidence interval; $\mathrm{CI} = c^{\mathrm{u}} - c^{\mathrm{l}}$ is the length of the 95% confidence interval; $\frac{\mathrm{CI}}{\mu}$ is the confidence interval with respect to the mean; the last column represents the number of outliers. The complete sample size is 1000. For calculating the statistics we used reduced sample with size 492, without outliers.

c	c^*	μ	σ	$\frac{\lvert c^*-\mu\rvert}{c^*}$[%]	c^{l}	c^{m}	c^{u}	CI	$\frac{\mathrm{CI}}{\mu}$[%]	outliers
c_1	1	2.105	1.347	110.50	0.182	2.082	3.999	3.817	181.31	0
c_2	0.3	2.487	1.191	728.85	0.266	2.637	4.000	3.734	150.16	0
c_3	0.1	0.210	0.025	109.83	0.151	0.213	0.254	0.102	48.73	175
c_4	2.5	3.303	0.409	32.14	2.396	3.327	3.999	1.603	48.53	125
c_5	1	2.058	1.360	105.76	0.107	2.041	3.997	3.890	189.07	0
c_6	0.483	1.552	1.336	221.30	0.191	1.034	3.994	3.803	245.03	0
c_7	0.21	0.914	1.232	335.25	0.006	0.251	3.961	3.954	432.64	0
c_8	3	3.253	0.585	8.42	2.089	3.305	4.000	1.911	58.74	0
c_9	0.1	0.110	0.106	10.00	0.023	0.070	0.430	0.408	370.68	142
c_{10}	0.021	0.527	0.541	2407.80	0.010	0.307	1.852	1.842	349.79	89
c_{11}	1	2.830	0.844	183.05	1.330	2.887	4.000	2.670	94.32	0
c_{12}	3	1.608	0.334	46.39	1.083	1.584	2.400	1.317	81.89	43
c_{13}	0.31	2.859	1.170	822.36	0.540	3.301	4.000	3.460	121.02	0
c_{14}	0.3	0.443	0.168	47.58	0.154	0.442	0.779	0.625	141.23	14
c_{15}	3	2.342	0.755	21.93	1.236	2.170	3.997	2.761	117.89	0
c_{16}	0.483	2.084	1.361	331.56	0.044	2.015	3.999	3.955	189.72	0
c_{17}	0.06	0.629	0.640	948.57	0.000	0.396	2.048	2.048	325.52	9
c_{18}	0.15	0.166	0.095	10.88	0.077	0.132	0.451	0.374	224.91	142
$r_5(0)$	1	0.757	0.671	24.31	0.000	0.649	1.600	1.600	211.38	0
$R_5(0)$	0.001	0.004	0.005	310.55	0.000	0.002	0.017	0.017	412.72	72
$r_7(0)$	1	0.640	0.646	36.04	0.000	0.324	1.600	1.600	250.15	0
$R_7(0)$	0.001	0.001	0.002	43.10	0.000	5.32E-04	0.007	0.007	517.71	77

Table 6.15: **Summary statistics for the parameter values generated with a Monte Carlo-based approach (estimated by DE using the TO observation scenario and artificial data with 20% relative noise)**. The column μ represents the mean values of estimated parameters; σ represents the standard deviation of the estimated parameters; $\frac{|c^*-\mu|}{c^*}$ is the relative error; c^{l} is the 2.5 percentile of the sample values, the lower bound of the 95% confidence interval; c^{m} is the 50 percentile of the sample values, the median; c^{u} is the 97.5 percentile of the sample values, the upper bound of the 95% confidence interval; $\mathrm{CI} = c^{\mathrm{u}} - c^{\mathrm{l}}$ is the length of the 95% confidence interval; $\frac{\mathrm{CI}}{\mu}$ is the confidence interval with respect to the mean; the last column represents the number of outliers. The complete sample size is 1000. For calculating the statistics we used reduced sample with size 564, without outliers.

c	c^*	μ	σ	$\frac{\lvert c^*-\mu\rvert}{c^*}$[%]	c^{l}	c^{m}	c^{u}	CI	$\frac{\mathrm{CI}}{\mu}$[%]	outliers
c_1	1	2.388	1.386	138.82	0.230	2.785	3.999	3.770	157.86	0
c_2	0.3	2.781	1.185	827.14	0.312	3.168	4.000	3.688	132.59	0
c_3	0.1	0.165	0.094	65.29	2.06E-05	0.188	0.309	0.309	186.85	129
c_4	2.5	2.924	1.072	16.94	0.162	3.174	4.000	3.838	131.29	0
c_5	1	2.462	1.423	146.24	0.231	2.856	4.000	3.768	153.04	0
c_6	0.483	2.569	1.408	431.90	0.236	3.158	4.000	3.764	146.50	0
c_7	0.21	0.231	0.273	9.91	0.085	0.137	1.146	1.061	459.63	171
c_8	3	3.305	0.697	10.15	1.796	3.554	4.000	2.204	66.69	0
c_9	0.1	0.111	0.161	11.10	0.011	0.043	0.630	0.620	557.72	126
c_{10}	0.021	0.893	1.128	4154.40	0.017	0.352	3.862	3.845	430.39	44
c_{11}	1	2.796	0.839	179.56	1.127	2.801	4.000	2.872	102.75	0
c_{12}	3	2.399	0.975	20.03	0.779	2.333	3.998	3.219	134.16	0
c_{13}	0.31	2.664	1.169	759.42	0.259	3.023	3.999	3.740	140.38	0
c_{14}	0.3	0.439	0.191	46.36	0.122	0.420	0.882	0.760	173.13	66
c_{15}	3	2.763	0.873	7.92	1.228	2.712	4.000	2.772	100.33	0
c_{16}	0.483	2.532	1.384	424.25	0.236	3.179	3.997	3.761	148.53	0
c_{17}	0.06	0.733	0.614	1122.12	1.36E-04	0.626	1.955	1.955	266.57	0
c_{18}	0.15	0.156	0.160	3.97	0.065	0.103	0.729	0.664	425.79	118
$r_5(0)$	1	0.781	0.370	21.91	5.07E-04	0.856	1.389	1.389	177.85	0
$R_5(0)$	0.001	0.210	0.273	20916.77	0.000	0.052	0.877	0.877	417.20	9
$r_7(0)$	1	0.982	0.173	1.82	0.626	0.975	1.313	0.686	69.92	70
$R_7(0)$	0.001	0.011	0.024	1010.99	0.000	0.000	0.092	0.092	829.44	159

7 Modeling the dynamics of Lake Bled

In this chapter, we address the task of parameter estimation in a representative ecological model, *i.e.*, a population dynamics model for Lake of Bled, located in Slovenia. So far, Lake Bled has been modeled using simple theoretical models (Rismal et al., 1997), machine learning approaches (Kompare et al., 1997), and automated discovery of the structure and parameters of a model of its dynamics (Atanasova et al., 2006). The considered modeling approaches indicate that Lake Bled is a complex ecosystem requiring appropriate models for describing its behavior.

The model structure we consider was discovered in a previous study (Atanasova et al., 2006) with the automated modeling tool LAGRAMGE 2.0 (Todorovski and Džeroski, 2006) from measured data. The model includes three ordinary differential equations (ODEs) for three ecological variables, *i.e.*, dissolved phosphorus, total phytoplankton concentration, and the concentration of a zooplankton species *Daphnia hyalina*[1], that describe the dynamics of the food web in Lake Bled. The model was calibrated with a limited amount of measured data. Due to the ecosystem's complexity, the estimated parameters explain the calibration data well, but fail to provide accurate predictions for unseen data.

One of the reasons for the low quality of system dynamics reconstruction by the model at hand is the use of a local optimization method for parameter estimation. Namely, LAGRAMGE 2.0 uses a derivative-based local search method, *i.e.*, Algorithm 717 (Bunch et al., 1993). Moreover, LAGRAMGE 2.0 simulates the considered ODE models (during parameter estimation) using the so-called teacher forcing simulation. As already described in Chapter 2, teacher forcing simulation uses the measured values of the system variables at a given time point to calculate the system response at the next time point, unlike full numerical ODE integration, where only the initial values of the system variables are needed for the model simulation over longer periods of time.

In this context, we conduct an empirical comparison of four meta-heuristic optimization methods, and one local optimization method as a baseline, on a representative task of parameter estimation in a nonlinear dynamic model of an aquatic ecosystem, *i.e.*, Lake Bled. The five methods compared are the differential ant-stigmergy algorithm (DASA) and its continuous variant (CDASA), particle swarm optimization (PSO), differential evolution (DE) and Algorithm 717 (A717). We use artificial data, both without and with different levels of noise, as well as real measurements from Lake Bled. We also consider experiments with two different simulation approaches: teacher forcing (one-step supervised) and full (multistep) simulation. In addition, the preliminary findings from the single-model parameter estimation task are subjected to a follow-up study that investigates the influence of parameter estimation, e.g., using local (A717) vs. global (DASA) optimization, on the model selection step within the process of automated modeling of phytoplankton dynamics in Lake Bled.

The remainder of this chapter is organized as follows. Section 7.1 describes the conceptual and the specific mathematical formulation of the food web model for Lake Bled.

[1]We will use the term *D.hyalina* instead of *Daphnia hyalina* in the following.

Section 7.2 formulates the problem of parameter estimation, specifying the objective function, parameter value ranges and data used for the estimation of the model parameters. Section 7.3 outlines the specific experimental design related to the particular setup of the optimization methods. Section 7.4 presents and discusses the experimental results of parameter estimation in the specific (single structure) model. Section 7.5 is devoted to the follow-up study investigating the relation between parameter estimation and model selection, including problem statement, specific experimental setup, presentation and discussion of the experimental results. Finally, Section 7.6 summarizes the work in this chapter and outlines the conclusions of both experimental studies. The material presented in this chapter has been already published (or submitted for publication) in the form of journal paper and conference abstracts (Tashkova et al., 2011b, 2012b; Čerepnalkoski et al., 2011a,b).

7.1 Model

We address the task of parameter estimation in a practically relevant food web model describing the dynamics of three state variables, *i.e.*, phosphorus, phytoplankton and the zooplankton species *D. hyalina* in the aquatic ecosystem of Lake Bled. The specific model structure used here was obtained in a previous study. The model as a whole was induced from data about Lake Bled measured in the period from January 1996 to December 1996 using the automated modeling tool LAGRAMGE 2.0 (Atanasova et al., 2006).

Lake Bled is a typical subalpine lake of glacial-tectonic origin. It occupies an area of 1.4 km^2 with a maximum depth of 30.1 m and an average depth of 17.9 m The lake is naturally divided into two basins (eastern and western) with quite different characteristics and dynamics. The model described here refers to the upper layer of the eastern basin, *i.e.*, epilimnion, and no communication with the lower (hypolimnion) layer was taken into account. The epilimnion layer is 10 m deep and has a volume of $7 \cdot 10^6$ m^3. Further details concerning the model's assumptions are given by Atanasova et al. (2006).

Conceptually, the food web is modeled as follows. Phosphorus, the predominantly limiting nutrient for phytoplankton growth in this lake, enters the system with the inflow from three major rivers and leaves the lake with the outflow. In the lake, phosphorous is taken up by phytoplankton for its growth. Loss of phytoplankton biomass is due to the processes of respiration, sedimentation and grazing by the zooplankton species *D. hyalina*. *D. hyalina* biomass increases due to grazing on phytoplankton and decreases due to respiration and natural mortality. Through the respiration processes of phytoplankton and *D. hyalina*, inorganic phosphorus is released to the water column, entering again the food-web. The conceptual diagram of the food web model is presented in Figure 7.1.

The conceptual model from Figure 7.1 is mathematically formulated with three ODEs (Eq. (7.1)), one for each state variable, *i.e.*, soluble phosphorus (ps), phytoplankton (phyto), and *D. hyalina* (daph). Each equation is composed of several terms, representing a mathematical formulation of the ecological processes. The first equation captures the phosphorus dynamics: it includes the inflow and outflow of soluble phosphorus, phytoplankton respiration (*i.e.*, release of phosphorus due to phytoplankton respiration), *D. hyalina* respiration (*i.e.*, release of phosphorus due to *D. hyalina* respiration), and growth of phytoplankton (*i.e.*, consumption of phosphorus due to phytoplankton growth). The second equation, modeling the phytoplankton dynamics, includes contributions influences from four processes: phytoplankton growth, respiration and sedimentation, as well as grazing of *D. hyalina* on phytoplankton. Finally, the third equation aggregates

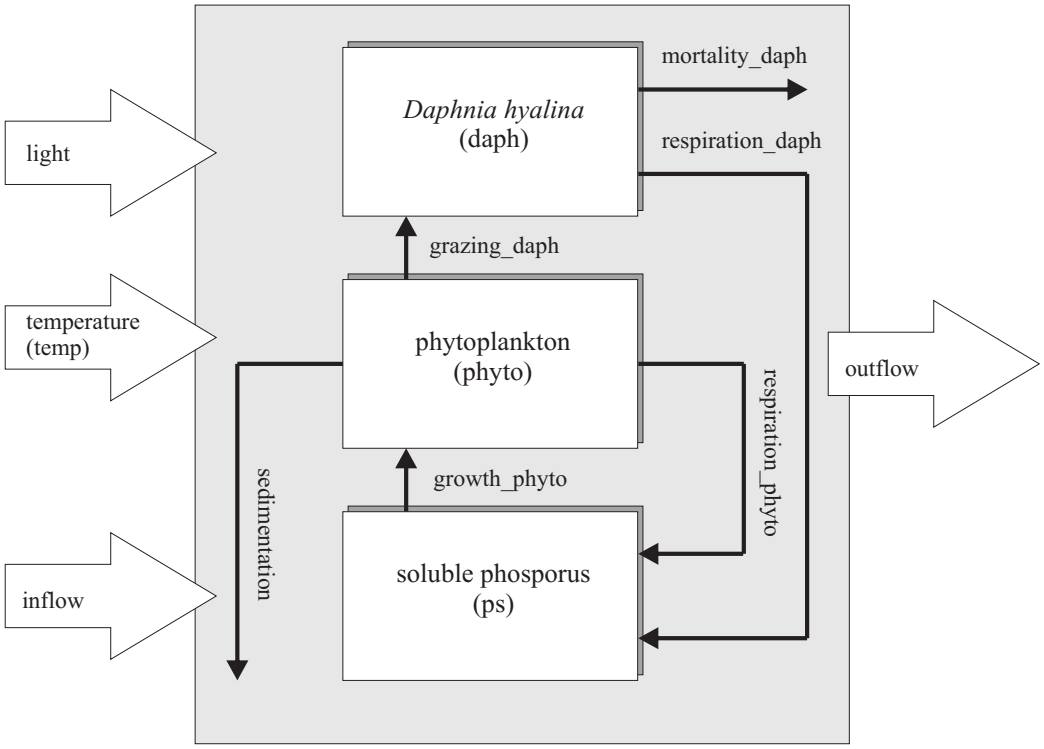


Figure 7.1: **A conceptual diagram of the Lake Bled model.** The system (endogenous) variables are represented with boxes, while the influence of the driving forces of the system (exogenous variables) are represented with thick white arrows. Finally, the thin black arrows represent the processes responsible for the system dynamics. Note that, the figure is adapted with modifications from Atanasova et al. (2006).

the influences of three processes on *D. hyalina*, *i.e.*, grazing (growth of *D. hyalina* due to grazing on phytoplankton), respiration and mortality of *D. hyalina*.

$$\begin{aligned}
\frac{d}{dt}\text{ps} &= \text{inflow} - \text{outflow} + c_1 \cdot \text{respiration_phyto} \\
&\quad + c_3 \cdot \text{respiration_daph} - c_5 \cdot \text{growth_phyto} \\
\frac{d}{dt}\text{phyto} &= \text{growth_phyto} - \text{respiration_phyto} \\
&\quad - \text{sedimentation} - \text{grazing_daph} \\
\frac{d}{dt}\text{daph} &= c_{12} \cdot \text{grazing_daph} - \text{respiration_daph} \\
&\quad - \text{mortality_daph}
\end{aligned} \tag{7.1}$$

Eqs. (7.2–7.9) present the mathematical formulations of the ecological processes. The variables in the model are listed and described in Table 7.1, while the model parameters and their descriptions are given in Table 7.2. Note that in the equations describing the ecological processes, H and V are constants that denote the depth and the volume of the modeled part of the lake (epilimnion of the eastern basin).

The processes of the inflow and the outflow of the nutrient (Eqs. (7.2) and (7.3)) into/from the lake are formulated simply by dividing the nutrient load from each contributing river, which equals to the product between the phosphorus concentration in the river and the flow rate of the river, and the volume of the modeled part of the lake.

Table 7.1: **Measured data about Lake Bled used for model induction.** The table lists the system (endogenous) variables together with the independent (exogenous) variables of the model, all of which are experimentally measured. The table includes information about the used notation, the variables' physical meaning, the measurement frequency and the corresponding units.

Variable name	Type	Description	Unit	Measurement frequency
q_k	Exogenous	Inflow to the lake, river Krivica	m^3/day	Daily
q_m	Exogenous	Inflow to the lake, torrent Mišca	m^3/day	Daily
q_r	Exogenous	Inflow to the lake, creek Radovna	m^3/day	Daily
q_j	Exogenous	Outflow (at surface), river Jezernica	m^3/day	Daily
q_n	Exogenous	Outflow (syphon)	m^3/day	Daily
ps_k, ps_m, ps_r	Exogenous	Soluble phosphorus concentration in the inflows	mg/l	Monthly
temp	Exogenous	Water temperature	°C	Monthly
light	Exogenous	Calculated underwater light	J/(cm^2·day)	Monthly
ps	Endogenous	Soluble phosphate concentration in the lake	mg/l	Monthly
phyto	Endogenous	Phytoplankton biomass concentration in the lake	mg/l *	Monthly
daph	Endogenous	Zooplankton (*D. hyalina*) concentration in the lake	mg/l *	Monthly

* mg of dry weight biomass per liter volumen

The phytoplankton respiration process (Eq. (7.4)) is formulated to have temperature influenced second order kinetics, while the process of *D. hyalina* respiration (Eq. (7.5)) has temperature influenced first order kinetics. The growth of phytoplankton (Eq. (7.6)) is formulated as an exponential function, limited by the temperature, light, and the nutrient concentration. Furthermore, the nutrient limitation is modeled by a Monod kinetics, light limitation by a photo-inhibition curve (Steele, 1965) and temperature influence by a linear response curve. Due to the process of sedimentation, removal of phytoplankton from the water column is modeled by first order temperature influenced kinetics (Eq. (7.7)).

The process (Eq. (7.8)) of grazing of *D. hyalina* (as filter feeders) is formulated using the filtration rate coefficient, a linear temperature response curve and an exponential term for food (phytoplankton) limitation on *D. hyalina* growth. The mortality of *D. hyalina* is the closure term in the model, represented by Eq. (7.9). It is modeled with a more complex expression, *i.e.*, a hyperbolic function, to account for predatory mortality as well.

$$\text{inflow} = \frac{\text{ps}_\text{k} \cdot \text{q}_\text{k} + \text{ps}_\text{m} \cdot \text{q}_\text{m} + \text{ps}_\text{r} \cdot \text{q}_\text{r}}{V} \tag{7.2}$$

$$\text{outflow} = \frac{\text{ps} \cdot \text{q}_\text{j} + \text{ps} \cdot \text{q}_\text{n}}{V} \tag{7.3}$$

$$\text{respiration\_phyto} = c_2 \cdot \text{phyto}^2 \cdot \frac{\text{temp} - c_{15}}{c_{16} - c_{15}} \tag{7.4}$$

$$\text{respiration\_daph} = \frac{c_4 \cdot \text{daph} \cdot \text{temp}}{c_{18}} \tag{7.5}$$

$$\text{growth\_phyto} = \text{phyto} \cdot \frac{c_6 \cdot \text{ps}}{\text{ps} + c_7} \cdot \frac{\text{temp}}{c_{17}} \cdot \frac{\text{light}}{c_{23}} \cdot e^{1 - \frac{\text{light}}{c_{23}}} \tag{7.6}$$

$$\text{sedimentation} = \frac{c_8 \cdot \text{phyto}}{H} \cdot \frac{\text{temp} - c_{19}}{c_{20} - c_{19}} \tag{7.7}$$

$$\text{grazing\_daph} = c_9 \cdot \text{daph} \cdot \frac{\text{temp} - c_{21}}{c_{22} - c_{21}} \cdot (1 - e^{-c_{10} \cdot \text{phyto}}) \cdot c_{11} \cdot \text{phyto} \tag{7.8}$$

$$\text{mortality\_daph} = \frac{c_{13} \cdot \text{daph}^2}{\text{daph} + c_{14}} \tag{7.9}$$

Table 7.2: **Model Parameters.** The table summarizes the used notation, physical meaning and ranges of values for the 23 model parameters. It also includes the reference set of parameter values used for the generation of artificial data.

ID	Name	Unit	Lower bound	Upper bound	Reference value
c_1	Phosphorous release during the respiration process of phytoplankton	–	0.001	1	0.0023
c_2	Phytoplankton respiration rate	1/day	0.009	0.15	0.072
c_3	Phosphorous release during the respiration process of *D. hyalina*	–	0.001	1	0.07
c_4	*D. hyalina* respiration rate	1/day	0.001	1.5	0.0026
c_5	Conversion factor between phytoplankton and phosphorus	–	0.001	1	0.0023
c_6	Phytoplankton maximal growth rate	1/day	0.05	3	0.21
c_7	Half-saturation constant for phosphorus	mg/l	0	5	0.00042
c_8	Sedimentation rate	m/day	0.1	0.9	0.5
c_9	*D. hyalina* filtration rate	l/(mg·day)	0.01	5	0.5
c_{10}	Half-saturation constant for phytoplankton	mg/l	0.001	3	0.58
c_{11}	Fraction of phytoplankton consumed by *D. hyalina*	–	0.001	1	0.56
c_{12}	*D. hyalina* assimilation coefficient	–	0.01	1	0.14
c_{13}	*D. hyalina* mortality rate	1/day	0.001	1.5	0.01
c_{14}	Coefficient in the hyperbolic term of the *D. hyalina* mortality process	–	0.0001	1.5	0.001
c_{15}	Minimal temperature in the phytoplankton respiration process	°C	2	4	3
c_{16}	Maximal temperature in the phytoplankton respiration process	°C	18	22	20
c_{17}	Reference temperature for the phytoplankton growth coefficient	°C	10	20	16
c_{18}	Reference temperature for the *D. hyalina* respiration rate	°C	10	20	16
c_{19}	Minimal temperature in the phytoplankton sedimentation process	°C	2	4	2
c_{20}	Maximal temperature in the phytoplankton sedimentation process	°C	18	22	20
c_{21}	Minimal temperature for the *D. hyalina* filtration rate	°C	2	4	3
c_{22}	Minimal temperature for the *D. hyalina* filtration rate	°C	18	22	20
c_{23}	Optimal light for phytoplankton growth	J/(cm^2·day)	150	300	170

7.2 Problem statement and data

In sum, the task of parameter estimation in the food web model for Lake Bled leads to a 23-dimensional continuous minimization problem with 23 dimensions corresponding to the model parameters. The objective function, *sum of squared errors* (SSE, as defined with Eq. (2.6)) between the observed and predicted values of the system output, is minimized with respect to the given data, subject to the structure of the ODEs of the food web model (described by Eq. (7.1) and the bound constraints on the values of the constant parameters listed in Table 7.2.

In order to evaluate the performance of different parameter optimization methods on this task, we conducted experiments with artificial data, obtained by simulating the food web model, and with real data from field measurements.

Measured (real-experimental) data. The data used for model formulation (Eq. (7.1)) were obtained from the Slovenian Environment Agency. The measurements include physical, chemical, and biological data from the year of 1996 taken for two water columns in the lake (Table 7.1). Collected data were pre-processed as follows:

1. All data were depth-averaged for the upper ten meters layer of the lake, *i.e.*, for the illuminated zone.

2. As evident from Table 7.1, the majority of the measurements were performed with a monthly frequency, which is generally too scarce for automated modeling tools. Therefore, we apply cubic spline interpolation between the measured points to obtain a convenient dataset of "daily" measurements for the induction of differential equations with LAGRAMGE 2.0 (Atanasova et al., 2006).

3. The concentrations of the biological data (phytoplankton and *D. hyalina*) are measured in number of individuals per volume unit [no_ind/ml]. Since the model is formulated using the mass balance principle, compatible measurement units, *i.e.*, [mass/volume] are necessary for all state variables. For the phytoplankton concentration, the transformation into [mass/volume] unit was done by the data providers, while for *D. hyalina* we used available information from the literature, *i.e.*, length-dry-weight regressions (Dumont et al., 1975). We estimated the average *D. hyalina* body length to be $L = 2$ mm and calculated the dry weight of a single individual of *D. hyalina* using Eq. (7.10) suggested by Dumont et al. (1975).

$$\text{dry\_weight } [\mu\text{g}] = 5.29L^{2.7} \tag{7.10}$$

The final dataset has 325 time points, *i.e.*, 325 data points for each model variable (for the 13 model variables as listed in Table 7.1), corresponding to 325 days in the period from January 1996 to December 1996. For validation of the model estimated from 1996 data, we used another dataset: it includes data from measurements performed in the period from March to December 1997. Further details about the measurements in Lake Bled are given by Atanasova et al. (2006).

Artificial (pseudo-experimental) data. We generated artificial data by full simulation of the ODE model from Eq. (7.1) at 325 equally spaced time points corresponding to 325 days in the period from January 1996 to December 1996 using the real-measured data for the exogenous variables and data corresponding to the first measurement in January 1996 (t_0) as initial values for the three system variables (*i.e.*, $\text{ps}(t_0) = 0.01650$ mg/l,

phyto(t_0) = 0.60547 mg/l, daph(t_0) = 0.28945 mg/l). To obtain more realistic artificial data, we added a normal Gaussian noise $N(0,1)$ to the noise-free simulated data for the system variables. Given the relative noise level s expressed as percentage, we calculate the noisy data as $Y_{\text{noisy}}(t) = Y(t) \cdot (1 + s \cdot N(0,1))$. In our experiments, we use three noise levels: 5%, 20% and 50%. Note that the noise-free data correspond to a noise level of 0%.

7.3 Experimental setup

Subject to five optimization methods, two model simulation approaches, and five datasets, that is, four artificial and one real, we performed in total 50 experiments. In addition to the experimental methodology described in Chapter 5, in this section we outline the specific setup of the optimization methods used to perform the experiments for parameter estimation of the ODE model represented by Eq. (7.1).

The values for the algorithms' parameters in this particular study were chosen based on the suggestions in existing literature, more precisely the settings proposed by the authors of the optimization methods. The parameter settings outlined below were used across all experimental scenarios corresponding to the two simulation approaches and all considered datasets.

DASA setup. The discretization base is set to 10, the maximum parameter precision is set to 10^{-15}, the number of ants is set to 10, the global scale increase factor to 0.02, the global scale decrease factor to 0.01, and the pheromone evaporation factor to 0.2.

CDASA setup. The reset threshold is set to 10^{-20}, the number of ants is set to 30, the global scale increase factor to 0.01, the global scale decrease factor to 0.02, and the pheromone evaporation factor to 0.2.

PSO setup. A variable random topology was chosen, the particle swarm size was set to 40, the neighborhood size to 3, the inertia weight to 0.721, and the acceleration coefficient to 1.193. In addition, default settings were used for the remaining parameters (advanced options not included in the standard PSO method).

DE setup. The chosen strategy was "DE/rand-to-best/1/exp", the population size was set to 200, the weight factor to 0.85, and the crossover factor to 1.0.

A717 setup. Since A717 is not a global search method, we wrapped the original procedure in a loop of restarts with randomly chosen initial points, providing in a way a simple global search. The procedure was restarted as many times as needed to achieve a comparable number of function evaluations to the other four methods. We used the module for bound constraint optimization with exact calculation of the derivatives.

7.4 Results and discussion

In the following subsections, we summarize and discuss the results from the experiments associated with the task of parameter estimation in the food web model of Lake Bled. The presented results are grouped into subsections by the type of data considered (artificial/real), where each subsection discusses the methods' performance and the quality of the models obtained by the five optimization methods (DASA, CASA, PSO, DE, and

A717) using the two simulation approaches (TF and FS). Note that the exact procedure for parameter estimation based on the TF approach involves a final simulation of the best model (defined with the best parameter values found by the optimization method) with full simulation: the FS dynamics is then used to calculate the model error reported as the final outcome of the estimation procedure. This procedure was originally used by Atanasova et al. (2006) to obtain the model structure considered here: for consistency, we adopt it in this work as well.

7.4.1 Parameter estimation from artificial data

RMSE performance. Figure 7.2 and Table 7.3 summarize the performance of the methods in terms of the RMSE metric for the artificial data with four levels of noise (0%, 5%, 20%, and 50%) and the two simulation approaches. While Figure 7.2 shows the RMSE of the models obtained by the optimization methods in all runs (25 executions) of the estimation procedure, Table 7.3 presents the RMSE values related to the best obtained models by the optimization methods in the considered experimental cases. The best obtained models are visually inspected in Section 7.4.1.

Table 7.3: **The RMSE values for the best models estimated from artificial data.** Five optimization methods (DASA, CDASA, PSO, DE, and A717) and two simulation approaches (TF and FS) are applied to artificial data with four levels of noise (0%, 5%, 20%, and 50%). The best values for each task, *i.e.*, combination of noise level and simulation approach are given in bold. The minimum (best) RMSE value corresponding to the simulations of the models with the parameter estimates obtained by a given optimization method is chosen over the 25 runs of that method.

Noise		DASA	CDASA	PSO	DE	A717
0%	TF	0.2751	0.0425	0.3950	**0.0096**	1.2933
	FS	0.1114	0.0998	0.3271	**0.0006**	0.8611
5%	TF	0.2779	0.1261	0.4570	**0.0840**	1.1043
	FS	0.1560	0.1327	0.3002	**0.0823**	0.9140
20%	TF	0.5680	0.4144	0.5714	**0.3975**	1.1011
	FS	0.3862	0.3840	0.5160	**0.3643**	0.8849
50%	TF	2.0765	2.5634	2.7377	2.1783	**2.0203**
	FS	1.3283	1.3243	1.3653	**1.3224**	1.4320

The eight panels in Figure 7.2 correspond to the two simulation approaches (in columns) and four artificial datasets (in rows), where each panel depicts a performance comparison of the five parameter estimation methods. The panels show that the median performance of A717 is significantly worse when compared to the four other meta-heuristic methods. The comparison among the latter indicate that the median RMSE performance of DE is significantly better than the performance of DASA, CASA and PSO. These findings hold for estimation based on both simulation approaches and at all noise levels. Moreover, DE has the lowest variance of the RMSE values across all data cases regardless of the simulation approach used (except the noise-free data case).

Considering the performance at different levels of noise, we observe a systematic decrease of the performance in terms of RMSE with the increase of the level of noise in the data. The noise affects the performance of all methods, but the magnitude of the effect differs. The performance of A717 is influenced less by the increasing noise than the best performing method DE. While we observe very large and remarkable differences among

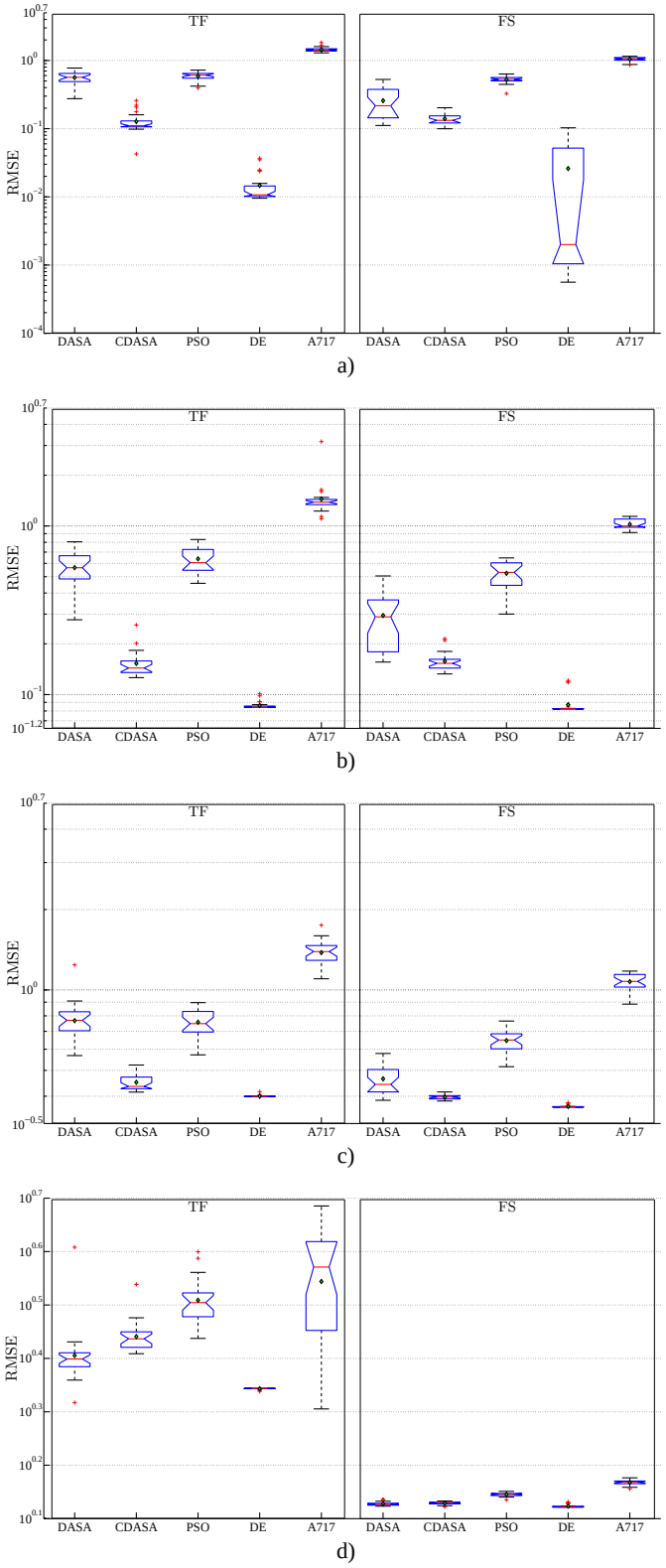


Figure 7.2: **RMSE performance of the models obtained by parameter estimation from artificial data.** Boxplots representation of the performance (in terms of the reconstructed output, RMSE) of the five optimization methods (DASA, CDASA, PSO, DE, and A717) using the two simulation approaches (columns TF and FS) and four artificial datasets (rows): a) noise-free, $s = 0\%$; b) noisy data, $s = 5\%$; c) noisy data, $s = 20\%$; and d) noisy data, $s = 50\%$. Due to the large differences in the order of magnitude, the RMSE values are plotted on a logarithmic scale. Note that for the same reason the y-axes of the plots corresponding to the different datasets are capturing different ranges of values: the upper bound is the same, but the lower bound is different.

the performance of the different meta-heuristic methods (*e.g.*, the performance of DE which is one or two orders of magnitude better than the other methods) in the noise-free case (or in the case of a small amount of noise, *i.e.*, $s = 5\%$), there is much less difference in their performance at high amounts of noise, *i.e.*, $s = 50\%$. Overall, the ranking of the optimization methods is very similar at all noise levels, regardless of the simulation approach used: DE is the best method; CDASA is ranked second; DASA is ranked third; PSO is ranked fourth and A717 is the worst performing method. At the noise level of 50%, the performance of DASA is better as compared to CDASA when using TF, while there is no difference in their performance when using FS.

The comparison of the two simulation approaches shows that the optimization methods perform better when using full simulation of the models. The difference in the median RMSE performance of the models found by the different methods is statistically significant in general (except for CDASA, PSO and DE on data with no or little noise). However, the difference in performance is smaller than the difference in the computational cost of both simulation approaches. Table 7.5 represents the average (over the 25 runs) CPU time of the parameter estimation procedure for all considered optimization methods and both simulation approaches. As evident, the difference is two orders of magnitude and, depending on the method/platform/data case, TF-based estimation is approximately 40–100 times faster than FS-based estimation.

Convergence. The convergence curves in Figure 7.3 further confirm that DE is the most suitable method for parameter estimation of the given food web model, at the given amount (half a million) of function evaluations. DE has faster convergence compared to the other meta-heuristics at all noise levels: this is in most cases clear after ten thousand evaluations. The difference is especially remarkable in the cases of no noise and a low level of noise in the data (see Figures 7.3.a) and 7.3.b)).

Regarding the remaining methods, at lower noise levels CDASA has better convergence than DASA. PSO, even though promising at the beginning of the search, seems to suffer from premature convergence (convergence to a sub-optimal solution). Finally, A717 seems to be trapped in local optima all the time. An interesting observation is that if the method is able to converge close to the optimum, then the estimation with both simulation approaches (TF and FS) achieves almost equal model errors, except in the case of a high noise level (see Figure 7.3.d)), where FS-based estimation (with all methods) leads to smaller errors.

The convergence rate of DE and the other methods is notably influenced by the noise level increase. Regardless of the simulation approach used, at the 50% noise level there is a small difference in the convergence rates of DASA, PSO, and DE, and it is clear that all methods (and not only A717) have extremely slow convergence. A717 is clearly showing the poorest convergence, which is only slightly affected by the different noise scenarios or simulation approaches.

An important remark is related to estimation with the TF approach. The average error performance calculated based on the TF model simulations (left-column graphs Figure 7.3) is not directly comparable to the average error performance of the models obtained by parameter estimation with FS model simulation (right-column graphs Figure 7.3): TF-based estimation, in this experimental setup, involves a final full simulation of the best model. In order to make a fair comparison of the influence of both simulation approaches on the estimation process at different time points over the given amount of time ("time" in terms of function evaluations), the middle-column graphs in Figure 7.3 depict the convergence of the average error of the full model simulations for the best models found by TF-based estimation across time. These graphs additionally confirm that as long as

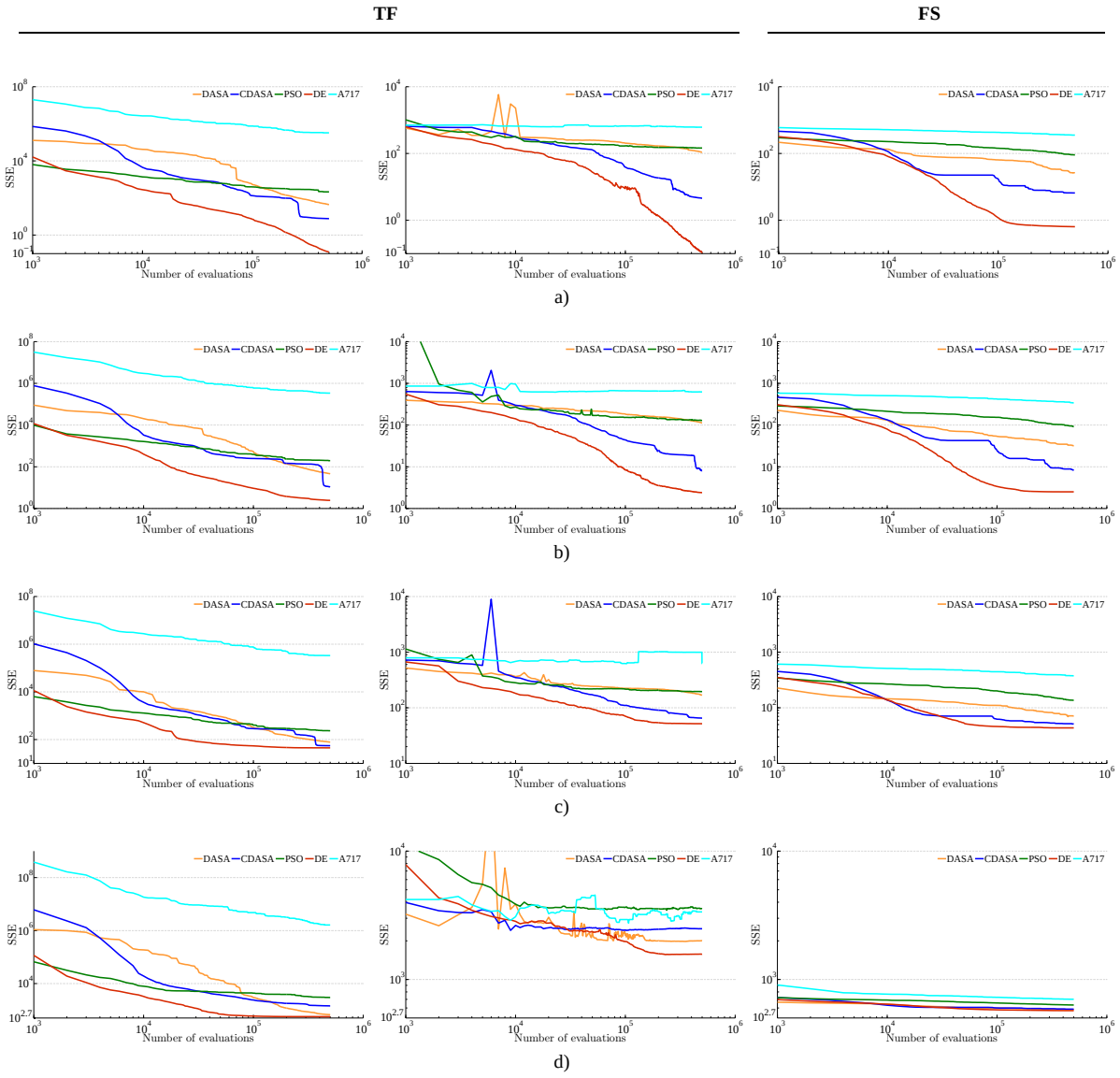


Figure 7.3: **Convergence of the optimization methods on the task of parameter estimation from artificial data.** Convergence curves for the five parameter estimation methods (DASA, CDASA, PSO, DE, and A717) using the two simulation approaches, TF (graphs in the left-hand side two columns) and FS (graphs in the right-hand side column) and four artificial datasets (rows): a) noise-free, $s = 0\%$; b) noisy data, $s = 5\%$; c) noisy data, $s = 20\%$; and d) noisy data, $s = 50\%$. The graphs in the first column from the left depict the average SSE performance of the best models estimated with TF, while the graphs in the middle depict the corresponding average SSE performance of the same best models simulated with FS. Each color represents one method: orange for DASA, blue for CDASA, green for PSO, red for DE, and cyan for A717. In order to capture the convergence trend over a wide range of values, the convergence curves are plotted using a logarithmic scale for both axes.

the noise is not high, *i.e.*, $s = 50\%$, estimation with the TF approach results in models of comparable quality. Note that, at a noise level of 50%, the average error performance of PSO is worse than the average error performance of A717.

Simulated dynamics of the best models. A common test of the quality of the obtained models includes visual inspection, *i.e.*, a graphical comparison of the observed outputs with the outputs predicted by the models. In this context, Figures 7.4–7.6 visualize the observed (black dotted line) *vs.* simulated dynamics (colored solid lines) of the three system variables (phosphorus, phytoplankton, and *D. hyalina*, respectively): this is done for the model corresponding to the best parameter values found by each of the five optimization methods (colors correspond to methods), using both simulation approaches (TF and FS) on the four artificial datasets. The graphs on the left-hand side correspond to the best models obtained by estimation with TF, while the graphs on the right-hand side correspond to the best models obtained by estimation with FS. The graphs in different rows correspond to models estimated from different datasets, presented from top to bottom in order of increasing level of noise in the dataset. Note that the RMSE of these models is given in Table 7.3.

A brief look at the three figures reveals that A717 fails to reconstruct the observed dynamics of all systems variables, regardless of the simulation approach, for all considered datasets. On the other hand, almost all meta-heuristic methods are able to capture the observed behavior in a satisfactory manner with a clear advantage of DE and CDASA over DASA and PSO. Regardless of the simulation approach, DE is the only method that almost perfectly fits the noise-free data for all three systems variables.

As long as the noise in the data is not extremely high, the meta-heuristic methods, especially DE and CDASA are able to avoid noise fitting and capture the actual trend in the observed dynamics of all system variables. While A717 seems to fail in all cases, all methods have problems at high levels of data noise (see the graphs in the last row of all three figures corresponding to the 50% data noise): meta-heuristics fail to reconstruct the model dynamics as well. Finally, note that as long as the noise in the data is not high, there is no clear difference between the models obtained by estimation based on TF and FS, as both seem to produce models of comparable quality.

7.4.2 Parameter estimation from measured data

Table 7.4 summarizes the results of parameter estimation in the food web model of Lake Bled using real data obtained from measurements preformed in 1996. The rows Best, Median, and Worst give the RMSE value corresponding to the best, median and worst solution (over the 25 runs) found by the five optimization methods using the two simulation approaches. The remaining two rows report the average RMSE performance (Average) and its standard deviation (Std). The two-panel Figure 7.7 visually summarizes these results.

Overall, the results on measured data confirm the findings of the experiments performed on artificial data. DE consistently leads to models with smallest RMSE (best performance), regardless of whether we consider the best, median, worst, or average RMSE (over the 25 runs): except for the best model obtained by TF-based estimation, which is ranked second (after the DASA best model). DE is again the most precise method with smallest RMSE variance. DASA and CDASA share the second place: DASA seems to be better when TF-based estimation is used and CDASA in the case of FS-based estimation. PSO is ranked third and A717 is ranked last: note that, in the case of TF-based estimation, PSO and A717 have similar performance on average.

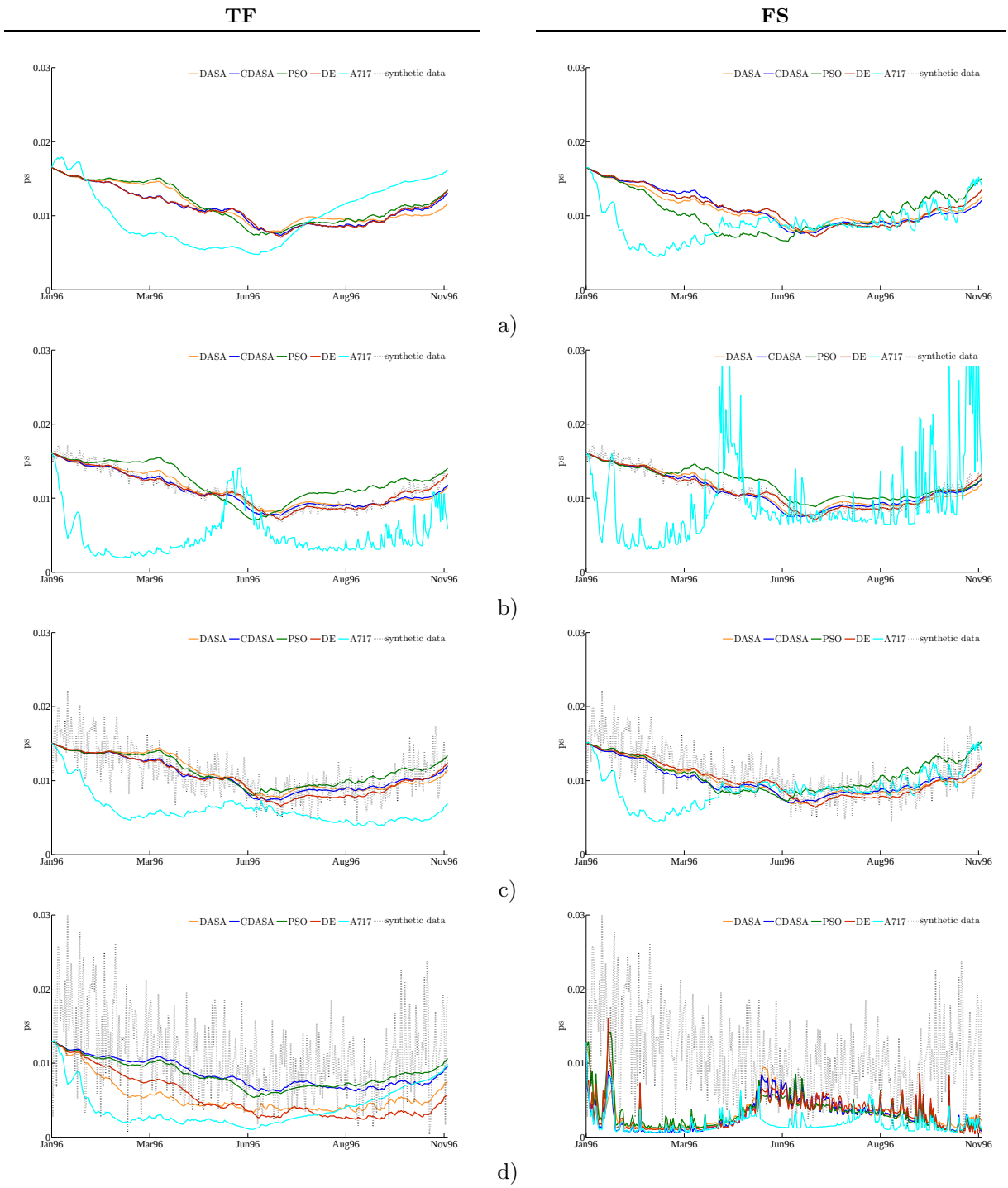


Figure 7.4: **Simulated dynamics of the best food web models obtained by parameter estimation from artificial data: phosphorus.** The graphs represent the observed behavior vs. that predicted by the best model in the case of parameter estimation from artificial: a) noise-free data, $s = 0\%$; b) noisy data, $s = 5\%$; c) noisy data, $s = 20\%$; and d) noisy data, $s = 50\%$. The left-hand side graphs depict predictions of the best models obtained by parameter estimation with TF, while the right-hand side graphs correspond to the predictions of the best models obtained by parameter estimation with FS. The observed behavior is represented by a black dotted line. The predicted behaviors are represented by solid lines with a different color for each optimization method: orange for DASA, blue for CDASA, green for PSO, red for DE, and cyan for A717. Note that, in the case of noise-free data, there is an almost perfect match between the red solid line and the black dotted line, therefore the last is not easy to discern.

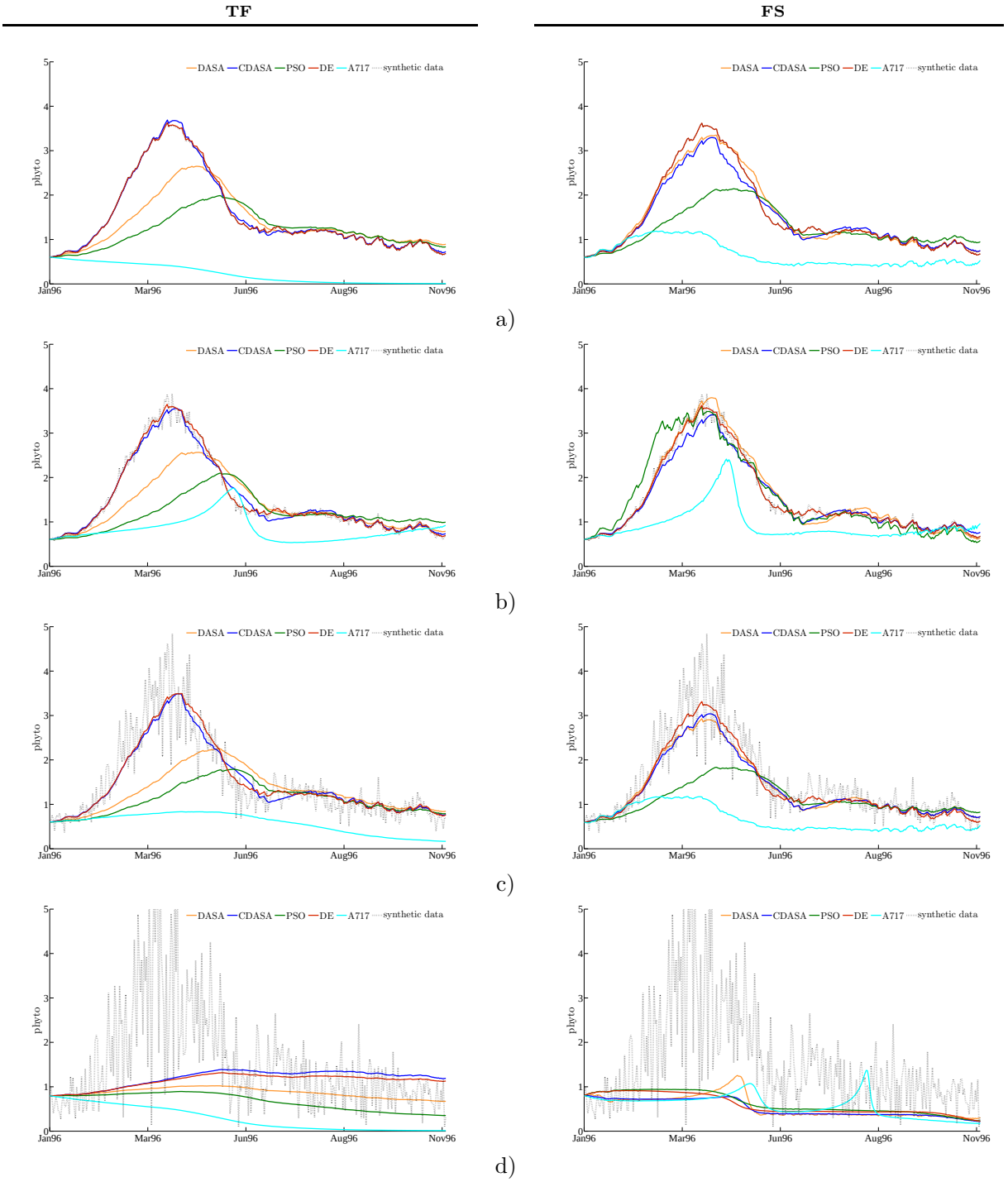


Figure 7.5: **Simulated dynamics of the best food web model obtained by parameter estimation from artificial data: phytoplankton.** The graphs represent the observed behavior vs. that predicted by the best model in the case of parameter estimation from artificial: a) noise-free data, $s = 0\%$; b) noisy data, $s = 5\%$; c) noisy data, $s = 20\%$; and d) noisy data, $s = 50\%$. The left-hand side graphs depict predictions of the best models obtained by parameter estimation with TF, while the right-hand side graphs correspond to the predictions of the best models obtained by parameter estimation with FS. The observed behavior is represented by a black dotted line. The predicted behaviors are represented by solid lines with a different color for each optimization method: orange for DASA, blue for CDASA, green for PSO, red for DE, and cyan for A717. Note that, in the case of noise-free data, there is an almost perfect match between the red solid line and the black dotted line, therefore the last is not easy to discern.

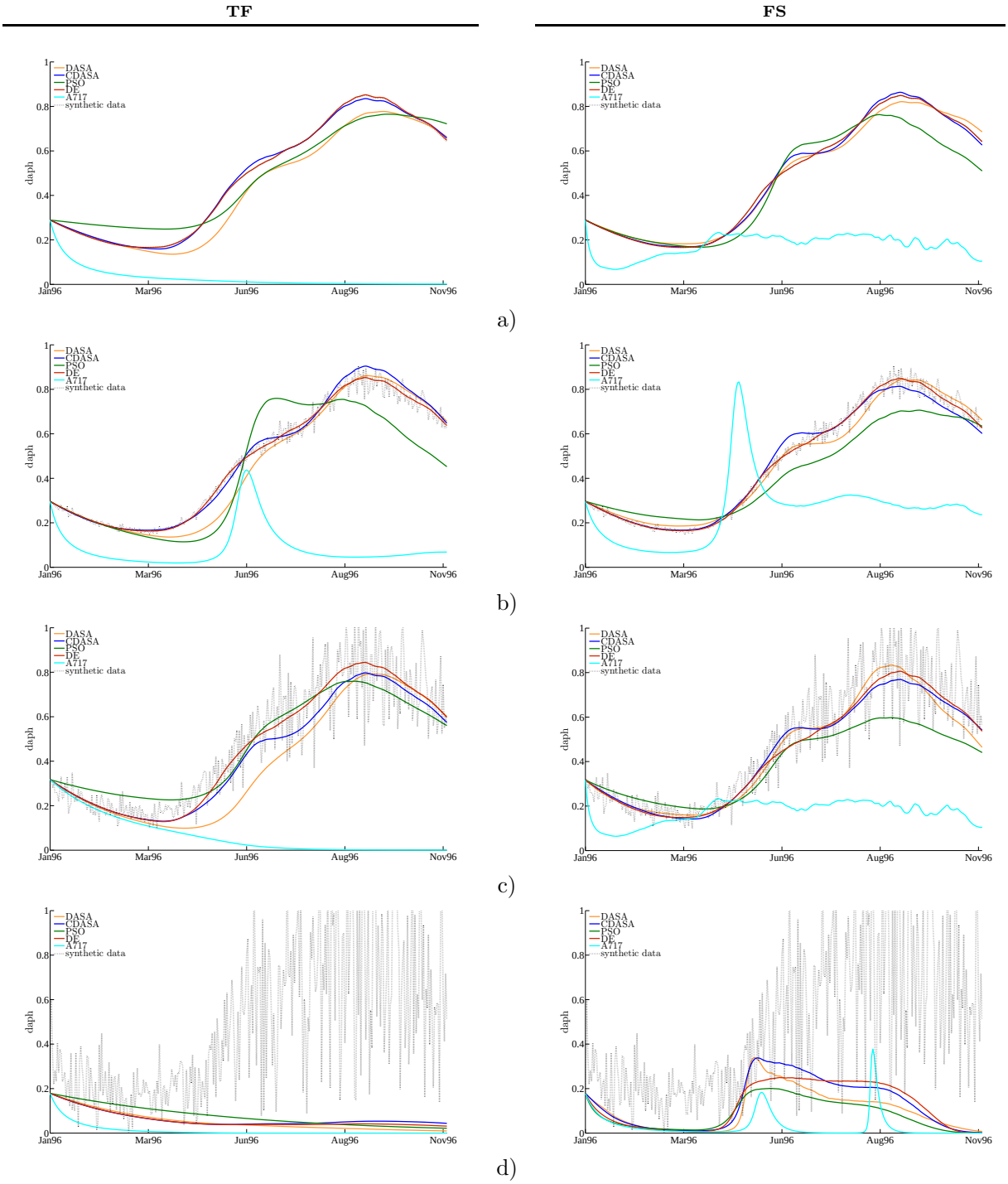


Figure 7.6: **Simulated dynamics of the best food web model obtained by parameter estimation from artificial data: *D. hyalina*.** The graphs represent the observed behavior vs. that predicted by the best model in the case of parameter estimation from artificial: a) noise-free data, $s = 0\%$; b) noisy data, $s = 5\%$; c) noisy data, $s = 20\%$; and d) noisy data, $s = 50\%$. The left-hand side graphs depict predictions of the best models obtained by parameter estimation with TF, while the right-hand side graphs correspond to the predictions of the best models obtained by parameter estimation with FS. The observed behavior is represented by a black dotted line. The predicted behaviors are represented by solid lines with a different color for each optimization method: orange for DASA, blue for CDASA, green for PSO, red for DE, and cyan for A717. Note that, in the case of noise-free data, there is an almost perfect match between the red solid line and the black dotted line, therefore the last is not easy to discern.

Table 7.4: **The RMSE values for the models estimated from measured data.** Five optimization methods (DASA, CDASA, PSO, DE, and A717) and two simulation approaches (TF and FS) are applied to measured data. The best values for each statistic are given in bold. The summary statistics for the RMSE values associated with the predictions of the models obtained by parameter estimation are calculated over 25 runs.

		DASA	CDASA	PSO	DE	A717
TF	Best	**0.7337**	0.8101	0.8450	0.7609	1.2106
	Median	0.9136	0.9258	1.2749	**0.7614**	1.4170
	Worst	1.1284	1.7869	1.6937	**0.7614**	2.5180
	Average	0.9012	1.0012	1.2463	**0.7614**	1.4948
	Std	0.1170	0.2436	0.2177	**0.0001**	0.2710
FS	Best	0.5300	0.5307	0.6246	**0.4773**	0.8684
	Median	0.5890	0.5467	0.6903	**0.4775**	1.0785
	Worst	0.6786	0.5964	0.8156	**0.5121**	1.1704
	Average	0.5910	0.5498	0.6886	**0.4834**	1.0609
	Std	0.0320	0.0154	0.0520	**0.0114**	0.0816

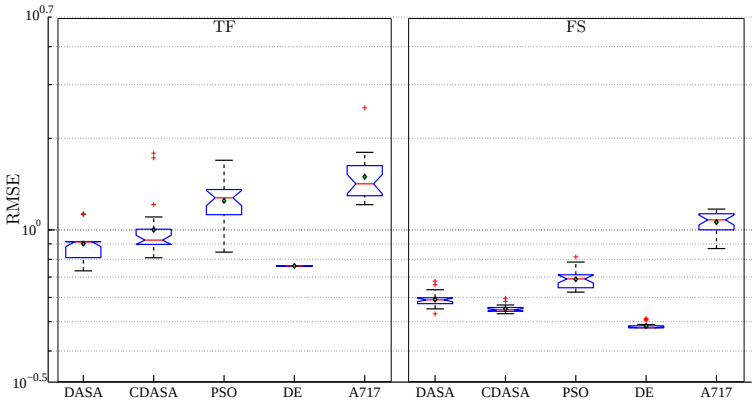


Figure 7.7: **RMSE performance of the models obtained by parameter estimation from measured data.** Boxplots representation of the performance (in terms of the reconstructed output, RMSE) of the five optimization methods (DASA, CDASA, PSO, DE, and A717) on the task of parameter estimation from measured data, using the two simulation approaches (columns TF and FS). Due to the large differences in the order of magnitude, the RMSE values are plotted on a logarithmic scale.

The boxplots clearly show the performance differences between the five methods. The boxplots further confirm that the median performance of the methods is significantly improved when FS-based estimation is used.

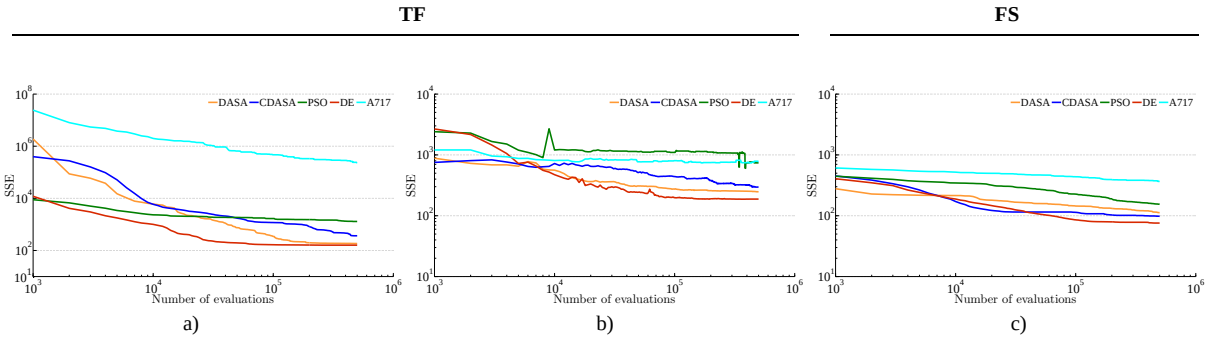


Figure 7.8: **Convergence of the optimization methods on the task of parameter estimation from measured data.** Convergence curves of the five parameter estimation methods (DASA, CDASA, PSO, DE, and A717) using the two simulation approaches: TF (a) and b)) and FS (c)). The two left-hand side graphs depict the convergence curves for each method corresponding to the parameter estimation task with TF: a) the average SSE performance of the best models simulated with the TF approach, and b) the corresponding average SSE performance of the best models simulated with the FS approach. Each color represents one method: orange for DASA, blue for CDASA, green for PSO, red for DE, and cyan for A717. In order to capture the convergence trend over a wide range of values the convergence curves are plotted using a logarithmic scale for both axes.

The convergence curves in Figure 7.8 resemble the ones for the artificial data. The DE method converges faster to better solutions than the other four methods and FS-based estimation leads to slightly better convergence than TF-based estimation. The graph depicting the convergence of the error associated with the full model simulation of the best models found by TF-based estimation (Figure 7.8.b)) shows that PSO performs (on average) worse than A717 on measured data. We observed a similar behavior in the case of parameter estimation from artificial data with a noise level of 50%.

Concerning the running time of the estimation procedures, Table 7.5 shows that TF-based estimation is significantly faster than FS-based estimation. This holds for artificial, as well as for measured data. In the case of estimation with TF, all optimization methods have roughly the same running times, except A717, which is almost three times slower. Note, however, that the experiments with A717 were performed on a different computer platform than the other methods.

A Holm test was conducted using the (best) values of the RMSE metric for the best models obtained by each method for all artificial and the measured datasets and both (TF and FS) simulation approaches. These RMSE values are given in Table 7.3 (artificial data) and Table 7.4 (measured data). The outcome of the statistical comparison test is summarized in Table 7.6. In terms of the RMSE of the best models, DE is the best ranked method that significantly outperforms the A717 and PSO method, at the 0.05 significance level, while the advantage of DE over DASA and CDASA is not statistically significant at the same significance level.

Simulated dynamics of the best models. We now discuss the quality of the models estimated from measured data, performing a visual inspection of the match between their predictions and the real measured data. The six graphs presented in Figure 7.9, visualize the observed (black dotted line) *vs.* simulated dynamics (colored solid lines) of the three

Table 7.5: **Time performance.** The average CPU time of one run (execution of half a million of objective function evaluations) is reported for all optimization methods across all simulation-data scenarios. The average time is given in minutes. Note that, the platform used to perform the experiments with DASA, CDASA, PSO, and DE was based on a six-core AMD Opteron 2.54 GHz processor 2427, 5 GB RAM with 64-bit Microsoft Windows Server 2008 R2 operating system, while A717 experiments were performed on a dual-core AMD Opteron 2.4 GHz processor 2216, 16 GB RAM with 64-bit Fedora Linux operating system.

Noise		DASA	CDASA	PSO	DE	A717
	TF	1.72	1.69	1.74	1.68	4.39
0%	FS	111.24	82.22	71.24	71.46	163.63
	TF	1.72	1.69	1.74	1.68	4.51
5%	FS	112.89	81.18	68.63	71.91	168.50
	TF	1.73	1.71	1.73	1.67	4.52
20%	FS	119.53	80.68	70.86	73.14	164.63
	TF	1.73	1.71	1.74	1.67	4.51
50%	FS	170.27	218.21	98.19	240.64	174.42
Real	TF	1.70	1.67	1.75	1.65	4.39
data	FS	137.01	78.25	87.88	78.26	162.84

Table 7.6: **Results of the Holm test for a significance level of** $\alpha = 0.05$**.** The table summarizes the outcome of the Holm test performed on the best RMSE values estimated by the five optimization methods (DASA, CDASA, PSO, DE, and A717) from artificial and real data, using both simulation approaches (TF and FS). DE is the reference method with rank $i = 0$. The hypothesis "there is no difference in performance between DE and the i-th ranked method" is rejected if the statement $p_i < \alpha/i$ holds.

i	α/i	Method	z_i	p_i	Hypothesis
4	0.0125	A717	4.667	0.000003	Rejected
3	0.0167	PSO	3.960	0.000075	Rejected
2	0.0250	DASA	1.838	0.065992	Not rejected
1	0.0500	CDASA	1.556	0.119795	Not rejected

system variables (phosphorus, phytoplankton, and *D. hyalina*): this is done for the model corresponding to the best parameter values found by each of the five optimization methods (colors correspond to methods), using both simulation approaches. The graphs on the left-hand side correspond to the best models obtained by estimation with TF, while the graphs on the right-hand side correspond to the best models obtained by estimation with FS. Every row corresponds to a different system variable.

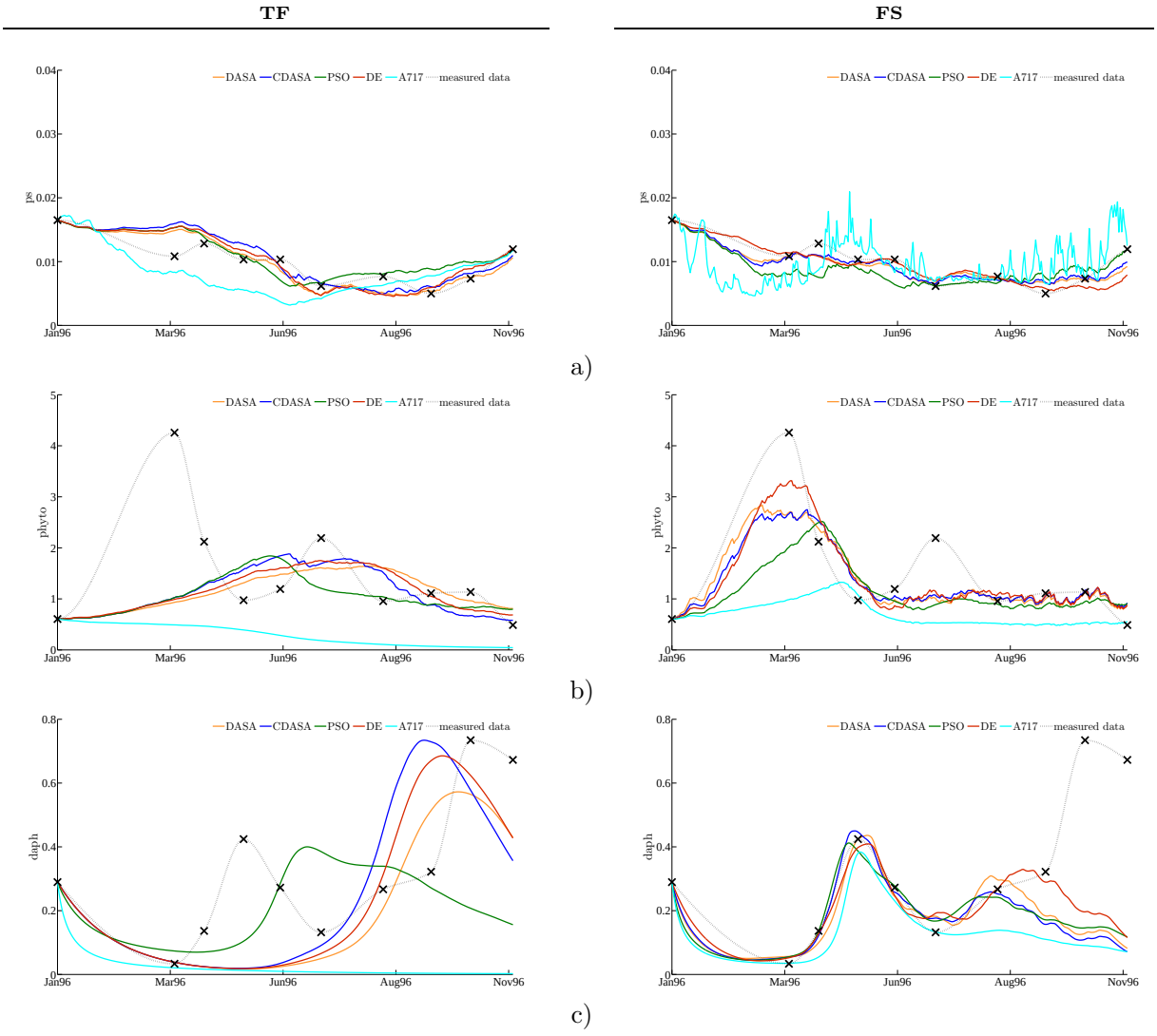


Figure 7.9: **Simulated dynamics of the best food web models obtained by parameter estimation from measured data.** The graphs represent the observed behavior *vs.* that predicted by the best food web model for: a) phosphorus; b) phytoplankton; and c) *D. hyalina*. The graphs on the left-hand side correspond to the best models obtained by estimation with TF, while the graphs on the right-hand side correspond to the best models obtained by estimation with FS. The observed behavior is represented by a black dotted line, obtained by interpolation of the real measurements marked with "×". The predicted behavior is represented by solid lines with a different color for each optimization method: orange for DASA, blue for CDASA, green for PSO, red for DE, and cyan for A717.

Overall, A717 leads to a model that exhibits the poorest performance when compared to the models produced by the other methods, except for modeling *D. hyalina* using FS-based estimation. The general impression is that all obtained models find it difficult to capture the observed dynamics of the system variables. While the actual trend in the

phosphorus dynamics is approximately captured by almost all models, this is not the case with the other two systems variables. Namely, the phytoplankton concentration has two seasonal peaks, in spring (higher) and in summer (lower), but no model fits them both at the same time. The models obtained by TF-based estimation with meta-heuristic methods are trying to capture the second peak, but are not successful. The model obtained by FS-based estimation with meta-heuristic methods (especially DE) are able to capture the spring peak, but underestimate its magnitude. Likewise, the *D. hyalina* concentration has two seasonal peaks, in spring (lower) and in autumn (higher), but again no model fits them both at the same time. While the models obtained by TF-based estimation with meta-heuristic methods (except PSO) approximately capture the autumn peak, the models obtained by FS-based estimation with all methods (especially DE) capture the spring peak quite well.

The results of parameter estimation with artificial data did not show a clear advantage of FS-based estimation over TF-based estimation. When using real measured data, the picture is clearer. According to the visual inspection of the model predictions, FS-based estimation resulted in better models.

Validation of the best models. The performance of the best models obtained on data measured in 1996, by the five optimization methods using the two simulation approaches, applied on data measured in 1997, is graphically shown in Figure 7.10. The figure has six graphs: the three on the left-hand side visualize the performance of the best models obtained by estimation with TF, while the three on the right-hand side visualize the performance of the best models obtained by estimation with FS. The graphs in each row depict the performance of the models for each of the three system variables. Each graph compares the performance of the best models obtained by each of the five optimization methods (represented by five differently colored lines) to the observed dynamics of the system variables (represented by a black dotted line).

The visual inspection clearly shows that there is a large discrepancy between the measured data and the predictions of the models for all system variables, with almost no correlation between the two dynamics. This is the case with all models, obtained by all the different methods of estimation using both simulation approaches.

If we analyze the measured data for 1997 and compare them with the measured data for 1996, we can see that the dynamics, even though similar at first sight, is quite different. There is an evident difference in the phytoplankton dynamics, as the spring peak is missing: there is a single peak that happens in early summer in 1997 (earlier than the summer peak in 1996), which appears after the spring peak in the *D. hyalina* dynamics. The dynamics of *D. hyalina* is different as well, with the spring peak twice higher than the autumn peak and the peaks at a clear distance, unlike the 1996 dynamics where the second peak is much higher and the peaks seem to be connected (with a smooth transition from the first to the second peak). These observations point at the complexity of the given ecosystem, which obviously requires a more complex model structure for its description. The phytoplankton population, for example, is composed of various groups, each being predominant at a different time. This leads to nutrient limitations not only by phosphorous, but also by nitrogen and silica, which are not taken into account by this model structure (Atanasova et al., 2006).

In short, the models of the dynamics of Lake Bled with the estimated parameter values are completely inappropriate for modeling the lake dynamics in 1997 and not completely appropriate for modeling the dynamics in 1996. Two possible reasons for this come to mind. The first reason is related to the difference in the dynamics of the lake between the

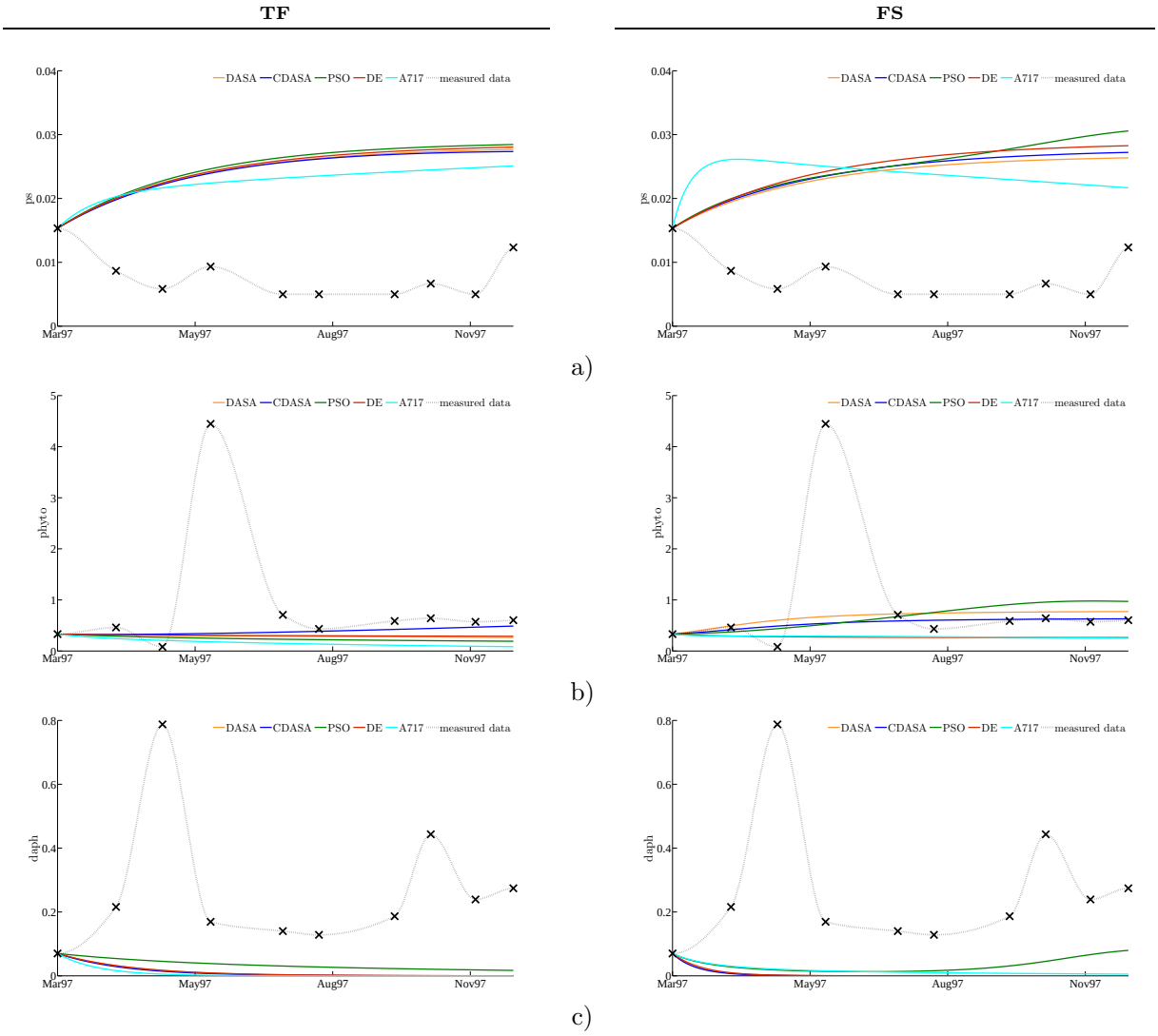


Figure 7.10: **Validation of the best food web model obtained by parameter estimation from data measured in 1996 on data measured in 1997.** The graphs represent the observed behavior *vs.* that predicted by the best food web model for: a) phosphorus; b) phytoplankton; and c) *D. hyalina*. The left-hand side graphs depict predictions of the best models obtained by parameter estimation with TF, while the right-hand side graphs correspond to the predictions of the best models obtained by parameter estimation with FS. The observed behavior is represented by a black dotted line, obtained by interpolation of the real measurements marked with "×". The predicted behavior is represented by solid lines with a different color for each optimization method: orange for DASA, blue for CDASA, green for PSO, red for DE, and cyan for A717.

two years as evident from the observed data and discussed above: due to these differences, a model (structure) appropriate for 1996 need not be appropriate for 1997.

The second reason concerns the appropriateness of the model structure, even for the case of modeling only the 1996 dynamics. Note that this model structure was induced based on data from one year (1996) only. Due to the sparsity and scarcity, these data were preprocessed before being used for induction of the model structure. The automated modeling system used to select this structure used the A717 method combined with TF to estimate parameters. The results of the present study clearly show that this approach has the poorest performance among the alternatives considered. This may have lead to the selection of an inappropriate model structure.

7.4.3 Parameter values and practical parameter identifiability

The output of the parameter estimation procedure is a set of parameter values that is found by the optimization method to reproduce the experimental behavior best, *i.e.*, with a smallest model error compared to all other investigated parameter solutions. If we have artificial data, generated by the simulation of the model with a set of specific parameter values, then it is easer to assess the performance of optimization methods, as we know the"true" set of parameter values. Related to this, Table 7.8 compares the reference parameter values given in Table 7.2 with the best parameter values obtained using each of the five optimization methods for parameter estimation based on both simulation approaches and artificial data at all noise levels. Since the parameters are defined on quite different scales, the comparison is presented in terms of relative error of the estimated parameters with respect to their reference values.

Despite the fact that DE finds parameter values that lead to low values of the objective function, especially from noise-free data or data with 5% noise, the obtained values can differ quite substantially from the reference values for some of the parameters. While some parameters, such as c_{16}, c_{20}, and c_{22}, are easy to estimate, regardless of the simulation approach and amount of considered noise, a few parameters, such as c_7 and c_{14}, are characterized with extremely high relative errors for almost all optimization methods. Overall, A717 seems to have problems with the estimation of almost a half of the model parameters. In the case of parameter estimation from real measured data, we do not have reference values, therefore it is difficult to draw firm conclusions, but it is clear that these values are far from the reference values. The best estimated parameter values obtained from real data are given in Table 7.9.

Evidently, many different sets of parameter values produce behaviors that resemble the reference model behavior, suggesting that the food web model for Lake Bled, as many others ecological models, has parameter identifiability problems. To empirically confirm this conjecture about our model, we performed a practical parameter identifiability test using the Monte Carlo-based approach: we used DE for parameter estimation, artificial data with 20% noise, and model simulation based on the FS approach.

The results of the test (see Table 7.10) confirm that the considered parameter estimation task has identifiability issues. The results reveal high relative errors of the estimated parameters, greater than 20% for 11 parameters. The relative error of the parameters c_4, c_8, c_{10}, c_{11}, c_{12}, c_{13}, and c_{19} go up to 90%, while in extreme cases the relative errors are over 175%, *i.e.*, for parameters c_3, c_7, c_9 and c_{14}.

Furthermore, the calculated uncertainties (95% confidence interval) of the parameters are large for almost all parameters: only four parameters (c_{16}, c_{20}, c_{22}, and c_{23}) have uncertainty (relative to the mean, see the results in the tenth column of Table 7.10) of approximately 20% of the mean estimate, which is the level of the artificially introduced

noise; six parameters have uncertainties up to 80% of the mean estimate; the remaining parameters have substantially larger uncertainties, which are especially large for the parameters c_7, c_9 and c_{14}.

If we take a look at the histograms of the distributions of the estimated parameter values (see Figure 7.15), we can see that the estimated values for a half of the model parameters are evenly distributed across the parameter ranges. More precisely, the parameters c_1, c_4, c_5, c_7, c_8, c_{11}, c_{16}, c_{17}, c_{20}, c_{21} and c_{22} have very similar, almost uniform, distributions with higher concentration of the estimates at the bounds of the prescribed range. For parameter c_4, the confidence interval does not include the reference value of the parameter, emphasizing the complexity of the optimization problem and the objective function. A closer look at the histograms reveals that some pairs of parameters (like c_1 and c_5) have very similar distributions of the estimates. Finally, we can notice that the distributions of the estimates of only a few parameters, like c_2, c_{15} and c_{23}, have Gaussian-like shapes, meaning that the optimization method is able to find the "true" solutions with high probability.

The correlation matrices for the estimated parameter values (presented in Figure 7.11.a)), re-confirm the practical identifiability problems, by emphasizing two pairs of correlated parameters: the (c_1,c_5) pair with correlation $R = 0.9234$, and the (c_6, c_{17}) pair with correlation $R = 0.8272$. The high pairwise correlations can be observed visually in Figures 7.11.b) and 7.11.c), where the scatter plots of the obtained solutions are combined with the contour plots of the objective function landscape for these two pairs of parameters. Both scatter plots show that the parameters estimates are in a linear relationship. But, the more interesting observation comes from the corresponding contour plots, indicating structural non-identifiability of c_1 and c_6 in the considered search interval. We observe very large flat regions in the considered part of the parameter space, where c_1 (for the second pair c_6) can take any value and does not influence the objective function: only c_5 (for the second pair c_{17}) has influence on the objective function.

7.5 Parameter estimation and model selection

Given the imperfect match of even the best models found in the experiments described above with the data used for calibration (1996) and poor fit to the data used for validation (1997), it is evident that the model structure selected (by LAGRAMGE 2.0, (Atanasova et al., 2006)) is not appropriate. However, it is clear that the use of global optimization methods with full simulation finds better parameter estimates as compared to a local optimization method with teacher forcing simulation. This brings to the front the important issue of the interplay between parameter estimation and model (structure) selection.

On one hand, one can expect that better parameter estimation would make it easier to select a more appropriate model structure. On the other hand, one might suspect that better parameter estimation may lead to over-fitting. It is not clear a-priori what the balance between the two would be in the context of automated modeling approaches, such as LAGRAMGE 2.0. Therefore, we conducted an extensive follow-up study of this issue (Čerepnalkoski et al., 2011a,b), *i.e.*, of the influence of parameter estimation methods on the final output of automated modeling, i.e., on what models are selected. The study involves modeling the dynamics of four aquatic ecosystems, one of them being Lake Bled, with the automated modeling tool ProBMoT (introduced in Chapter 2). Estimation of the numerical parameters in the considered model structures was performed by one local (A717) and one global (DASA) optimization method. A detailed report on the construction of the library of domain knowledge and the model candidates is beyond

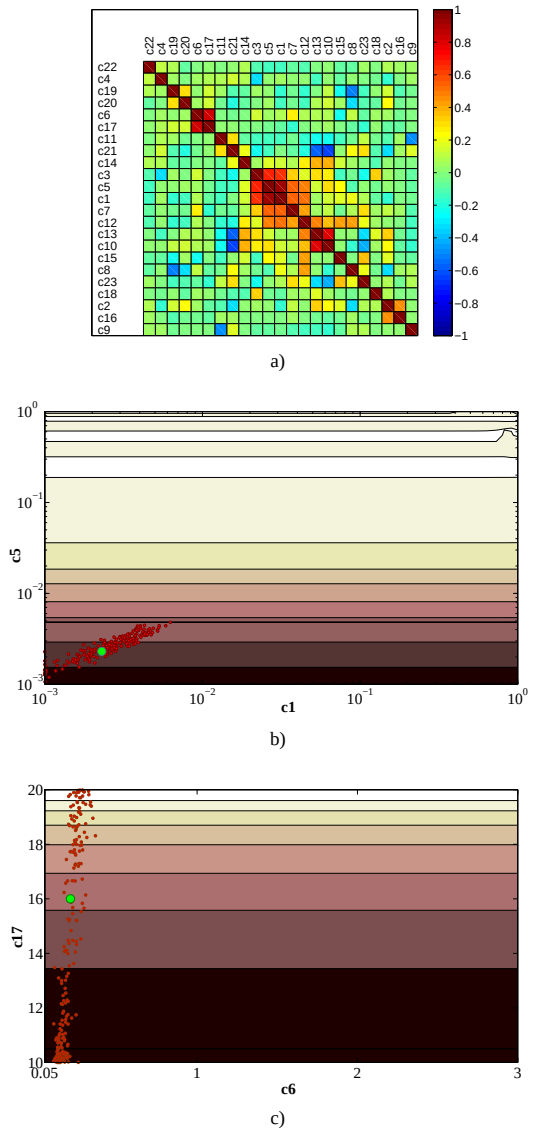


Figure 7.11: **Correlation of the model parameters generated with a Monte Carlo-based approach (estimated by DE using full simulation and artificial data with 20% relative noise).** a) Colored matrix cells visualize the correlation R for parameter pairs based on a scale-to-color mapping. The cells on the main diagonal represent the self-correlations of the parameters (equal to 1). The most correlated pairs of parameters are c_1/c_5 ($R = 0.9234$), and c_6/c_{17} ($R = 0.8272$). Smallest correlation ($R = 0.0011$) is obtained for the parameters c_4 and c_8. b) Contour plot of the objective function combined with scatter plot of the parameters' estimates corresponding to the most correlated pair of parameters, c_1 and c_5, plotted using a logarithmic scale for both axes. c) Contour plot of the objective function combined with scatter plot of the parameters' estimates corresponding to the second most correlated pair of parameters, c_6 and c_{17}. The green dots represent the reference parameter values given in Table 7.2. The red dots on the plots are the best parameters' estimates obtained by DE from datasets generated in the Monte Carlo fashion. Light-colored contours correspond to higher values, while dark-colored contours correspond to lower values of the objective function.

the scope of this dissertation: we focus mainly on the results concerning Lake Bled and the preliminary findings about the benefit of using global (instead of local) optimization within automated modeling.

7.5.1 Problem statement

We consider here the task of automated modeling phytoplankton dynamics using a general library of background knowledge on aquatic ecosystems, adequate conceptual model and real data for Lake Bled. The background knowledge and conceptual model were similar to the ones used by Atanasova et al. (2006) to derive the single model studied above. The data used were also the same as used by (Atanasova et al., 2006), *i.e.*, a superset of the data used to derive the single model.

The data, obtained from the Slovenian Environment Agency, were measured in the period from 1995 to 2002. They were pre-processed in the same way as described in the Section 7.2. In fact, the training (1996) and validation (1997) data used for the single-model parameter estimation task belong to this 1995-2002 data collection.

A conceptual model appropriate for modeling population dynamics in Lake Bled was designed in the format specified by Čerepnalkoski et al. (2011a). This model is quite similar to the one presented in Figure 7.1: it includes entities, such as nutrients, phytoplankton and zooplankton, for which we have measurements and conceptual top-level processes appropriate for modeling phytoplankton dynamics, *i.e.*, growth, respiration, sedimentation, and grazing by zooplankton. But, unlike the conceptual model in Figure 7.1, it focuses solely on modeling the primary producer dynamics, *i.e.*, phytoplankton dynamics, and does not explicitly model the dynamics of the zooplankton and the nutrients. While phosphorus was the only nutrient considered in the food web model in Figure 7.1, the influence of two additional nutrients, nitrogen and silica, was taken into account in modeling the phytoplankton dynamics in the current conceptual model.

The general background knowledge for modeling the specific class of aquatic ecosystems, *i.e.*, lake food webs, is formulated in a library of domain knowledge (Čerepnalkoski et al., 2011a). The library is capable of expressing relations between nutrients, phytoplankton, zooplankton and detritus. These relations can take on different forms of limited or unlimited phytoplankton growth, zooplankton grazing on phytoplankton with a number of formulations of the grazing rate, respiration of phytoplankton and zooplankton, as well as sedimentation. Subject to the requirements for the particular modeling task in Lake Bled and previous analysis/knowledge with respect to modeling population dynamics in Lake Bled, the library was tailored for Lake Bled by including/omitting processes and specific process alternatives which are needed for completing the conceptual model of the system.

For the specific modeling task, starting with a conceptual model and the library of domain knowledge, ProBMoT performed exhaustive search in the space of possible structures, resulting in 27216 candidate models structures, that is, candidate models for Lake Bled phytoplankton dynamics.

The available collection of data was divided into separate datasets for each year, as the goal was to construct a separate model for each year. The obtained model is an explanatory model of the system for the given time range. For each model structure and each year we have one parameter estimation task.

Finally, the parameters of the generated models were estimated by minimization of the sum of squared error (SSE, as defined with Eq. (2.7)) between the trajectories of phytoplankton in the measurements and in the simulation of the model as the objective function. The RMSE is used as a quality measure of each model. The model which has

the lowest RMSE value is the model that fits the data best. Note that, since the modeling was constrained to a single population dynamics, *i.e.*, phytoplankton dynamics, we used the ordinary least-squares estimation instead of the weighted least-squares estimation as was the case with the parameter estimation of the single model structure addressed in the previous section.

7.5.2 Experimental setup

The resulting experimental setup consisted of eight tasks of modeling phytoplankton dynamics, where a task is defined by the combination of the given conceptual model and a (one year) dataset. The parameter settings outlined below were used in all considered modeling tasks. Due to the reduced food web dynamics being modeled (only phytoplankton) and the large space of candidate models (27216) the number of function evaluations was limited to 10000 evolutions. The computational time was roughly proportional to the size of the parameter estimation problem, and the parameter estimation task was only restarted two times (as opposed to the 25 runs in the single-structure case). Furthermore, the evaluation of all discovered ODE models was based on full simulation.

The following settings were used for the optimization methods DASA and A717.

DASA setup. The discretization base is set to 10, the maximum parameter precision is set to 10^{-15}, the number of ants is set to 10, the global scale increase factor to 0.02, the global scale decrease factor to 0.01, and the pheromone evaporation factor to 0.2.

A717 setup. Since A717 is not a global search method, we wrapped the original procedure in a loop of restarts with randomly chosen initial points, providing in a way a simple global search. The procedure was restarted as many times as needed to achieve approximately 10000 evolutions. We used the module for bound constraint optimization with exact calculation of the derivatives.

7.5.3 Results and discussion

This section presents and discuses the results from the experiments carried to investigate how the choice of parameter estimation method (local-A717 vs. global-DASA) influences the process of automated modeling, in particular the subprocess of model selection. Since the primary goal of automated modeling is to discover the most suitable model for a given system, we compare the most suitable models found by ProBMoT with A717 and with DASA used for parameter estimation. The quality of each model is assessed by its RMSE value.

Table 7.7 shows the RMSE values for each of the best models found with ProBMoT/A717 and ProBMoT/DASA on all datasets[1]. The models selected by ProBMoT/DASA always have a lower RMSE than those selected by ProBMoT/A717. From this quantitative comparison, we can conclude that DASA outperforms A717 on each modeling task.

Figure 7.12 shows the simulated phytoplankton dynamics provided by the best models found with A717 and DASA on the different datasets. There is a considerable difference in the errors of the best models found in the cases corresponding to data measured in 1997, 2000 and 2002. This quantitative difference in the RMSE values is directly visible as a qualitative difference in the simulations. The A717 model simulation exhibits some

[1]We will use A717 and DASA for ProBMoT/A717 and ProBMoT/DASA in the following.

Table 7.7: **The errors of the best models of phytoplankton dynamics in Lake Bled obtained by automated modeling from yearly measured data.** The table gives the values for the RMSE of the best models discovered by ProBMoT using A717 and DASA for parameter estimation, from data measured in each of eight consecutive years.

	Year							
Method	1995	1996	1997	1998	1999	2000	2001	2002
A717	0.650	0.362	0.800	0.802	1.282	2.439	0.683	0.771
DASA	0.376	0.206	0.157	0.443	1.205	0.821	0.455	0.240

resemblance to the measured data, whereas in most cases the DASA model simulations follows the dynamics of phytoplankton concentration very closely, both in terms of time of rapid phytoplankton growth and peak amplitude. However, in some cases, *e.g.*, for the data measured in 1999, both models capture the overall trend but overestimate the peak amplitudes. Overall, the DASA model represents a considerable improvement over A717.

Previous experience shows that the first-ranked model is not necessarily the most suitable model. Many models may have RMSE values marginally different from each other, yet produce qualitatively different behavior. RMSE is the most commonly used error measure for evaluating models against time-series of measurements, because it captures the overall degree of fitness of the simulation. However, a domain expert would often deem as more suitable a model with a slightly higher RMSE if it provides a better explanation of the system dynamics.

Therefore, we do not limit the comparison (between A717 and DASA) to the single best model. Instead, we are interested in the comparison of the N_{bm} best models ($N_{\text{bm}} > 1$) found by using each parameter estimation method. The most comprehensive setting is when N_{bm} is equal to the total number of candidate models. In that case, all candidate models are sorted in order of increasing error. We then compare the resulting profile curves of model errors. Figure 7.13 shows the error profiles for both A717 and DASA on the eight automated modeling tasks. The profile curves clearly show that DASA manages to find better models throughout the space of model structures. In other words, not only is the best model found with DASA better than the best one found with A717, but also the second best model found with DASA is better than the second best model found with A717, the third best model found with DASA is better than the third best model found with A717, and so forth. The only exceptions are the few models at the very tail of the ranked list of models derived from data measured in 1999. In this case, both A717 and DASA find poor parameter values, but, the models with parameter values found with DASA have larger RMSE than those with parameters found with A717.

The best model found with DASA does not have to have the same structure as the best one found with A717. In general, the model structures in the A717 and DASA rankings are not listed in the same order. The last and most stringent comparison that we perform is by ordering the models fitted with DASA not by their own errors, but according to the errors that the model structure has when fitted with A717: the model structures found by A717 are refitted by DASA and thus, the RMSEs achieved by A717 and DASA with the same model structure are compared. In this way, the models fitted with DASA are ordered in the same way as those with A717. We then compute the difference $diff(i) = \text{RMSE}_{\text{A717}}(i) - \text{RMSE}_{\text{DASA}}(i)$, where $\text{RMSE}_{\text{A717}}(i)$ is the RMSE of the i-th model fitted with A717, whereas $\text{RMSE}_{\text{DASA}}(i)$ is the RMSE of the i-th model fitted with DASA. A positive value of *diff* means that DASA performs better (has a

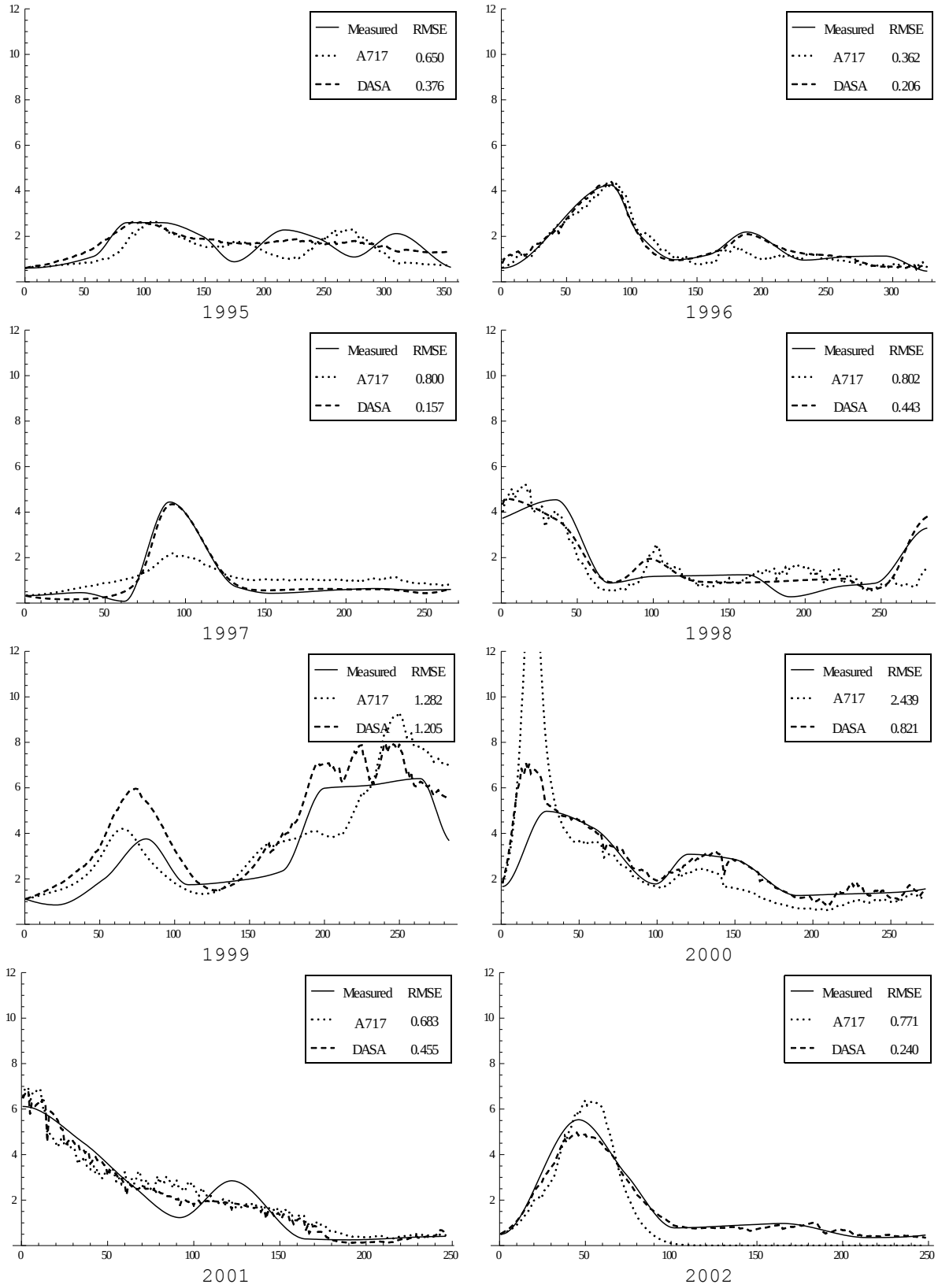


Figure 7.12: **Simulated dynamics of the best models obtained by automated modeling from measured data.** The eight graphs represent the observed behavior *vs.* the behavior predicted by the best phytoplankton model obtained by ProBMoT on a single-year automated modeling task for eight consecutive years. The observed behavior is represented by a black solid line. The phytoplankton dynamics predicted by ProBMoT/A717 is represented by a black dotted line, while the phytoplankton dynamics predicted by ProBMoT/DASA is represented by a black dashed line.

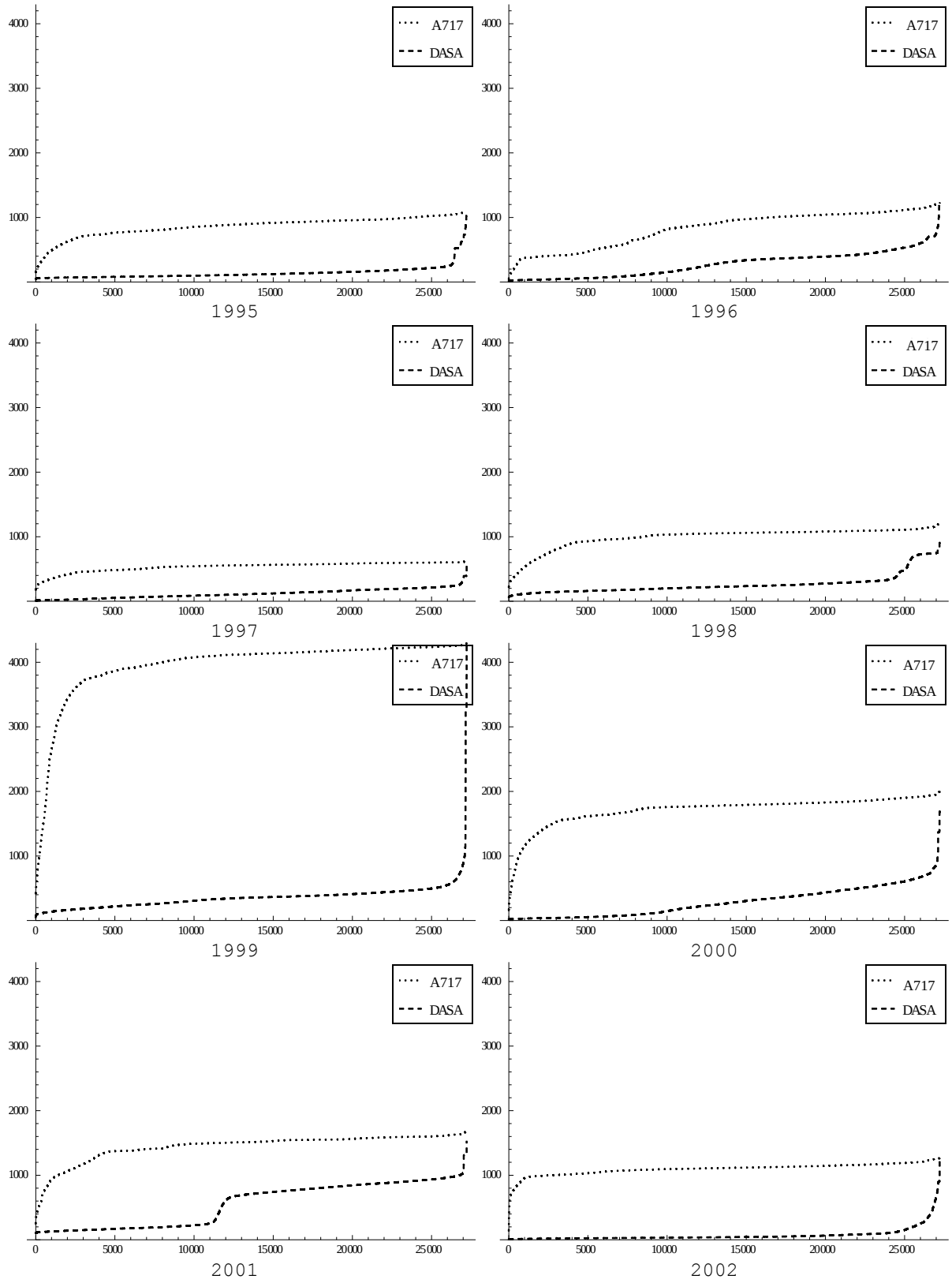


Figure 7.13: **Error profiles.** The eight graphs represent the error profile curves for the eight automated modeling tasks corresponding to the eight yearly datasets. The two curves in each graph depict the error (in terms of RMSE) profile obtained by ProBMoT performing parameter estimation with A717 (black dotted line) and DASA (black dashed line).

smaller error), whereas a negative value means that A717 is better than DASA on that model structure. We are interested in the differences which correspond to the best model structures found by A717. If the distribution of these differences is above 0, then DASA does not miss the best model structures found by A717.

In this context, Figure 7.14 depicts the distributions of error differences between models fitted with DASA and with A717 for all considered automated modeling tasks, that is, on all considered datasets. The first box-plot in each graph represents a summary of the differences of the two best models found with A717. If these differences are above zero, DASA does not miss the two best solutions found by A717. Next, the distributions of differences for the 10, 100, 1000 and 10000 best models are shown. Finally, the distribution of differences for all models is shown. In all cases, there is a clear shift of the distribution away from zero, which indicates that DASA overall finds better parameter values than A717 for the same model structures. Moreover, in the vast majority of cases, DASA manages to find better parameter values for the model structures for which A717 finds the best values.

7.6 Summary

We addressed the task of parameter estimation (from measured data) in ordinary differential equation models of ecosystem dynamics. The example model considered is a model of the food web dynamics in Lake Bled, which is nonlinear (due to the nonlinearity of the behavior of the modeled system) and has many parameters (high dimensionality). The measurements available are sparse and imperfect (due to measurement noise). These properties make parameter estimation in ecological models, in general, and the model of Lake Bled, in particular, a challenging optimization problem, calling for the use of advanced optimization methods.

We conducted an experimental comparison of five optimization methods: the differential ant-stigmergy algorithm (DASA), the continuous differential ant-stigmergy algorithm (CDASA), particle-swarm optimization (PSO), and differential evolution (DE) belong to the class of meta-heuristic methods, while Algorithm 717 (A717) is a derivative-based local search method. These methods were considered in conjunction with two model simulation approaches, teacher forcing simulation (TF) and full simulation (FS): the first is fast, but requires full observability, while the second allows for partial observability at a higher computational cost. We used both measured (real observational) data and artificial (simulated) data with different amounts of artificial (Gaussian) noise: the use of artificial data allowed us a more controlled study of the influence of noise and the underlying simulation approach on the performance of parameter estimation methods.

The meta-heuristic global optimization methods for parameter estimation are clearly superior and should be preferred over local optimization methods. While the differences in performance between the different methods within the class of meta-heuristics are not significant across all conditions, differential evolution yields the best results in terms of quality of the reconstructed system dynamics as well as speed of convergence. While the use of teacher forcing simulation makes parameter estimation much faster, the use of full simulation produces much better parameter estimates from real measured data.

We further investigated the benefit of using global optimization methods within the automated modeling of aquatic ecosystems, such as Lake Bled. When searching the same (large) space of model structures considered by Atanasova et al. (2006) and one-year data batches for parameter estimation, a per-model-structure comparison reveals that global optimization methods (DASA) find much better parameter values than local optimization

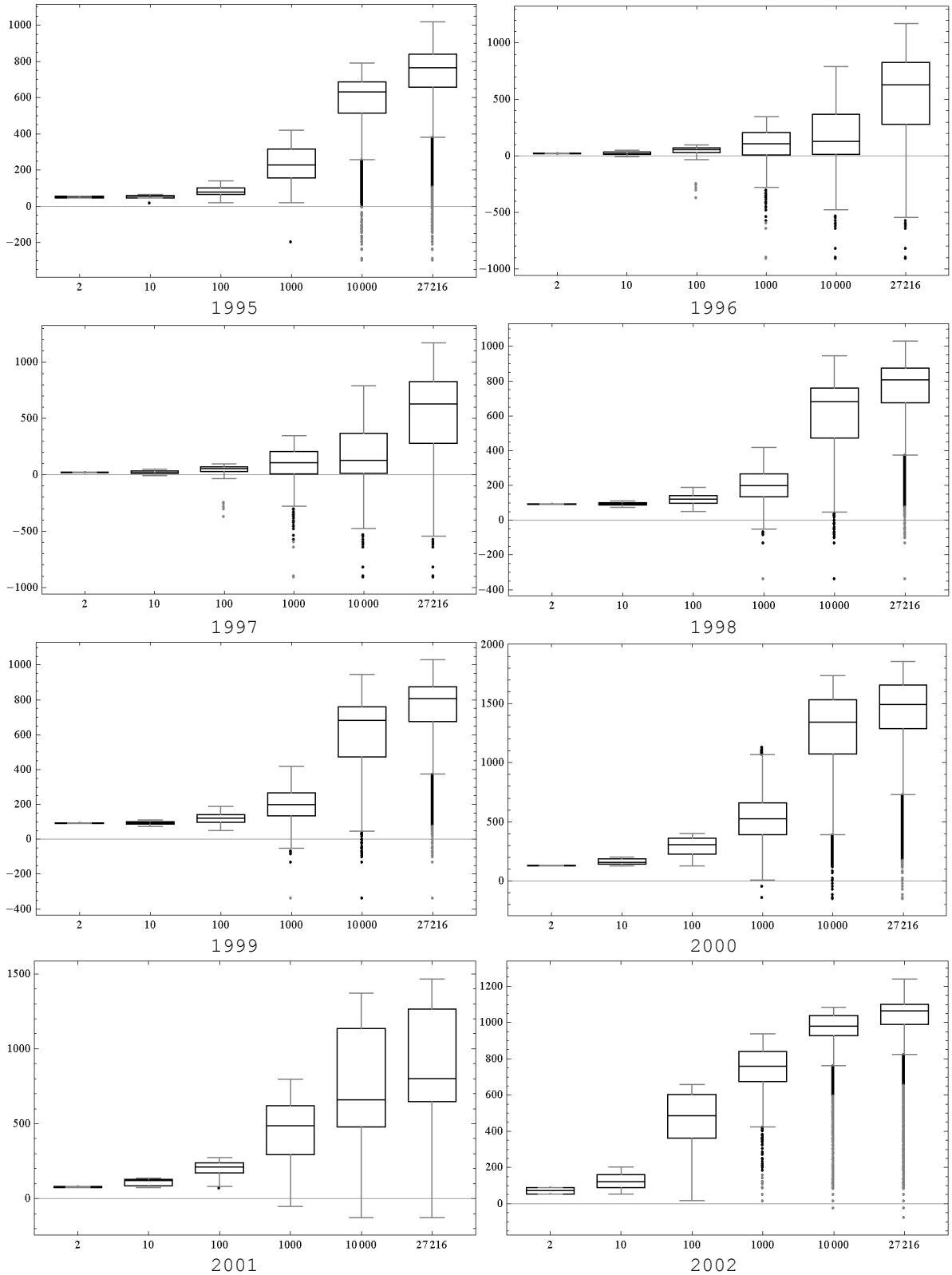


Figure 7.14: **Distribution of the model-wise error differences.** The graphs depict the distribution of the differences of RMSE for the models obtained by ProBMoT/A717 when refitted with DASA for all considered automated modeling tasks. The distributions of RMSE differences for the 2, 10, 100, 1000, and 10000 best models, and finally all 27216 models, are represented by boxplots.

approaches (A717), confirming the findings of the single-model parameter estimation task. Furthermore, the parameter estimation approach using local optimization (A717) is only able to find good parameter values for a small number of model structures. As these stick out from the large number of model structures for which poor parameter values are found, the problem of selecting a good model structure appears much easier than it really is.

The global optimization method (DASA), on the other hand, finds (equally) good parameter values for a large number of model structures, based on a single one-year data batch. This makes it clear that it is not easy to select an appropriate model structure, or more precisely, that a single one-year data batch does not provide enough information to narrow down the choice to a single (or a few) model structure(s). It is thus necessary to use more data on several years (a longer batch or several one-year batches) to obtain additional information that would further narrow down the choice among the large number of model structures. While LAGRAMGE 2.0 could not find models that fit well longer batches of data, this was likely due to the unsuccessful parameter estimation with local approaches: with the use of global approaches, we expect that we will be able to find such models (structures and parameters).

7.7 Additional figures and tables

Additional results regarding the task of parameter estimation in the food web model of Lake Bled are summarized in Figure 7.15 and Tables 7.8–7.10. The corresponding figure and tables are properly referenced in Section 7.4, which discusses all results obtained on the parameter estimation task in the food web model.

Figure 7.15 visualizes the distribution of the estimated model parameters generated with a Monte Carlo-based approach, using DE as the baseline estimation method, full model simulation and artificial data with 20% relative noise.

Table 7.8 summarizes the relative errors of the estimated model parameters obtained by the five optimization methods (DASA, CDASA, PSO, DE, A717) using the two simulation approaches (TF, FS) and four artificial datasets corresponding to four noise levels.

Table 7.9 presents the best parameter values as estimated by the five optimization methods (DASA, CDASA, PSO, DE, A717) using the two simulation approaches (TF, FS) and real data measured in 1996.

Table 7.10 summarizes the statistics for the estimated model parameters with the Monte Carlo-based approach, using DE as the baseline estimation method, full model simulation and artificial data with 20% relative noise.

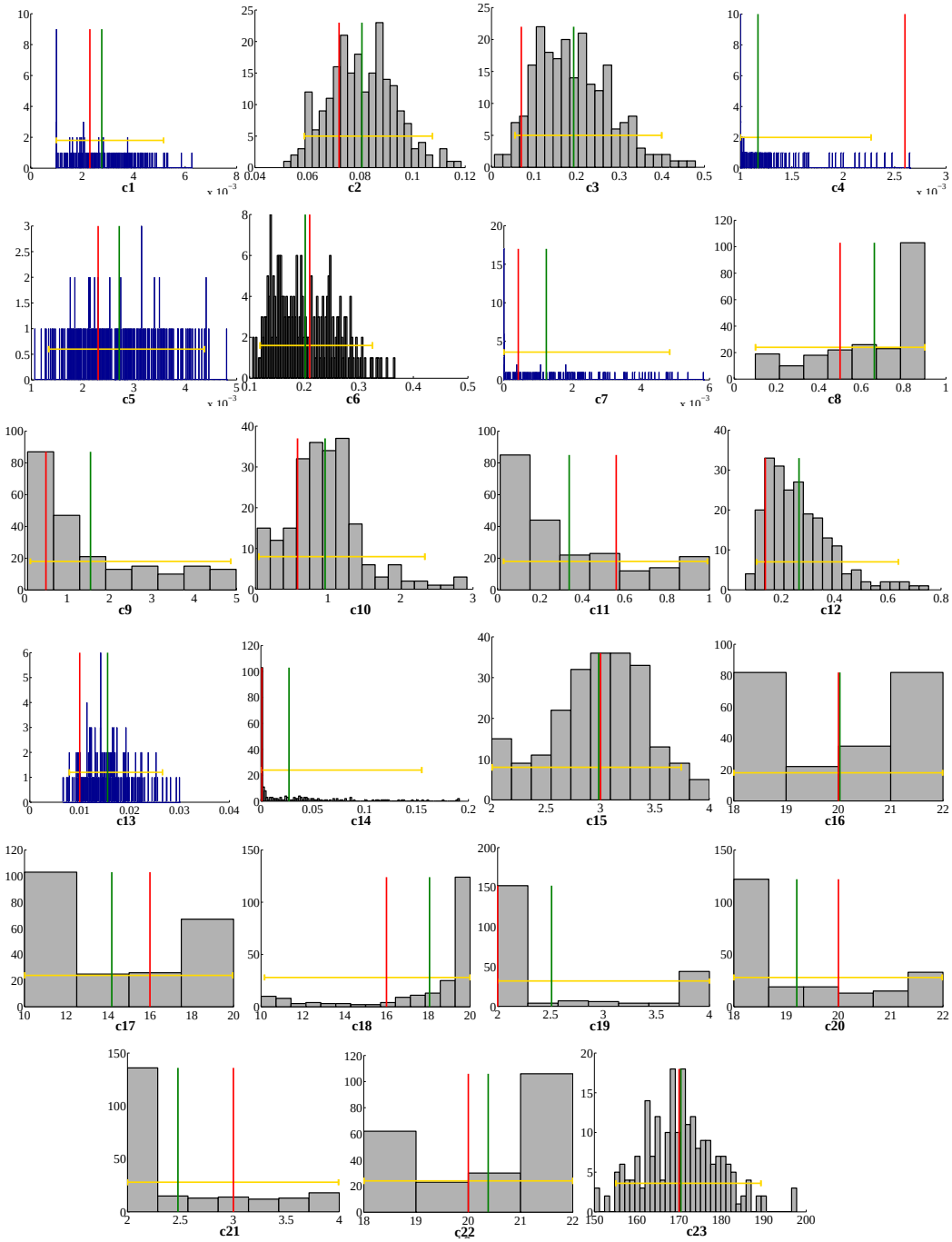


Figure 7.15: **Histograms of the parameter values generated with a Monte Carlo-based approach (estimated by DE using full simulation and artificial data with 20% relative noise).** The red vertical line represents the "true" value of the estimated parameter; the green vertical line represents the mean value of the sample μ; the yellow horizontal line visualizes the uncertainty of the estimated parameter, *i.e.*, its 95% confidence interval. The width of the bins h for a single histogram is calculated according to the Freedman-Diaconis rule $h = \frac{2\text{IQR}}{\sqrt[3]{N}}$, where IQR is the interquartile range of the sample and N is the size of the filtered sample (it includes the estimates obtained by those datasets that did not produce any outlier over all parameters), here $N = 221$.

Table 7.8: **Relative errors of the best parameter values estimated from artificial data.** A summary of the relative errors of the estimated parameters obtained by the five optimization methods (DASA, CDASA, PSO, DE, and A717) using the two simulation approaches (TF and FS) and four artificial datasets (four noise levels). The relative error (given in percent) of an estimated parameter c is calculated with regard to its reference value c^* by the formula $\frac{|c^*-c|}{c^*}$. For an explanation of the estimated model parameters see Table 7.2.

			TF					FS				
c	c^*	Noise	DASA	CDASA	PSO	DE	A717	DASA	CDASA	PSO	DE	A717
c_1	0.0023	0%	18	6	190	4	1143	16	39	452	0	17298
		5%	49	38	170	8	41408	47	27	51	9	18391
		20%	66	34	432	10	17826	21	13	494	8	11977
		50%	6116	1137	2532	1885	5888	12804	34316	19364	40022	38961
c_2	0.072	0%	60	5	77	6	38	40	4	69	10	65
		5%	57	7	87	9	83	27	10	64	9	77
		20%	72	17	77	16	84	7	41	61	51	1
		50%	87	87	87	87	39	51	86	24	106	96
c_3	0.07	0%	99	26	64	123	891	108	96	782	9	208
		5%	99	97	89	43	141	98	28	61	16	716
		20%	99	99	20	101	72	48	8	616	170	98
		50%	275	62	99	639	69	1288	848	801	1021	31
c_4	0.0026	0%	62	45	102	54	427	61	151	62	14	788
		5%	62	19	970	23	1885	34	62	61	29	1031
		20%	62	62	52	61	802	62	62	62	61	18755
		50%	24	62	61	61	3694	325	6179	11286	11211	7787
c_5	0.0023	0%	71	15	208	3	38055	27	2	424	0	1230
		5%	96	5	239	1	8774	27	5	44	3	8686
		20%	141	9	313	10	4459	22	58	462	7	5009
		50%	4331	885	3254	1504	29790	2121	16109	17556	29290	22494
c_6	0.21	0%	633	12	550	38	383	392	856	756	15	1035
		5%	474	597	779	34	126	1328	1030	1281	38	477
		20%	1209	180	1302	14	683	201	393	806	38	482
		50%	57	957	784	75	1017	1329	791	1213	1312	944
c_7	0.00042	0%	99945	527	60991	18	568426	21734	27493	51064	2	18091
		5%	62971	18369	89901	100	11385	69747	45826	29735	48	12128
		20%	207937	8729	147652	100	104853	8556	16752	77050	100	6273
		50%	5128	163919	377231	895	316767	7881	5491	18186	10401	5732
c_8	0.5	0%	75	3	43	1	45	27	48	73	10	19
		5%	62	37	79	30	64	56	80	33	19	11
		20%	74	64	79	53	27	26	56	73	10	45
		50%	80	80	80	80	50	80	77	25	41	8
c_9	0.5	0%	83	654	199	869	48	29	34	145	820	380
		5%	205	414	202	898	455	96	862	51	609	511
		20%	91	681	125	362	31	89	351	225	40	329
		50%	424	34	290	264	60	876	279	331	891	821
c_{10}	0.58	0%	97	80	13	10	18	67	0	54	1	112
		5%	5	65	97	8	374	144	43	97	2	37
		20%	19	52	96	81	125	417	93	54	38	169
		50%	96	84	155	98	86	191	416	85	43	37
c_{11}	0.56	0%	79	70	96	90	34	40	72	86	88	76
		5%	96	97	74	88	76	79	90	34	85	56
		20%	78	96	78	37	88	71	23	74	40	3
		50%	63	32	100	6	55	26	15	57	54	34
c_{12}	0.14	0%	489	50	455	1	215	36	432	203	1	441
		5%	614	395	533	0	295	614	61	49	4	237
		20%	614	500	230	32	78	465	285	142	39	484
		50%	614	614	82	614	548	131	614	614	612	345
c_{13}	0.01	0%	11	12	37	10	3089	4	1	531	2	8103
		5%	22	41	60	11	4428	6	11	46	11	3726
		20%	33	26	208	23	287	420	30	310	53	5454
		50%	614	614	82	614	548	131	614	614	612	345
c_{14}	0.001	0%	90	89	112041	1757	89314	90	2772	147299	285	51256
		5%	2929	5153	4754	716	133905	90	44	697	404	122920
		20%	89	90	147399	87	144470	62937	52	116046	77	88209
		50%	90	90	90	90	28962	90	308	69970	3225	36989
c_{15}	3	0%	28	9	29	0	5	31	4	5	0	26
		5%	9	8	33	5	1	12	4	29	5	5
		20%	5	22	31	19	5	8	23	6	13	25
		50%	10	33	33	32	22	2	23	28	33	26
c_{16}	20	0%	10	0	0	9	1	10	3	2	9	3
		5%	8	10	2	8	2	10	9	9	1	4
		20%	10	6	4	9	8	10	10	1	10	10
		50%	10	3	1	10	9	4	1	3	1	7
c_{17}	16	0%	37	11	1	37	1	27	17	6	15	10
		5%	35	24	20	27	22	34	17	14	36	27
		20%	37	0	1	19	19	14	10	16	35	1
		50%	17	5	3	21	13	17	14	22	37	19
c_{18}	16	0%	25	1	16	4	9	25	23	6	25	11
		5%	6	19	9	25	21	15	24	2	9	11
		20%	25	25	3	25	22	25	11	3	20	24
		50%	25	19	0	25	16	37	4	18	35	8
c_{19}	2	0%	100	74	66	2	88	39	74	83	0	48
		5%	100	66	62	0	84	50	22	10	0	55
		20%	100	98	1	0	89	46	5	94	0	93
		50%	100	0	81	0	34	0	64	83	0	70
c_{20}	20	0%	10	1	10	1	4	10	9	2	9	6
		5%	10	1	4	5	5	2	3	5	10	10
		20%	3	8	4	5	8	8	9	1	6	10
		50%	10	5	4	10	6	5	9	4	10	5
c_{21}	3	0%	33	33	0	4	30	22	4	23	0	8
		5%	7	10	31	3	14	25	15	32	3	8
		20%	11	25	32	33	17	15	21	24	20	33
		50%	33	2	7	15	10	33	20	31	33	16
c_{22}	20	0%	10	4	2	2	8	10	10	0	10	4
		5%	10	4	9	9	1	10	8	1	7	7
		20%	10	2	5	5	8	1	2	6	1	4
		50%	1	10	6	10	7	9	9	7	6	5
c_{23}	170	0%	12	11	38	1	10	6	37	47	0	16
		5%	17	26	39	0	7	24	47	19	1	18
		20%	14	48	67	9	18	18	44	53	2	1
		50%	35	1	61	12	70	12	9	13	12	61

Table 7.9: **Best estimated values of the model parameters from measured data.** Best parameter values as estimated by the five optimization methods (DASA, CDASA, PSO, DE, and A717) using the two simulation approaches (TF and FS) and real-experimental data measured in 1996. Note that the reference solution c^* corresponding to the artificial data case is given for comparison, since the "true" solution is unknown.

		TF					FS				
c	c^*	DASA	CDASA	PSO	DE	A717	DASA	CDASA	PSO	DE	A717
c_1	0.0023	0.0309	0.0010	0.0020	0.0118	0.0965	0.0020	0.0037	0.0292	0.0010	0.4531
c_2	0.072	0.0090	0.0090	0.0090	0.0101	0.0139	0.1491	0.1349	0.0223	0.1499	0.0758
c_3	0.07	0.4743	0.0010	0.0870	0.3277	0.0448	0.0012	0.0048	0.0258	0.0018	0.5381
c_4	0.0026	0.0010	0.0010	0.0053	0.0010	0.0832	0.0763	0.0916	0.0299	0.0672	0.0236
c_5	0.0023	0.0223	0.0065	0.0106	0.0128	0.8119	0.0026	0.0041	0.0133	0.0018	0.2243
c_6	0.21	0.0500	2.3011	0.6789	0.0500	1.3045	1.2918	0.6295	2.5939	0.1798	2.6245
c_7	0.00042	0.0035	1.0090	0.2823	0.0016	4.4166	0.0335	0.0186	0.2668	6.62E-07	0.3121
c_8	0.5	0.1032	0.1000	0.1396	0.1000	0.1163	0.1001	0.1004	0.1020	0.2179	0.3474
c_9	0.5	1.5608	0.1783	1.8332	4.0940	1.0316	0.8618	3.2755	1.6837	4.9873	2.5992
c_{10}	0.58	3.0000	2.5544	2.9578	3.0000	1.8641	0.6507	0.3040	0.2999	0.0457	0.2130
c_{11}	0.56	0.0329	0.2572	0.0198	0.0140	0.6284	0.5539	0.3663	0.6502	1.0000	0.7937
c_{12}	0.14	1.0000	1.0000	0.8136	0.8680	0.2505	0.7417	0.5360	0.5817	0.5115	0.9699
c_{13}	0.01	0.0257	0.0288	0.1430	0.0257	0.1490	0.2228	0.1662	0.5147	0.0228	0.5392
c_{14}	0.001	0.0001	0.0001	1.2673	0.0001	1.2660	1.4997	1.3878	1.2036	0.0001	0.8656
c_{15}	3	2.0000	3.1233	2.8119	2.1071	3.7151	2.0055	2.0011	2.9836	3.4209	3.0201
c_{16}	20	21.9990	21.7645	18.6412	18.8725	18.1007	18.0180	19.1627	18.2608	18.0075	19.4007
c_{17}	16	19.4268	11.9359	13.3350	20.0000	15.2645	16.8503	13.7229	13.8550	12.3858	11.6181
c_{18}	16	20.0000	16.0744	17.5405	20.0000	17.3212	10.0071	11.8315	16.8954	16.4055	19.1534
c_{19}	2	3.3257	3.0596	2.0336	4.0000	3.7967	3.3049	2.3678	3.7292	2.0290	3.8328
c_{20}	20	21.7262	20.2927	20.1014	22.0000	19.4098	21.6538	20.0177	21.1171	21.5881	21.2931
c_{21}	3	4.0000	3.1173	3.0174	4.0000	3.6806	4.0000	3.9998	3.4659	4.0000	3.5911
c_{22}	20	21.9924	19.3173	20.8803	21.4531	19.1552	18.0063	19.5728	20.7855	21.1962	19.9506
c_{23}	170	154.0525	180.9137	224.2833	150.0000	196.6290	150.3526	169.6731	258.5966	150.0161	216.1660

Table 7.10: **Summary statistics for the parameter values generated with a Monte Carlo-based approach (estimated by DE using full simulation and artificial data with 20% relative noise).** The column μ represents the mean values of estimated parameters; σ represents the standard deviation of the estimated parameters; $\frac{|c^*-\mu|}{c^*}$ is the relative error; c^{l} is the 2.5 percentile of the sample values, the lower bound of the 95% confidence interval; c^{m} is the 50 percentile of the sample values, the median; c^{u} is the 97.5 percentile of the sample values, the upper bound of the 95% confidence interval; $\mathrm{CI} = c^{\mathrm{u}} - c^{\mathrm{l}}$ is the length of the 95% confidence interval; $\frac{\mathrm{CI}}{\mu}$ is the confidence interval relative to the mean; the last column represents the number of outliers. The complete sample size is 400. For calculating the statistics, we used a reduced sample with size 221, without outliers.

c	c^*	μ	σ	$\frac{\lvert c^*-\mu\rvert}{c^*}$[%]	c^{l}	c^{m}	c^{u}	CI	$\frac{\mathrm{CI}}{\mu}$[%]	outliers
c_1	0.0023	0.003	0.001	20.0	0.001	0.003	0.005	0.004	150.7	6
c_2	0.072	0.081	0.013	12.1	0.059	0.080	0.108	0.049	60.4	8
c_3	0.07	0.193	0.088	175.6	0.055	0.182	0.400	0.344	178.5	6
c_4	0.0026	0.001	0.000	55.0	0.001	0.001	0.002	0.001	108.5	74
c_5	0.0023	0.003	0.001	17.9	0.001	0.003	0.004	0.003	111.9	10
c_6	0.21	0.202	0.057	3.8	0.119	0.193	0.325	0.206	101.8	11
c_7	0.00042	0.001	0.001	194.8	0.000	0.001	0.005	0.005	390.8	25
c_8	0.5	0.662	0.254	32.4	0.101	0.741	0.900	0.799	120.8	0
c_9	0.5	1.553	1.414	210.5	0.133	1.000	4.864	4.731	304.7	0
c_{10}	0.58	0.961	0.520	65.6	0.047	0.940	2.336	2.290	238.3	60
c_{11}	0.56	0.338	0.298	39.7	0.029	0.227	0.990	0.962	284.6	0
c_{12}	0.14	0.267	0.126	90.4	0.107	0.244	0.639	0.532	199.7	41
c_{13}	0.01	0.016	0.005	56.0	0.008	0.015	0.027	0.019	119.7	16
c_{14}	0.001	0.027	0.043	2607.8	0.000	0.002	0.155	0.155	572.3	47
c_{15}	3	2.983	0.452	0.6	2.003	3.005	3.742	1.738	58.3	0
c_{16}	20	20.028	1.519	0.1	18.000	20.154	21.993	3.992	19.9	0
c_{17}	16	14.176	3.715	11.4	10.014	13.268	19.946	9.932	70.1	0
c_{18}	16	18.059	2.880	12.9	10.170	19.430	19.998	9.828	54.4	0
c_{19}	2	2.510	0.801	25.5	2.000	2.005	4.000	2.000	79.7	0
c_{20}	20	19.204	1.429	4.0	18.000	18.389	21.986	3.986	20.8	0
c_{21}	3	2.477	0.658	17.4	2.000	2.043	3.993	1.993	80.5	0
c_{22}	20	20.381	1.516	1.9	18.004	20.836	22.000	3.995	19.6	0
c_{23}	170	170.427	8.858	0.3	155.096	170.326	189.322	34.226	20.1	7

8 Conclusions

In this dissertation, we focused on the task of parameter identification in nonlinear dynamics models from measured data: given a model structure and measured data, the goal of parameter identification is to estimate the unknown model parameters that will determine a best possible fit between the measurements and model predictions. We approached the task from the frequentist point of view using least-squares estimation. We used meta-heuristic optimization in the context of modeling deterministic systems represented by ordinary differential equations (ODE) relevant for real-life domains, such as systems biology and ecological modeling.

In general, comprehensible biological and ecological models, obtained as a result of first-principle mechanistic modeling, have nonlinear continuous dynamics described with a large number of unknown or roughly known parameters. Accurate knowledge of the parameter values is important for describing and analyzing the behavior of the modeled systems. As a result, the task of parameter identification has an important role in the emerging domains of systems biology and ecological modeling.

However, due to the complex nonlinear dynamics of the modeled systems and the limitations of measurement technology, parameter identification is recognized to be among the most challenging tasks in the process of modeling biological and ecological systems. Moreover, it is the computational bottleneck of this process. More precisely, least-squares parameter estimation is a difficult optimization problem, calling for the use of global optimization methods, such as meta-heuristics.

With this motivation, we performed a thorough empirical evaluation of representative meta-heuristic methods on the task of estimating parameters in nonlinear dynamic (ODE) models of two practically relevant real-life systems, *i.e.*, endosome maturation in endocytosis and the food web in Lake Bled. Different experimental scenarios were considered to investigate the performance of the methods to identify the parameters in the case of noisy, limited (incomplete), and misinterpreted data as well as when (two) different approaches for ODE integration are used. The final empirical analysis considered the problem of parameter identification as related to the task of structure identification within the process of automated modeling of dynamic systems. In this context, we empirically evaluated the influence of the parameter identification method (global vs. local optimization method) on the outcome of the automated modeling process.

The remainder of this chapter first summarizes the findings of the conducted empirical studies. These are presented in Section 8.1 and 8.2, each corresponding to one of the two real-life systems being studied. Next, Section 8.3 outlines the original contributions of the dissertation. Finally, Section 8.4 discusses the possible directions for further work.

8.1 Modeling a biological system

We addressed the task of parameter identification in models of the dynamics of biological systems as considered in the field of systems biology. In this context, it is typical that the considered models are nonlinear (due to the nonlinearity of the behavior of the modeled

systems) and have many parameters (are of high dimensionality). The measurements can be imperfect (due to measurement noise) and the system can be only partially observed (leading to incomplete or misinterpreted measurements). These properties make parameter identification a challenging optimization problem, calling for the use of advanced optimization methods. In this context, we focused on the use of meta-heuristic optimization methods for parameter identification in dynamic system models typical of systems biology.

We conducted an extensive experimental comparison of four optimization methods: the differential ant-stigmergy algorithm (DASA), particle-swarm optimization (PSO), and differential evolution (DE), all from the same class of meta-heuristic methods, as well as a derivative-based local search method, Algorithm 717 (A717). We compared these four methods as applied to a parameter identification problem representative of the target class of problems described above. We used a practically relevant model of endocytosis that captures the nonlinear dynamics of endosome maturation reflected in a cut-out switch transition between the Rab5 and Rab7 domain protein concentrations. The model is nonlinear and has many parameters. We compared the performance of the four optimization methods on this task along a number of dimensions, including the quality of reconstructing the observed system output (the measured quantities) and the complete model dynamics (all system variables, including unobserved ones), as well as the speed of convergence. Comparisons were made under different observation scenarios, including complete observability and three different types of partial observability. We used both real (measured) data, containing partial observations of the system, and artificial (simulated) data obtained by simulating the model and adding different amounts of artificial (Gaussian) noise: the use of artificial data allows us a more controlled study of the influence of noise and observability on the performance of the parameter identification (optimization methods).

Noise in the measurements does influence the performance of the optimization methods, with higher amounts of noise making the task more difficult. The observability of the system (as varied through the observation scenarios), has a much stronger influence, where less complete observations make the optimization task much more difficult. As one would expect, the easiest optimization tasks stem from the complete observation scenario (CO), since in this scenario all system variables are directly observed. However, the total protein concentration observation scenario (TO) corresponding to the real biochemical measurement process, although complex (the observed outputs are linear combinations of the system variables) compares favorably to the other partial observability scenarios, active-state protein concentration observation (AO) and neglecting passive-state protein concentration observation (NPO), in terms of complete system dynamics reconstruction. This can be explained by the fact that the outputs observed in the TO scenario carry more information about the system (they include both active and passive states of proteins) than the observed outputs (active-state proteins only) in the other two scenarios. Worst results are obtained when the observations are misinterpreted (NPO), *i.e.*, when the actual total concentrations of Rab5 and Rab7 are taken to represent the concentrations of these proteins in their active states.

We also investigated the practical identifiability of the model parameters. Like many similar tasks in systems biology, the task considered has parameter identifiability problems. These are manifested by high relative errors of the reconstructed parameter values, spread uniform-like distributions of some parameter estimates, and strong correlations between some pairs of estimated parameters. The problems are present in all observation scenarios and are most severe in the case of incomplete observations. The performance of all three meta-heuristic methods is affected by these problems. On the other hand,

this explains the severe difficulties that the local search method A717 experienced on the given parameter identification task.

Overall, the global meta-heuristic methods (DASA, PSO, and DE) clearly and significantly outperform the local derivative-based method A717. Among the three meta-heuristics, DE performs best in terms of the objective function, *i.e.*, the quality of reconstructing the expected output, and in terms of the speed of convergence. These results hold for both real and artificial data, for all observability scenarios considered, and for all amounts of noise added to the artificial data. In terms of the quality of reconstructing the complete model dynamics and other qualitative aspects of the behavior of the obtained models (the time point of the switch and the ratio between active- and passive-state protein concentrations), the different meta-heuristic methods exhibit different behavior and relative performance under different conditions: more work needs to be done to better understand and objectively evaluate these differences in performance.

In sum, the bio-inspired meta-heuristic optimization methods considered are suitable for estimating the parameters in the ODE model of the dynamics of endocytosis under a range of conditions. The model considered, as well as the observational conditions (such as partial observability and noise) are representative of parameter identification tasks in ODE models of biochemical network dynamics. Thus, our results point out and clearly highlight the promise of bio-inspired meta-heuristic methods for solving problems of parameter identification in models of dynamic systems from the area of systems biology.

8.2 Modeling an aquatic ecosystem

The conclusions regarding the task of modeling dynamics of the food web in Lake Bled are summarized in two subsections. The first summarizes the findings with respect to the task of parameter identification in a single model structure of the Lake Bled dynamics, while the second outlines the findings of the study involving parameter identification in many model structures in the context of automated modeling of the Lake Bled dynamics.

8.2.1 Parameter identification in a single model

We addressed the task of parameter identification (from measured data) in ordinary differential equation models of ecosystem dynamics. The example model considered is a model of the food web dynamics in Lake Bled: the particular model structure used was discovered in a previous study (Atanasova et al., 2006) with the automated modeling tool LAGRAMGE 2.0 (Todorovski and Džeroski, 2006) from data measured in 1996. The model is nonlinear (due to the nonlinearity of the behavior of the modeled system) and has many parameters (high dimensionality). The measurements available are sparse and imperfect (due to measurement noise). These properties make parameter identification in ecological models in general and the model of Lake Bled in particular a challenging optimization problem, calling for the use of advanced optimization methods.

We conducted an experimental comparison of five optimization methods: the differential ant-stigmergy algorithm (DASA), the continuous differential ant-stigmergy algorithm (CDASA), particle-swarm optimization (PSO), and differential evolution (DE) belong to the class of meta-heuristic methods, while Algorithm 717 (A717) is a derivative-based local search method. These methods were considered in conjunction with two model simulation approaches, teacher forcing simulation (TF) and full simulation (FS): The first is fast, but requires full observability, while the second allows for partial observability at a higher computational cost. We used both real (measured in 1996) data and artificial

(simulated) data with different amounts of artificial (Gaussian) noise: the use of artificial data allowed us a more controlled study of the influence of noise and the simulation approach on the performance of parameter identification methods. Finally, we performed a model validation on real data measured in 1997.

We compared the performance of the different optimization methods on the task at hand in terms of the quality of reconstructing the complete model dynamics (all system variables), as well as the speed of convergence. Overall, the derivative-based method A717 is clearly and significantly inferior to the meta-heuristic methods (DASA, CDASA, PSO, and DE). Among these four, DE performs best in terms of the quality of reconstructed model dynamics and speed of convergence. DE is also the most precise method in terms of the variance of the model error. These results hold for both real and artificial data, for both simulation approaches (TF and FS) considered, and for all amounts of noise added to the artificial data. Nevertheless, noise in the measurements does influence the performance of the optimization methods, with higher amounts of noise making the task more difficult.

The results of parameter identification from artificial data did not show a clear advantage of the identification procedure with FS over the one with the TF model simulation approach. However, there is no doubt about the computational cost of the considered approaches: parameter identification with TF simulation is almost two orders of magnitude faster than identification with the FS approach. When using real measured data, however, the performance differences are clearer. According to the visual inspection of the model predictions, FS-based identification resulted in better models. Among the considered alternatives, parameter identification with the A717 method using the TF approach for model simulation, resulted in models with poorest quality.

We also assessed the practical parameter identifiability of the model at hand. This revealed that the task considered has parameter identifiability problems, which is typical for complex nonlinear ecological models. These are manifested by high relative errors of the reconstructed parameter values, spread uniform-like distributions of some parameter estimates, and strong correlations between some pairs of parameter values. Furthermore, two parameters seem to be structurally non-identifiable, as they have no influence on the profile of the model error landscape. The performance of all four meta-heuristic methods is affected by practical identifiability problems on the addressed task: on the other hand, these explain the severe difficulties that the local search method A717 experienced on this task.

While parameter identification from artificial (noisy) data clearly showed the capability of meta-heuristics to identify parameter values that are able to capture the observed dynamics, this is not the case with parameter identification from measured data. Even the best parameters found for the considered structure of the Lake Bled model, are not completely appropriate for modeling the dynamics of the lake in 1996 and are completely inappropriate for modeling the dynamics in 1997. This may be due to the difference in the dynamics of the lake between the two years, but may also be due to the inappropriateness of the model structure. Recall that the model structure was chosen among a large set of candidates based on the degree of fit produced by the worst-performing parameter optimization method considered in this study.

Nevertheless, the results of our comparative study lead to a clear recommendation: given their superior performance, the use of meta-heuristic global methods for parameter identification should be highly preferred over the use of local search methods. While the differences in performance between the different methods within the class of meta-heuristics are not significant across all conditions, differential evolution yields the best

results in terms of quality of system dynamics reconstruction as well as speed of convergence.

8.2.2 Parameter identification and model selection in automated modeling

Given the imperfect match of even the best models found in the previous study with the data used for calibration (measured in 1996) and poor fit to the data used for validation (measured in 1997), it is evident that the model structure selected (by LAGRAMGE 2.0, (Atanasova et al., 2006)) is not appropriate. However, it is clear that the use of global optimization methods with full simulation finds better parameter estimates as compared to local optimization method with teacher forcing. This brings to the front the important issue of the interplay between parameter identification and model (structure) selection.

On one hand, one can expect that better parameter identification would make it easier to select a more appropriate model structure. On the other hand, one might suspect that better parameter identification may lead to over-fitting. It is not clear a-priori what the balance between the two would be in the context of automated modeling approaches, such as LAGRAMGE 2.0. Therefore, we conducted an extensive follow-up study of this issue: the influence of parameter identification methods on the final output of automated modeling, i.e., on what models are selected. More precisely, we investigated the effects of using DASA as a global optimization method instead of the previously used local optimization method A717 for parameter identification in the discovered model structures. The study involved eight tasks of automated modeling of Lake Bled dynamics corresponding to eight different sets of real measured data. The modeling tasks were performed with the automated modeling tool ProBMoT.

The results conclusively show that DASA outperforms A717 on all modeling tasks. Not only ProBMoT using DASA manages to find better models for the systems, DASA manages to find better parameter values across the whole spectrum of model structures. Furthermore, refitting the best model structure found by A717 with DASA yields better results in most cases. Automated modeling of Lake Bled dynamics has so far focused mostly on discovering single year models and this is the approach taken in this dissertation. The reason for this is that experiments with discovering models which hold for longer periods of time yielded poor results. We conjecture that this is largely due to the poor parameter identification when using local search methods. Using global search methods opens the opportunity to tackle long term modeling of dynamical systems with automated modeling tools.

One clue to support this conjecture is given by the error profile curves provided in Figure 7.13, Chapter 7. Almost without exceptions, A717 manages to find good parameter values for very few model structures, which can be seen by the shape of the error profile curve: the curve increases rapidly at the beginning and reaches a plateau of models with high error values afterwards. In contrast, the error profile curves of DASA show a plateau of models with low error values which are equally good, and poor parameter values for very few model structures, which is reflected in the increase in error values near the end of the curve. This means that the small datasets used do not provide enough information to discriminate among the different model structures. Hence, we need to use longer, multi-year datasets, to narrow down the choice of model structures.

8.3 Contributions

The research presented in this dissertation addresses important aspects of parameter identification in the context of both single-model identification and multiple-model identification arising within the context of automated modeling of dynamic system from experimental data. The corresponding research findings are published in several conference and journal publications (Tashkova et al., 2010b, 2011a, 2012b; Čerepnalkoski et al., 2011a): the complete list of related publications is given in *Appendix 1.* In the following, we summarize the main contributions of the work presented in this dissertation.

- **An overview of existing approaches for parameters identification** and their application to the class of problems of our interest (nonlinear ODE models in the domain of systems biology and ecological modeling) is presented. Given that parameter identification in ODE models of nonlinear dynamic systems belongs to the class of nonlinear optimization problems, methods for nonlinear optimization are surveyed. Furthermore, an overview of the most promising meta-heuristics and their application to the parameter identification problem is provided as well.

- **A thorough empirical evaluation of representative meta-heuristic methods is performed on the task of estimating parameters in nonlinear dynamic (ODEs) models.** In this respect, four global-search meta-heuristic algorithms for continuous optimization, *i.e.*, the differential ant-stigmergy algorithm (DASA) and its continuous variant (CDASA), particle-swarm optimization (PSO), and differential evolution (DE), as well as Algorithm 717 (A717), a derivative-based local search method, are considered for comparison. These are applied to representative tasks of estimating parameters in the field of systems biology and ecological modeling.

  - **Identification of parameters in ODE models of two practically relevant real-life systems, *i.e.*, endosome maturation in endocytosis and the food web in Lake Bled**. The first model describes a key cellular process that switches from cargo transport in early endosomes to cargo disintegration in mature endosomes in the context of the phagocytosis process, through which bacteria are eliminated from human cells. Understanding the underlying mechanism in this case is important for properly treating bacterial infections in living organisms (*e.g.*, for drug development). The second model describes the dynamics of a food web in a representative aquatic ecosystem, *i.e.*, Lake Bled (Slovenia). Understanding the food web dynamics is important for preserving the natural ecosystem in the lake (environmental protection).
  - **The performance of the methods on the considered tasks is evaluated along a number of metrics, including the quality of reconstructing the system dynamics as well as the speed of convergence, both on real-experimental data and on artificial pseudo-experimental data with varying amounts of noise** (Tashkova et al., 2011a, 2012b). Furthermore, problem-specific tuning of the optimization methods is considered to provide a fair comparison of the meta-heuristics' performance (Tashkova et al., 2011a). *The evaluation shows that the meta-heuristic global optimization methods for parameter identification are clearly superior and should be preferred over local optimization methods.* While the differences in performance between the different methods within the class of meta-heuristics are not significant across

all conditions, DE yields the best results in terms of the quality of the reconstructed system dynamics as well as the speed of convergence.

- **The impact of limited observability on the difficulty of the parameter identification task** (and consequently on the performance of different optimization methods) is evaluated in the context of reconstructing endosome maturation dynamics (Tashkova et al., 2011a). For this purpose, the optimization methods are compared under a range of observation scenarios, where data of different completeness and accuracy of interpretation are given as input. *The observability of the system (as varied through the observation scenarios), shows a strong influence on the success of the parameter identification, where less complete observations make the optimization task much more difficult.* Worst results are obtained when the observations are misinterpreted. In terms of the quality of reconstructing the complete model dynamics, the different meta-heuristic methods exhibit different behavior and relative performance under different conditions. Since these qualitative aspects have not been included in the cost function (sum of squared errors) used by the optimization methods, we cannot objectively and fairly compare the methods along this dimension. However, the observation that a good model with respect to the observed system dynamics does not necessarily mean a good match of the complete system dynamic, clearly shows the importance of choosing a relevant cost function when the modeled system dynamics is partially observed.
- **The impact of the ODE simulation method on the parameter identification task** is evaluated in the context of reconstructing Lake Bled dynamics (Tashkova et al., 2012b). For this purpose, two different simulation approaches are considered: teacher forcing, a one-step trapezoidal-based integrator for supervised predictions, and full simulation, a multistep variable-coefficient integrator which makes unsupervised predictions based on the history predictions for longer time periods. *While the use of teacher forcing simulation makes parameter identification much faster, the use of full simulation produces much better parameter estimates from real-experimental data. In the case of artificial (noisy) data, however, there is no clear difference between the models obtained by estimation based on teacher forcing and full simulation. Both seem to produce models of comparable quality as long as the noise in the data is not very high.* This result opens the opportunity to significantly reduce the computational time of the overall parameter identification process by combining teacher forcing with full simulation in some kind of a hybrid schema for model simulation.

- **Empirical analysis of the interplay between parameter identification and model structure selection within the process of automated modeling of dynamic systems**. More precisely, the influence of the parameter identification method (global vs. local optimization method) on the automated modeling process was empirically evaluated. In this context, the meta-heuristic method DASA was integrated within a new automated modeling tool ProBMoT (Čerepnalkoski et al., 2011a,b) and tested against the performance of ProBMoT with the local search method A717 on a real-life modeling problem, *i.e.*, the problem of modeling phytoplankton dynamics in Lake Bled from data measured across eight years. *The experimental evaluation on eight single-year modeling tasks empirically proved the benefit of estimating model parameters by global optimization method to the model (structure) selection process* (Čerepnalkoski et al., 2011a,b). The latter is

furthermore important in the context of modeling long term system dynamics, since automated modeling has so far focused mostly on discovering single-year models of this aquatic ecosystem, as considered here as well. The reason for this is that experiments for discovering models which hold for longer periods of time yielded poor results (Atanasova et al., 2006). One evident reason is the poor parameter identification due to the use of local search methods. Using global search methods therefore opens the opportunity to tackle long term modeling of dynamical systems with automated modeling tools.

8.4 Further work

The results of the undertaken experimental evaluations clearly reconfirmed how challenging and essential is the task of parameter identification for deriving accurate mathematical models of complex dynamic systems. However, further work is needed to confirm and strengthen the conclusions presented in this dissertation (primarily in the direction of conducting additional experiments). In this respect, several important directions for further work can be identified as follows.

On one hand, we need to address the task of parameter identification in other ecosystems, e.g., food web models of other aquatic ecosystems, and other biological systems. On the other hand, we can extend the set of optimization methods applied to the identify the unknown model parameters, considering other state-of-the-art algorithms used for parameter estimation in the domain of computational systems biology and ecological modeling (Sun et al., 2012). More precisely, we can use other meta-heuristics, such as scatter-search (Rodriguez-Fernandez et al., 2006a) and evolutionary strategies (Moles et al., 2003). We can also use hybrid global-local search methods (Rodriguez-Fernandez et al., 2006b; Balsa-Canto et al., 2008; Sun et al., 2012) designed to cope efficiently with highly multimodal and high-dimensional optimization problems.

Concerning efficient and effective optimization methods, we need to consider new or improved optimization strategies, especially meta-heuristics that are self-adaptive with respect to the characteristics of the optimization problem being solved. One of the main problems of the majority of meta-heuristic algorithms is the setting (tuning) of their control parameters. This can be especially important in the context of automated modeling: as the parameters of thousands of model structures with different complexity (and number of parameters) have to be estimated, tuning the method's parameters for each model structure is prohibitive in this case. In this respect, it is important to understand the behavior of existing optimization methods and furthermore summarize that knowledge in the form of guiding rules or adaptive schemas related to the characteristics of the search space landscape. Related to this, we are currently investigating the possibility to learn models of behavior of the differential ant-stigmergy algorithm (DASA) for different types of optimization problems. We considered a systematic evaluation of the DASA performance on 24 black-box dimension-scalable optimization problems with 5000 Sobol' sampled DASA parameter settings regarding two dimensions, 20 and 40. The idea is to perform intelligent analysis of the gathered data, by applying data mining methods, in order to extract patterns (regularities) in the explored DASA parameter space defining a specific behavior of DASA (Tashkova et al., 2012a).

Furthermore, we can replace the LS estimation with M-estimation as a more robust approach with respect to outliers in the data (data drawn from other distributions than the Gaussian) (Staudte and Sheather, 1990), or we can hybridize LS estimation with collocation methods, that will smooth the noise and impute the missing values in the

experimental data. Finally, although computationally more expensive, we can employ Bayesian approaches for parameter identification that are especially convenient for models with identifiability problems (Dowd and Mayer, 2003; Toni et al., 2009).

The empirical analysis regarding both endosome maturation and Lake Bled dynamics revealed that the sum of squared errors is not the most appropriate objective function with respect to the quality of the reconstructed systems dynamics. In this respect, we need to formalize relevant qualitative aspects of model quality (such as the time point of switch between the observed Rab5 and Rab7 concentrations in the endocytosis model, or peaks in the population dynamics) and include these in the formulation of the optimization problem of parameter estimation. These aspects will typically depend on domain knowledge about the particular problem at hand and can be made a part of the overall objective function or formulated as a separate objective function in a multi-objective optimization setting. This will allow us to objectively and fairly evaluate and compare the different optimization approaches from these aspects.

Other relevant issues that deserve further attention include limited observability and supervised ODE integration (teacher forcing *vs.* full simulation). As teacher forcing simulation is inexpensive, we will consider combining teacher forcing with full simulation in some kind of a hybrid schema for cheaper and yet reliable model simulation. For stiff ODE models, integration can be computationally expensive: consequently, the computation of the objective function and the complete parameter identification process will be time-consuming. Another direction for further work would thus be to avoid integration (Chou and Voit, 2009) by using slope estimation strategies (including smoothing methods such as collocation methods for reliable estimation from noisy data) or decoupling/decomposing strategies for large state-space models.

Knowing how expensive the objective function evaluation is, it is important to reduce as much as possible the number of function evaluations required. This can be achieved by exploiting knowledge of past evaluated points to train an empirical model that can be used as an inexpensive surrogate of the objective function. For example, Büche et al. (2005) accelerated an evolutionary algorithm with a Gaussian process model of the objective function, while Egea et al. (2009) improved a scatter search method by krigging-based surrogate models.

All of the above mentioned directions can be further integrated and investigated within the framework for automated modeling of dynamic systems, and tools such as LAGRAMGE 2.0 and ProBMoT. In this respect, we will consider the use of other global methods for parameter estimation, such as differential evolution, particle swarm optimization, or surrogate-based meta-heuristics for automated modeling of endosome maturation and constructing long term models of the dynamics of aquatic ecosystems. Furthermore, we plan to explicitly integrate tools for sensitivity and identifiability analysis within the automated modeling process, as these can give valuable feedback to the structure identification task or structure selection in particular. Sensitivity analysis is an important tool in studying the dependencies of observed dynamics on model parameters and can implicitly influence the optimization process and automated modeling in general. Identifying the parameters that have more impact on the system outputs and capture the essential characteristics of the system may reduce the complexity and dimensionality of the model. Therefore, it can be particularly useful when modeling complex biological networks that involve a large number of variables and parameters (Yue et al., 2006).

Identifiability analysis is closely related to parametric sensitivity analysis and techniques have been developed for identifiability analysis based on the sensitivity coefficient matrix (Jacquez and Greif, 1985). Related to this, De Pauw and De Baets (2008) used an aggregated metric based on a model identifiability measure (collinearity index, calcu-

lated from the parametric sensitivities) and relative mean squared error-based measure (Nash-Sutcliffe criterion) to assess the quality of a single model (structure) in an equation discovery system based on genetic programming. They considered the task of automated modeling in the context of reconstructing a river water quality model from artificially generated data using a simple two-state ODE model example. Their initial evaluation shows that identifiability information can be used to address complexity (overparametrization) and consequently overfitting issues arising within the process of automated modeling of dynamic systems.

As discussed towards the end of Chapter 7, successful parameter identification clearly reveals the difficulty of discriminating between model structures based on a small quantity of data. In this context, additional experiments may still need to be performed, in order to generate new data which will allow for discrimination between model structures that perform equally well, in terms of identifiability and degree of fit. As a result, future automated modeling approaches should not only passively accept experimental data, but also suggest experimental designs (quality and quantity of data) that will improve the accuracy and reliability of generated models (especially in terms of predictive performance).

9 Acknowledgments

The successful completion of this dissertation would have not been possible without the support of the following people and organizations during my PhD studies.

First of all, I would like to express my gratitude and appreciation to my supervisor Sašo Džeroski and co-supervisor Jurij Šilc for their guidance, inspiring discussions, encouragement and support during the course of this work. I'm sincerely thankful for their time spent in reviewing and editing my research work.

The next acknowledgment is addressed to Ljupčo Todorovski, Peter Korošec and Eva Balsa-Canto. I'm especially thankful to Ljupčo for the countless long-lasting fruitful discussions and PhD-survival advices. His insightful comments and suggestions greatly influenced my research work. I would like to thank Peter for his helpful suggestions and discussions related to DASA implementation, and other ACO-based optimization methods in the early stage of my research work. As members of my PhD evaluation board, I would like to thank both Ljupčo and Peter, as well as Eva Balsa-Canto, for carefully reading the text of the dissertation and providing useful feedback that improved the quality of the dissertation.

Furthermore, I thank my co-authors for the successful collaboration and their contribution to the papers related to this dissertation. Namely, Nataša Atanasova, Darko Čerepnalkoski, Peter Korošec, Ljupčo Todorovski, Jurij Šilc and Sašo Džeroski. Moreover, thanks to Perla Del Conte-Zerial and her coworkers for providing data on Rab5-Rab7 protein concentrations as well as Yannis Kalaidzidis for the invaluable discussions and feedback on the preliminary experimental results regarding the endosome maturation case study.

For the pleasant working environment at the Jožef Stefan Institute, I wish to thank all staff members of the Computer Systems Department. Special thanks, however, is addressed to my officemates Vida Vukasinović, Uroš Bole and Lucas Benedičič for making my time at the office more enjoyable.

I would like to acknowledge the financial support of the Slovenian Research Agency through the young researcher grant No. 1000-06-310026. In addition, thanks to NELA development center for the financial support during the period of writing the dissertation.

I would like to express my deepest gratitude to my family for their support during my studies. Especially, I wish to thank my mother Vangja for her unselfish love, complete dedication, persistent confidence in me and encouragement during the tough periods of my life. I dedicate this work to you Mom, as without your full support I would not have come to Ljubljana and complete this PhD journey. Finally, I owe a big thanks to Simon, who truly loved me, patiently stood by me and constantly encouraged me to follow my path: Thank You for the unconditional love and support in the last three years.

10 Bibliography

Afshar, A.; Kazemi, H.; Saadatpour, M. Particle swarm optimization for automatic calibration of large scale water quality model (CE-QUAL-W2): Application to Karheh Reservoir, Iran. *Water Resources Management* **25**, 2613–2632 (2011).

Akaike, H. A new look at the statistical model identification. *IEEE Transactions on Automatic Control* **19**, 716–723 (1974).

Apt, K. *Principles of Constraint Programming* (Cambridge University Press, Cambridge, 2003).

Arisi, I.; Cattaneo, A.; Rosato, V. Parameter estimate of signal transduction pathways. *BMC Neuroscience* **7**, S6 (2006).

Ashyraliyev, M.; Fomekong-Nanfack, Y.; Kaandorp, J. A.; Blom, J. G. Systems biology: Parameter estimation for biochemical models. *The FEBS Journal* **276**, 886–902 (2009).

Atanasova, N.; Todorovski, L.; Džeroski, S.; Rekar Remec, Š.; Recknagel, F.; Kompare, B. Automated modelling of a food web in lake Bled using measured data and library of domain knowledge. *Ecological Modelling* **194**, 37–48 (2006).

Athias, V.; Mazzega, P.; Jeandel, C. Selecting a global optimization method to estimate the oceanic particle cycling rate constants. *Journal of Marine Research* **58**, 675–707 (2000).

Bäck, T.; Fogel, D. B.; Michalewicz, Z. *Handbook of Evolutionary Computation* (IOP Publishing Ltd and Oxford University Press, New York, 1997).

Bäck, T.; Fogel, D. B.; Michalewicz, Z. *Evolutionary Computation 1: Basic Algorithms and Operators* (IOP Press, New York, 2000a).

Bäck, T.; Fogel, D. B.; Michalewicz, Z. *Evolutionary Computation 2: Advanced Algorithms and Operations* (IOP Press, New York, 2000b).

Balsa-Canto, E.; Alonso, A. A.; Banga, J. R. An iterative identification procedure for dynamic modeling of biochemical networks. *BMC Systems Biology* **4**, 11 (2010).

Balsa-Canto, E.; Peifer, M.; Banga, J.; Timmer, J.; Fleck, C. Hybrid optimization method with general switching strategy for parameter estimation. *BMC Systems Biology* **2**, 26 (2008).

Banga, J. R. Optimization in computational systems biology. *BMC Systems Biology* **2**, 47 (2008).

Bellman, R. *Dynamic Programming* (Princeton University Press, Princeton, NJ, 1957).

Bellomo, N.; De Angelis, E.; Delitala, M. *From Applied Sciences to Complex Systems, Lecture Notes on Mathematical Modelling* **8** (SIMAI – Società Italiana di Matematica Applicata e Industriale, Roma, Italy, 2008).

Benders, J. F. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik* **4**, 238–252 (1962).

Blum, C.; Roli, A. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys* **35**, 268–308 (2003).

Bonabeau, E.; Dorigo, M.; Theraulaz, G. *Swarm Intelligence: From Natural to Artificial Systems* (Oxford University Press, New York, Oxford, 1999).

Braun, D.; Basu, S.; Weiss, R. Parameter estimation for two synthetic gene networks: A case study. In: *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, March 18–23, 2005, Philadelphia, Pennsylvania, USA (ICASSP 2005).* **5**, 769–772 (2005).

Bridewell, W.; Langley, P.; Todorovski, L.; Džeroski, S. Inductive process modeling. *Machine Learning* **71**, 1–32 (2008).

Büche, D.; Schraudolph, N. N.; Koumoutsakos, P. Accelerating evolutionary algorithms with gaussian process fitness function models. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews* **35**, 183–194 (2005).

Bunch, D. S.; Gay, D. M.; Welsch, R. E. Algorithm 717: Subroutines for maximum likelihood and quasi-likelihood estimation of parameters in nonlinear regression models. *ACM Transactions on Mathematical Software* **19**, 109–130 (1993).

Burnham, K. P.; Anderson, D. R. *Model Selection and Multimodel Inference: A Practical Information-Theoretic Approach* (Springer-Verlag, New York, NY, 2nd ed., 2002).

Cao, H.; Recknagel, F.; Cetin, L. T.; Zhang, B. Process-based simulation library SALMO-OO for lake ecosystems. Part 2: Multi-objective parameter optimization by evolutionary algorithms. *Ecological Informatics* **3**, 181–190 (2008).

Caprara, A.; Fishetti, M. Branch-and-cut algorithms. In: Dell'Amico, M.; Maffioli, F.; Martello, S. (eds.) *Annotated Bibliographies in Combinatorial Optimization.* 45–64 (John Wiley, 1997).

Chen, M.-H.; Shao, Q.-M.; Ibrahim, J. G. *Monte Carlo Methods in Bayesian Computation* (Springer, New York, 2000).

Chou, I.; Voit, E. O. Recent developments in parameter estimation and structure identification of biochemical and genomic systems. *Mathematical Bioscience* **219**, 57–83 (2009).

Cohen, S. D.; Hindmarsh, A. C. CVODE, a stiff/nonstiff ODE solver in C. *Computers in Physics* **10**, 138–143 (1996).

Čerepnalkoski, D.; Tashkova, K.; Todorovski, L.; Atanasova, N.; Džeroski, S. The influence of parameter fitting methods on model structure selection in automated modeling of aquatic ecosystems. *Ecological Modelling* (2011a). Submitted.

Čerepnalkoski, D.; Tashkova, K.; Todorovski, L.; Atanasova, N.; Džeroski, S. Influence of parameter fitting methods on model structure selection in automated modeling of aquatic ecosystems. In: *Book of Abstarcts of the 7th European Conference on Ecological Modelling, 30 May – 2 June 2011, Riva del Garda, Italy (ECEM 2011)*. 49 (2011b).

Dantzing, G. B. Maximization of a linear function variables subject to linear inequalities. In: Koopmans, T. C. (ed.) *Activity Analysis of Production and Allocation*. 339–347 (John Wiley & Sons, New York, 1951).

Darwin, C. *On the Origin of Spieces by Means of Natural Selection* (John Murray, London, 1859).

De Pauw, D. J. W.; De Baets, B. Incorporating model identifiability into equation discovery of ode systems. In: *Proceedings of the GECCO Conference Companion on Genetic and Evolutionary Computation, July 12–16, Atlanta, GA, USA (GECCO 2008)*. 2135–2140 (ACM, New York, NY, USA, 2008).

Dechter, R.; Pearl, J. Generalized best-first search strategies and the optimality of A*. *Journal of the ACM* **32**, 505–536 (1985).

Del Conte-Zerial, P.; Brusch, L.; Rink, J. C.; Collinet, C.; Kalaidzidis, Y.; Zerial, M.; Deutsch, A. Membrane identity and GTPase cascades regulated by toggle and cut-out switches. *Molecular Systems Biology* **4**, 206 (2008).

del Rosario, R. C. H.; Mendoza, E.; Voit, E. O. Challenges in lin-log modelling of glycolysis in lactococcus lactis. *IET Systems Biology* **2**, 136–149 (2008).

Dennis, J. E.; Gay, D. M.; Welsch, R. E. Algorithm 573: NL2SOL–An adaptive nonlinear least-squares algorithm. *ACM Transactions on Mathematical Software* **7**, 369–383 (1981).

Dochain, D.; Vanrolleghem, P. A. *Dynamical modelling and estimation in wastewater treatment processes* (IWA Publishing, London, UK, 2001).

Dorigo, M. *Ottimizzazione, apprendimento automatico, ed algoritmi basati sumetafora naturale*. Ph. D. thesis, Dipartimento di Elettronica, Politecnico di Milano, Milan, Italy (1992).

Dorigo, M.; Stützle, T. *Ant Colony Optimization* (MIT Press, Cambridge, MA, 2004).

Dowd, M.; Mayer, R. A Bayesian approach to the ecosystem inverse problem. *Ecological Modelling* **168**, 39–55 (2003).

Dräger, A.; Kronfeld, M.; Ziller, M. J.; Supper, J.; Planatscher, H.; Magnus, J. B.; Oldiges, M.; Kohlbacher, O.; Zell, A. Modeling metabolic networks in c. glutamicum: A comparison of rate laws in combination with various parameter optimization strategies. *BMC Systems Biology* **3**, 5 (2009).

Dréo, J.; Aumasson, J.-P.; Tfaili, W.; Siarry, P. Adaptive learning search, a new tool to help comprehending metaheuristics. *International Journal on Artificial Intelligence Tools* **16**, 483–505 (2007).

Dréo, J.; Siarry, P. Continuous interacting ant colony algorithm based on dense heterarchy. *Future Generation Computer Systems* **20**, 841–856 (2004).

Dumont, H. J.; Van de Velde, I.; Dumont, S. The dry weight estimate of biomass in a selection of Cladocera, Copepoda and Rotifera from the plankton, periphyton and benthos of continental waters. *Oecologia* **19**, 75–97 (1975).

Džeroski, S.; Todorovski, L. Encoding and using domain knowledge on population dynamics for equation discovery. In: Magnani, L.; Nersessian, N. J.; Pizzi, C. (eds.) *Logical and Computational Aspects of Model-Based Reasoning*, *Applied Logic Series* **25**, 227–247 (Kluwer, Dordrecht, 2002).

Džeroski, S.; Todorovski, L. (eds.). *Computational Discovery of Scientific Knowledge*, *Lecture Notes in Computer Science* **4660** (Springer, Berlin, 2007).

Džeroski, S.; Todorovski, L. Modeling the dynamics of biological networks from time course data. In: Choi, S. (ed.) *Systems Biology for Signaling Networks*, *Systems Biology* **1**, 275–294 (Springer, New York, 2010).

Džeroski, S.; Todorovski, L.; Ljubič, P. Using constraints in discovering dynamics. In: Grieser, G.; Tanaka, Y.; Yamamoto, A. (eds.) *Discovery Science*, *Lecture Notes in Computer Science* **2843**, 297–305 (Springer, Berlin, 2003).

Egea, J. A.; Vazquez, E.; Banga, J. R.; Martí, R. Improved scatter search for the global optimization of computationally expensive dynamic models. *Journal of Global Optimization* **43**, 175–190 (2009).

Eiben, A. E.; Smit, S. K. Parameter tuning for configuring and analyzing evolutionary algorithms. *Swarm and Evolutionary Computation* **1**, 19–31 (2011).

Eiben, A. E.; Smith, J. E. *Introduction to Evolutionary Computing* (Springer-Verlag, Berlin, 2003).

Engelbrecht, A. P. *Fundamentals of Computational Swarm Intelligence* (John Wiley & Sons, Chichester, England, 2005).

Feo, T. A.; Resende, M. G. C. Greedy randomized adaptive search procedures. *Journal of Global Optimization* **6**, 109–133 (1995).

Finck, S.; Hansen, H.; Ros, R.; Auger, A. *Real-parameter black-box optimization benchmarking 2009: Presentation of the noiseless functions.* Tech. Rep. 2009/20 (Research Center PPE, 2009).

Fisher, M. L. An application oriented guide to Lagrangian relaxation. *Interfaces* **15**, 399–404 (1985).

Fogel, D. B. *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence* (IEEE Press, New York, 2000).

Fogel, L. J.; Owens, A. J.; J., W. M. *Artificial Intelligence through Simulated Evolution* (John Wiley & Sons, New York, 1966).

Fonlupt, C.; Robilliard, D.; Preux, P.; Talbi, E.-G. Fitness landscapes and performance of meta-heuristics. In: Voß, S.; Martello, S.; Osman, I. H.; Roucairol, C. (eds.) *Meta-heuristics – Advances and Trends in Local Search Paradigms for Optimization.* 255–266 (Kluwer Academic Press, 1999).

Freedman, D.; Diaconis, P. On the histogram as a density estimator: L2 theory. *Probability Theory and Related Fields* **57**, 453–476 (1981).

Geffen, D.; Findeisen, R.; Schliemann, M.; Allgower, F.; Guay, M. Observability based parameter identifiability for biochemical reaction networks. In: *Proceedings of the American Control Conference, June 11–13, 2008, Seattle, Washington, USA (ACC 2008).* 2130–2135 (2008).

Gershenfeld, N. *The Nature of Mathematical Modeling* (Cambridge University Press, Cambridge, UK, 1999).

Gilboa, Y.; Friedler, E.; Gal, G. Adapting empirical equations to Lake Kinneret data by using three calibration methods. *Ecological Modelling* **220**, 3291–3300 (2009).

Gill, P. E.; Murray, W.; Wright, M. H. *Practical Optimization* (Academic Press, 1981).

Glover, F. Future paths for integer programming and links to artificial intelligence. *Computers and Operation Research* **13**, 533–549 (1986).

Glover, F. Tabu search, Part I. *ORSA Journal on Computing* **1**, 190–206 (1989).

Glover, F. Tabu search, Part II. *ORSA Journal on Computing* **2**, 4–32 (1990).

Glover, F.; Laguna, M.; Martí, R. Fundamentals of scatter search and path relinking. *Control and Cybernetics* **29**, 653–684 (2000).

Goel, G.; Chou, I. C.; Voit, E. O. System estimation from metabolic time-series data. *Bioinformatics* **24**, 2505–2511 (2008).

Gonzalez, O. R.; Küper, C.; Jung, K.; Naval, P. C.; Mendoza, E. Parameter estimation using simulated annealing for s-system models of biochemical networks. *Bioinformatics* **23**, 480–486 (2007).

Grassé, P.-P. La reconstruction du nid et les coordinations inter-individuelles chez *Bellicositermes natalensis* et *Cubitermes* sp. La théorie de la stigmergie: Essai d'interprétation du comportement des termites constructeurs. *Insectes Sociaux* **6**, 41–81 (1959).

Gutenkunst, R. N.; Waterfall, J. J.; Casey, F. P.; Brown, K. S.; Myers, C. R.; Sethna, J. P. Universally sloppy parameter sensitivities in systems biology models. *PLoS Computational Biology* **3**, e189 (2007).

Holland, J. H. *Adaptation in Natural and Artificial Systems* (University of Michigan Press, Ann Arbor, MI, 1975).

Holm, S. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics* **6**, 65–70 (1979).

Hooke, R.; Jeeves, T. A. "direct search" solution of numerical and statistical problems. *Journal of the Association for Computing Machinery* **8**, 212–229 (1961).

Horst, R.; Pardalos, P. M.; Thoai, N. V. *Introduction to Global Optimization* (Kluwer Academic Publishers, Dordrecht, 1995).

Horst, R.; Tuy, H. *Global Optimization: Deterministic Approaches* (Springer, Heidelberg, 3rd ed., 1996).

Jacquez, A.; Greif, P. Numerical parameter identifiability and estimability: Integrating identifiability, estimability, and optimal sampling design. *Mathematical Biosciences* **77**, 201–227 (1985).

Janssen, P. H. M.; Heuberger, P. S. C. Calibration of process-oriented models. *Ecological Modelling* **83**, 55–66 (1995).

Jaqaman, K.; Danuse, G. Linking data to models: Data regression. *Nature Reviews Molecular Cell Biology* **7**, 813–819 (2006).

Jeroslow, R. An introduction to the theory of cutting-planes. In: Hammer, P. L.; Johnson, E. L.; Korte, B. H. (eds.) *Discrete Optimization II*, *Annals of Discrete Mathematics* **5**, 71–95 (Elsevier, 1979).

Joe, S.; Kuo, F. Y. Constructing sobol sequences with better two-dimensional projections. *SIAM Journal on Scientific Computing* **30**, 2635–2654 (2008).

Jones, E.; Parslow, J.; Murray, L. A Bayesian approach to state and parameter estimation in a phytoplankton-zooplankton model. *Australian Meteorological and Oceanographical Journal* **59**, 7–16 (2010).

Jopp, F.; Reuter, H.; Breckling, B. (eds.). *Modelling Complex Ecological Dynamics* (Springer, Berlin, 2011).

Jørgensen, S. E.; Bendoricchio, G. *Fundamentals of Ecological Modelling* (Elsevier Science Ltd, Oxford, UK, 2001).

Joshi, M.; Seidel-Morgenstern, A.; Kremling, A. Exploiting the bootstrap method for quantifying parameter confidence intervals in dynamical systems. *Metabolic Engineering* **8**, 447–455 (2006).

Kadane, J. B.; Lazar, N. A. Methods and criteria for model selection. *Journal of the American Statistical Association* **99**, 279–280 (2004).

Kargupta, H.; Goldberg, D. E. Search, blackbox optimization, and sample complexity. In: Belew, R.; Vose, M. (eds.) *Foundations of Genetic Algorithms*. 291–324 (Morgan Kaufmann, San Mateo, CA, 1996).

Kennedy, J.; Eberhart, R. C. Particle swarm optimization. In: *Proceedings of the IEEE International Conference on Neural Networks, 27 November – 1 December 1995, Perth, Australia*, **4**. 1942–1948 (1995).

Kennedy, J.; Mendes, R. Neighborhood topologies in fully-informed and best-of- neighborhood particle swarms. In: *Proceedings of the IEEE International Workshop on Soft Computing in Industrial Applications, June 23–25, 2003, Binghamton, NY, USA*. 45–50 (2003).

Kirkpatrick, S.; Gellatt, C. D.; Vecchi, M. P. Optimization by simulated annealing. *Science* **220**, 671–680 (1983).

Klee, H.; Allen, R. *Simulation of Dynamic Systems with MATLAB and Simulink* (CRC Press, Inc., 2nd ed., 2011).

Klipp, E.; Herwig, R.; Kowald, A.; Wierling, C.; Lehrach, H. *Systems Biology in Practice: Concepts, Implementation and Application* (Wiley-VCH, Weinheim, 2005).

Kolda, T. G.; Lewis, R. M.; Torczon, V. Optimization by direct search: New perspectives on some classical and modern methods. *SIAM Review* **45**, 385–482 (2003).

Kompare, B.; Džeroski, S.; Karalič, A. Identification of the lake Bled ecosystem with the artificial intelligence tools M5 and FORS. In: *Proceedings of the Fourth International Conference on Water Pollution, June 18–20, 1997, Bled, Slovenia*. 789–798 (Computational Mechanics Publications, Southampton, UK, 1997).

Korošec, P. *Stigmergy as an approach to metaheuristic optimization*. Ph. D. thesis, Jožef Stefan International Postgraduate School, Ljubljana (2006).

Korošec, P.; Tashkova, K.; Šilc, J. The differential ant-stigmergy algorithm for large-scale global optimization. In: *Proceedings of the IEEE Congress on Evolutionary Computation, July 18–23, 2010, Barcelona, Spain (CEC2010)*. 1–8 (IEEE, 2010).

Korošec, P.; Šilc, J. Using stigmergy to solve numerical optimization problems. *Computing and Informatics* **27**, 377–402 (2008).

Korošec, P.; Šilc, J. The continuous differential ant-stigmergy algorithm applied to real-world optimization problems. In: *Proceedings of the IEEE World Congress on Evolutionary Computation, June 5–8, 2011, New Orleans, LA, USA, (CEC2011)*. 1327–1334 (IEEE, 2011).

Korošec, P.; Šilc, J.; Filipič, B. The differential ant-stigmergy algorithm. *Information Sciences* **192**, 82–97 (2012).

Langley, P. *Elements of Machine Learning* (Morgan Kaufmann Publishers Inc., San Francisco, CA, 1996).

Langley, P.; Bradshaw, G.; Zytkow, J. *Scientific Discovery: Computational Explorations of the Creative Processes* (MIT Press, Cambridge, MA, 1987).

Larrañaga, P.; Lozano, J. A. *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation* (Kluwer Academic Publishers, 2002).

Lawler, E. L.; Wood, D. E. Branch-and-bound methods: A survey. *Operation Research* **14**, 699–719 (1966).

Lawrence, N. D.; Sanguinetti, G.; Rattray, M. Modelling transcriptional regulation using gaussian processes. In: Schölkopf, B.; Platt, J.; Hoffman, T. (eds.) *Advances in Neural Information Processing Systems* **19**, 785–792 (MIT Press, Cambridge, MA, 2007).

Lillacci, G.; Khammash, M. Parameter estimation and model selection in computational biology. *PLoS Computational Biology* **6**, e1000696 (2010).

Liu, P. K.; Wang, F. S. Hybrid differential evolution including geometric mean mutation for optimization of biochemical systems. *Journal of Compututer & Chemical Engineering* **41**, 65–72 (2010).

Ljung, L. *System Indentification: Theory for the User* (Prentice Hall, Englewood Cliffs, New Jersey, 1987).

Lourenço, H. R.; Martin, O. C.; Stützle, T. Iterated local search. In: Glover, F.; Kochenberger, G. A. (eds.) *Handbook of Metaheuristics*, *International Series in Operations Research & Management Science* **57**, 321–353 (Kluwer Academic Publishers, Norwell, MA, 2003).

Marino, S.; Voit, E. O. An automated procedure for the extraction of metabolic network information from time series data. *Journal of Bioinformatics and Computational Biology* **4**, 665–691 (2006).

Marsili-Libelli, S. Parameter estimation of ecological models. *Ecological Modelling* **62**, 233–258 (1992).

Marsili-Libelli, S.; Guerrizio, S.; Checchi, N. Confidence regions of estimated parameters for ecological systems. *Ecological Modelling* **165**, 127–146 (2003).

Matear, R. J. Parameter optimization and analysis of ecosystem models using simulated annealing: A case study at Station P. *Journal of Marine Research* **53**, 571–607 (1995).

Mendes, P.; Kell, D. B. Non-linear optimization of biochemical pathways: Applications to metabolic engineering and parameter estimation. *Bioinformatics* **14**, 869–883 (1998).

Miao, H.; Xia, X.; Perelson, A.; Wu, H. On identifiability of nonlinear ode models and applications in viral dynamics. *SIAM Review Society for Industrial and Applied Mathematics* **53**, 3–39 (2011).

Mitchell, T. *Machine Learning* (McGraw Hill, New York, NY, 1997).

Mocenni, C.; Spaacino, E.; Vicino, A.; Zubelli, J. P. Mathematical modelling and parameter estimation of the Serra de Mesa basin. *Mathematical and Computer Modelling* **47**, 765–780 (2008).

Modchang, C.; Triampo, W.; Lenbury, Y. Mathematical modeling and application of genetic algorithm to parameter estimation in signal transduction: Trafficking and promiscuous coupling of g-protein coupled receptors. *Computers in Biology and Medicine* **38**, 574–582 (2008).

Moles, C. G.; Mandes, P.; Banga, J. R. Parameter estimation in biochemical pathways: A comparison of global optimization methods. *Genome Research* **13**, 2467–2474 (2003).

Moscato, P. *On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts: Towards Memetic Algorithms*. Tech. Rep. 826, Caltech Concurrent Computation Program (California Institute of Technology, Pasadena, CA, 1989).

Mulligan, A. E.; Brown, L. C. Genetic algoritms for calibrating water quality models. *Journal of Environmental Engineering* **124**, 202–211 (1998).

Nelder, J. A.; Mead, R. A simplex method for function minimization. *The Computer Journal* **7**, 308–313 (1965).

Nelles, O. *Nonlinear System Identification: From Classical Approaches to Neural Networks and Fuzzy Models* (Springer-Verlag, Berlin, 2001).

Nocedal, J.; Wright, S. J. *Numerical Optimization* (Springer, New York, 1999).

Omlin, M.; Reichert, P. A comparison of techniques for the estimation of model prediction uncertainty. *Ecological Modelling* **115**, 45–59 (1999).

Peckov, A.; Džeroski, S.; Todorovski, L. A minimal description length scheme for polynomial regression. In: Washio, T.; Suzuki, E.; Ting, K.; Inokuchi, A. (eds.) *Advances in Knowledge Discovery and Data Mining*, *Lecture Notes in Computer Science* **5012**, 284–295 (Springer, Berlin, 2008).

Pintér, J. D. *Computational Global Optimization in Nonlinear Systems: An Interactive Tutorial* (Lionheart Publishing, Inc., Marietta, GA, 2001).

Polisetty, P. K.; Voit, E. O.; Gatzk, E. P. Identification of metabolic system parameters using global optimization methods. *Theoretical Biology and Medical Modelling* **3**, 4 (2006).

Press, W. H.; Teukolsky, S. A.; Vetterling, W. T.; Flannery, B. P. *Numerical Recipes* (Cambridge University Press, New York, 2nd ed., 1992).

Price, K.; Storn, R. M.; Lampinens, J. A. *Differential Evolution – A Practical Approach to Global Optimization* (Springer, Berlin Heidelberg, 2005).

Qian, S. S.; Stow, C. A.; Borsuk, M. E. On Monte Carlo methods for Bayesian inference. *Ecological Modelling* **159**, 269–277 (2003).

Rechenberg, I. Evolutionsstrategie – optimierung technischer systeme nach prinzipien der biologischen evolution. Ph. D. thesis reprinted by Fromman-Holzboog (1973).

Reeves, C. R. Landscapes, operators and heuristic search. *Annals of Operations Research* **86**, 473–490 (1999).

Reynolds, C. W. Flocks, herds, and schools: A distributed behavioral model. *ACM Computer Graphics* **21**, 25–34 (1987).

Reynolds, R. G. An introduction to cultural algorithms. In: *Proceedings of the 3rd Annual Conference on Evolutionary Programming, February 24–26, 1994, San Diego, CA, USA*. 131–139 (World Scientific Publishing, River Edge, NJ, 1994).

Rice, J. A. *Mathematical Statistics and Data Analysis* (Duxbury Press, Belmont, CA, 3rd ed., 2007).

Rink, J.; Ghigo, E.; Kalaidzidis, Y.; Zerial, M. Rab conversion as a mechanism of progression from early to late endosomes. *Cell* **122**, 735–749 (2005).

Rismal, M.; Kompare, B.; Rajar, R. Contribution of hydrodynamic and limnological modelling to the sanitation of lake Bled. In: Brebbia, C. A.; Rajar, E. (eds.) *Water Pollution IV*, *WIT Transactions on Ecology and the Environment* **20**, 125–139 (Wessex Institute of Technology, 1997).

Rodriguez-Fernandez, M.; Egea, J. A.; Banga, J. R. Novel metaheuristic for parameter estimation in nonlinear dynamic biological systems. *BMC Bioinformatics* **7**, 483 (2006a).

Rodriguez-Fernandez, M.; Mendes, P.; Banga, J. A hybrid approach for efficient and robust parameter estimation in biochemical pathways. *Biosystems* **83**, 248–265 (2006b).

Rogers, S.; Khanin, R.; Girolami, M. Bayesian model-based inference of transcription factor activity. *BMC Bioinformatics* **8**, S2 (2007).

Russell Finley, J.; Pintér, J. D.; Satish, M. G. Automatic model calibration applying global optimization techniques. *Environmental Modeling and Assesment* **3**, 117–126 (1998).

Samaniego, F. J. *A Comparison of the Bayesian and Frequentist Approaches to Estimation* (Springer, New York, 2010).

Schittkowski, K. *Numerical Data Fitting in Dynamical Systems – A Practical Introduction with Applications and Software* (Kluwer Academic Publishers, Dordrecht, The Netherlands, 2002).

Schwarz, G. Estimating the dimension of a model. *Annals of Statistics* **6**, 461–464 (1978).

Schwefel, H. Numerische optimierung von computer-modellen. Ph. D. thesis reprinted by Birkhäuser (1977).

Sirenko, S. Classification of heuristic methods in combinatorial optimization. *International Journal Information Theories & Applications* **16**, 303–319 (2009).

Socha, K.; Dorigo, M. Ant colony optimization for continuous domains. *European Journal of Operational Research* **185**, 1155–1173 (2008).

Spiser, M. *Introduction to the Theory of Computation* (Course Technology, Boston, MA, 2nd ed., 2005).

Stadler, P. F. Towards a theory of landscapes. *Lecture Notes in Physics* **461**, 78–163 (1995).

Stadler, P. F. Landscapes and their correlation functions. *Journal of Mathematical Chemistry* **20**, 1–45 (1996).

Staudte, R. G.; Sheather, S. J. *Robust Estimation and Testing* (John Wiley and Sons, New York, 1990).

Steele, J. H. Notes on some theoretical problems in production ecology. In: Goldman, C. R. (ed.) *Primary Production in Aquatic Environments*. 393–398 (University of California Press, Berkeley, CA, 1965).

Storn, R. M.; Price, K. *Differential Evolution – a simple and efficient adaptive scheme for global optimization over continuous spaces*. Tech. Rep. TR-95-012 (International Computer Science Institute, Berkeley, CA, 1995).

Storn, R. M.; Price, K. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization* **11**, 341–359 (1997).

Sun, J.; Garibaldi, J. M.; Hodgman, C. Parameter estimation using metaheuristics in systems biology: A comprehensive review. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* **9**, 185–202 (2012).

Taillard, E. D.; Gambardella, L. M.; Gendreau, M.; Potvin, J. Adaptive memory programming: A unified view of metaheuristics. *European Journal of Operational Research* **135**, 1–16 (2001).

Talbi, E. A taxanomy of hybrid metaheuristics. *Journal of Heuristics* **8**, 541–564 (2002).

Talbi, E. *Metaheuristics: From Design to Implementation* (John Wiley and Sons, Hoboken, New Jersey, 2009).

Tashkova, K.; Korošec, P.; Šilc, J. Exploring the parameter space of a search algorithm. In: *Proceedings of the 5th International Conference on Bioinspired Optimization Methods and their Application, May 24–25, 2012, Bohinj, Slovenia, (BIOMA 2012)*. (Jožef Stefan Institute, Ljubljana, Slovenia, 2010b).

Tashkova, K.; Korošec, P.; Šilc, J.; Džeroski, S. Parameter estimation in an endocytosis model with metaheuristic optimization algorithms. In: *Abstracts and Programm of the Workshop Data 2 Dynamics, September 23–25, 2009, Freising, Germany* (2009).

Tashkova, K.; Korošec, P.; Šilc, J.; Todorovski, L.; Džeroski, S. Parameter estimation in an endocytosis model. In: *Proceedings of the 4th International Workshop on Machine Learning in Systems Biology, October 15–16, 2010, Edinburgh, Scotland (MLSB 2010)*. 75–80 (2010a).

Tashkova, K.; Korošec, P.; Šilc, J.; Todorovski, L.; Džeroski, S. Parameter estimation in an endocytosis model using bioinspired optimization algorithms. In: *Proceedings of the 4th International Conference on Bioinspired Optimization Methods and their Application, May 20–21, 2010, Ljubljana, Slovenia, (BIOMA 2010)*. 67–82 (Jožef Stefan Institute, Ljubljana, Slovenia, 2010b).

Tashkova, K.; Korošec, P.; Šilc, J.; Todorovski, L.; Džeroski, S. Parameter estimation with bio-inspired meta-heuristic optimization: Modeling the dynamics of endocytosis. *BMC Systems Biology* **5**, 159 (2011a).

Tashkova, K.; Šilc, J.; Atanasova, N.; Džeroski, S. Parameter identification in a nonlinear dynamic model of an aquatic ecosystem with a metaheuristic optimization approach. In: *Book of Abstarcts of the 7th European Conference on Ecological Modelling, 30 May – 2 June 2011, Riva del Garda, Italy (ECEM 2011)*. 131 (2011b).

Tashkova, K.; Šilc, J.; Atanasova, N.; Džeroski, S. Parameter estimation in a nonlinear dynamic model of an aquatic ecosystem with meta-heuristic optimization. *Ecological Modelling* **226**, 36–61 (2012b).

Todorovski, L.; Bridewell, W.; Shiran, O.; Langley, P. Inducing hierarchical process models in dynamic domains. In: *Proceedings of the 20th National Conference on Artificial Intelligence, July 9–13, 2005 Pittsburgh, Pennsylvania, USA (AAAI 2005)*. **2**, 892–897 (AAAI Press, 2005).

Todorovski, L.; Džeroski, S. Declarative bias in equation discovery. In: *Proceedings of the 14th International Conference on Machine Learning, July 8–12, 1997, Nashville, Tennessee, USA (ICML 1997)*. 376–384 (Morgan Kaufmann Publishers Inc., San Francisco, CA, 1997).

Todorovski, L.; Džeroski, S. Using domain dpecific knowledge for automated modeling. In: R. Berthold, M.; Lenz, H.-J.; Bradley, E.; Kruse, R.; Borgelt, C. (eds.) *Advances in Intelligent Data Analysis V*, *Lecture Notes in Computer Science* **2810**, 48–59 (Springer, Berlin, 2003).

Todorovski, L.; Džeroski, S. Integrating knowledge-driven and data-driven approaches to modeling. *Ecological Modelling* **194**, 3–13 (2006).

Todorovski, L.; Ljubič, P.; Džeroski, S. Inducing polynomial equations for regression. In: Boulicaut, J. F.; Esposito, F.; Giannotti, F.; Pedreschi, D. (eds.) *Machine Learning ECML 2004*, *Lecture Notes in Computer Science* **3201**, 441–452 (Springer, Berlin, 2004).

Tominaga, D.; Koga, N.; Okamoto, M. Efficient numerical optimization algorithm based on genetic algorithm for inverse problem. In: *Proceedings of the Genetic and Evolutionary Computation Conference, July 8–12, 2000, Las Vegas, Nevada, USA, 2000 (GECCO 2000)*. 251–258 (Morgan Kaufmann, 2000).

Toni, T.; Welch, D.; Strelkowa, N.; Ipsen, A.; Stumpf, M. P. H. Approximate Bayesian computation scheme for parameter inference and model selection in dynamical systems. *Journal of the Royal Society Interface* **6**, 187–202 (2009).

Törn, A.; Ali, M.; Viitanen, S. Stochastic global optimization: Problem classes and solution techniques. *Jounral of Global Optimization* **14**, 437–447 (1999).

Velten, K. *Mathematical Modeling and Simulation: Introduction for Scientists and Engineers* (Wiley-VHC, Weinheim, 2009).

Voss, H. U.; Timmer, J.; Kurths, J. Nonlinear dynamical system identification from uncertain and indirect mesuremnts. *International Journal of Bifurcation and Chaos* **14**, 1905–1933 (2004).

Voudouris, C.; Tsang, E. P. K. Guided local search. In: Glover, F.; Kochenberger, G. A. (eds.) *Handbook of Metaheuristics*, *International Series in Operations Research & Management Science* **57**, 185–218 (Kluwer Academic Publishers, Norwell, MA, 2003).

Walter, E.; Pronzato, L. *Identification of Parametric Models from Experimental Data* (Springer, London, Berlin, Heidelberg, 1997).

Weise, T.; Zapf, M.; Chiong, R.; Nebro Urbaneja, A. J. Why is optimization difficult? In: Chiong, R. (ed.) *Nature-Inspired Algorithms for Optimisation*, *Studies in Computational Intelligence* **193**, 1–50 (Springer-Verlag, Berlin, 2009).

Whigham, P. A.; Recknagel, F. Predicting chlorophyll-*a* in freshwater lakes by hybridising process-based models and genetic algorithms. *Ecological Modelling* **146**, 243–251 (2001).

Williams, R. J.; Zisper, D. A learning algorithm for continually running fully recurrent neural networks. *Neural Computation* **1**, 270–280 (1989).

Yue, H.; Brown, M.; Knowles, J.; Wang, H.; Broomhead, D. S.; Kell, D. B. Insights into the behaviour of systems biology models from dynamic sensitivity and identifiability analysis: A case study of an NF-κB signalling pathway. *Molecular Biosystems* **2**, 640–649 (2006).

Zerial, M.; McBride, H. Rab proteins as membrane organizers. *Nature Reviews Molecular Cell Biology* **2**, 107–117 (2001).

Zhang, X.; Srinivasan, R.; Zhao, K.; Liew, M. V. Evaluation of global optimization algorithms for parameter calibration of a computationally intensive hydrologic model. *Hydrological Processes* **23**, 430–441 (2009).

Zuniga, P. C. C. Using the ant colony optimization algorithm in the network inference and parameter estimation of biochemical systems. *Journal of Computational Biology and Bioinformatics Research* **3**, 53–62 (2011).

List of Figures

List of Tables

Appendix 1: Bibliography

1.1 Publications related to this dissertation

1.1.1 Journal papers

Tashkova, K.; Korošec, P.; Šilc, J.; Todorovski, L.; Džeroski, S. Parameter estimation with bio-inspired meta-heuristic optimization: modeling the dynamics of endocytosis. *BMC Systems Biology* **5**, 159 (2011).

Tashkova, K.; Šilc, J.; Atanasova, N.; Džeroski, S. Parameter estimation in a nonlinear dynamic model of an aquatic ecosystem with meta-heuristic optimization. *Ecological Modelling* **226**, 36–61 (2012).

Čerepnalkoski, D.; Tashkova, K.; Todorovski, L.; Atanasova, N.; Džeroski, S. The influence of parameter fitting methods on model structure selection in automated modeling of aquatic ecosystems. *Ecological Modelling*. Submitted.

1.1.2 Conference and workshop abstracts and papers

Tashkova, K.; Korošec, P.; Šilc, J. Exploring the parameter space of a search algorithm. In: *Proceedings of the 5th International Conference on Bioinspired Optimization Methods and their Application, May 24–25, 2012, Bohinj, Slovenia, (BIOMA 2012)*. (Jožef Stefan Institute, Ljubljana, Slovenia, 2012).

Tashkova, K.; Šilc, J.; Atanasova, N.; Džeroski, S. Parameter estimation in a nonlinear dynamic model of an aquatic ecosystem with meta-heuristic optimization approach. In: *Book of Abstarcts of the 7th European Conference on Ecological Modelling, 30 May – 2 June 2011, Riva del Garda, Italy (ECEM 2011)*. 131 (2011).

Čerepnalkoski, D.; Tashkova, K.; Todorovski, L.; Atanasova, N.; Džeroski, S. Influence of parameter fitting methods on model structure selection in automated modeling of aquatic ecosystems. In: *Book of Abstarcts of the 7th European Conference on Ecological Modelling, 30 May – 2 June 2011, Riva del Garda, Italy (ECEM 2011)*. 49 (2011).

Tashkova, K.; Korošec, P.; Šilc, J.; Todorovski, L.; Džeroski, S. Parameter estimation in an endocytosis model. In: *Proceedings of the 4th International Workshop on Machine Learning in Systems Biology, October 15–16, 2010, Edinburgh, Scotland (MLSB 2010)*. 75–80 (2010).

Tashkova, K.; Korošec, P.; Šilc, J. The differential ant-stigmergy algorithm for large-scale global optimization. In: *Proceedings of the IEEE Congress on Evolutionary Computation, July 18–23, 2010, Barcelona, Spain (CEC 2010)*. 1–8 (IEEE, 2010).

Tashkova, K.; Korošec, P.; Šilc, J.; Todorovski, L.; Džeroski, S. Parameter estimation in an endocytosis model using bioinspired optimization algorithms.

In: *Proceedings of the 4th International Conference on Bioinspired Optimization Methods and their Application, May 20–21, 2010, Ljubljana, Slovenia, (BIOMA 2010).* 67–82 (Jožef Stefan Institute, Ljubljana, Slovenia, 2010).

Tashkova, K.; Korošec, P.; Šilc, J.; Džeroski, S. Parameter estimation in an endocytosis model with metaheuristic optimization algorithms. In: *Abstracts and Programm of the Workshop Data 2 Dynamics, September 23–25, 2009, Freising, Germany*, (Helmholtz Zentrum, München, Germany, 2009).

1.2 Other publications

1.2.1 Journal papers

Tashkova, K.; Korošec, P.; Šilc, J. A distributed multilevel ant-colony algorithm for the multi-way graph partitioning. *International Journal of Bio-inspired Computation* **3**, 286–296 (2011).

Tashkova, K.; Korošec, P.; Šilc, J. A distributed multilevel ant-colony approach for finite element mesh decomposition. *Parallel Processing and Applied Mathematics: Part II, Lecture Notes in Computer Science* **6068**, 398–407 (Springer-Verlag, Berlin, 2010).

Tashkova, K.; Korošec, P.; Šilc, J. A distributed multilevel ant colonies approach. *Informatica (Ljublj.)* **32**, 307–317 (2008).

Tashkova, K.; Stojanova, D.; Bohanec, M.; Džeroski, S. A qualitative decision-support model for evaluating researchers. *Informatica (Ljublj.)* **31**, 479–486 (2007).

1.2.2 Conference and workshop abstracts and papers

Tashkova, K.; Korošec, P.; Šilc, J. A distributed multilevel ant-colony approach for finite element mesh decomposition. In: *Book of abstracts of the 8th International Conference on Parallel Processing and Applied Mathematics, September 13–16, 2009, Wrocław, Poland (PPAM 2009).* 107 (Institute of Computer and Information Sciences, Czestoschowa University of Technology, Czestoschowa, Poland, 2009).

Tashkova, K.; Korošec, P.; Šilc, J. A ditributed multilevel ant colonies approach for graph partitioning. In: *Proceedings of the 3rd International Conference on Bioinspired Optimization Methods and their Applications, October 13–14, 2008, Ljubljana, Slovenia (BIOMA 2008).* 41–57 (Jožef Stefan Institute, Ljubljana, Slovenia, 2008).

Tashkova, K.; Mesh partitioning with MACA – a distributed approach. In: *Book of abstracts of the 11th Workshop "Algoritmi po vzorih iz narave", June 17, 2008, Ljubljana, Slovenia, (AVN 2007/2008).* 16 (Jožef Stefan Institute, Ljubljana, Slovenia, 2008).

Čerepnalkoski, D.; Tashkova, K.; Todorovski, L.; Džeroski, S. Inducing process-based models of dynamic systems from multiple data sets. In: *Proceedings of the 2nd International Workshop on the Induction of Process Models at the European Conference on Machine Learning and Principles and Practice of Knowledge*

Discovery in Databases, September 15, 2008, Antwerp, Belgium (IPM 2008). 5–12 (2008).

Čerepnalkoski, D.; Džeroski, S.; Tashkova, K.; Todorovski, L. Learning generic models of dynamic systems. In: *Proceedings of the 10th International Multiconference Information Society, October 8–12, 2007, Ljubljana, Slovenia (IS 2007).* **A**, 186–189 (Jožef Stefan Institute, Ljubljana, Slovenia, 2007).

Stojanova, D.; Tashkova, K.; Bohanec, M.; Džeroski, S. A qualitative decision-support model for evaluating researchers. In: *Proceedings of the 10th International Multiconference Information Society, October 8–12, 2007, Ljubljana, Slovenia (IS 2007).* **A**, 60–63 (Jožef Stefan Institute, Ljubljana, Slovenia, 2007).

Stojanova, D.; Panov, P.; Kobler, A.; Džeroski, S.; Tashkova, K. Learning to predict forest fires with different data mining techniques. In: *Proceedings of the 9th International Multiconference Information Society, October 9–14, 2006, Ljubljana, Slovenia (IS 2006).* **A**, 255–258 (Jožef Stefan Institute, Ljubljana, Slovenia, 2006).

Tashkova, K.; Panov, P.; Kobler, A.; Džeroski, S.; Stojanova, D. Predicting forest stand properties from satellite images with different data mining techniques. In: *Proceedings of the 9th International Multiconference Information Society, October 9–14, 2006, Ljubljana, Slovenia (IS 2006).* **A**, 259–262 (Jožef Stefan Institute, Ljubljana, Slovenia, 2006).

Appendix 2: Biography

Katerina Tashkova was born on September 27, 1981 in Skopje, Macedonia. She attended elementary and secondary (gymnasium) school in Skopje. After finishing gymnasium majoring in natural sciences and mathematics, in 2000 she started her studies at the Faculty of Electrical Engineering, University "Ss Cyril and Methodius", Skopje. She was enrolled in a 9 semester BSc program in the area of Computer Engineering, Information Science and Automatics, which she successfully finished in 2005 as the best graduated student of the class 2000–2005. During the secondary and undergraduate study she held a state scholarship for talented students, awarded by the Ministry of Education and Science of Macedonia. As a best graduated student, she also received the Engineers Ring award in 2007, awarded by the Engineering Institution of Macedonia. During her studies she successfully participated at several math, physics, history, and language knowledge (Macedonian and Latin) competitions.

In the period 2005–2006 she was working as a junior teaching assistant at the Faculty of Information Technology, European University, Skopje, Macedonia. In the fall of 2006, she enrolled in the PhD program entitled "New Media and e-Science" under supervision of Professor Dr. Sašo Džeroski at the Jožef Stefan International Postgraduate School, Ljubljana, Slovenia. From January 2007 to July 2011 she was working as research assistant at the Computer Systems Department at the Jožef Stefan Institute, Ljubljana, Slovenia under supervision of Assistant Professor Dr. Jurij Šilc. Since July 2011 she holds a researcher position at NELA, Železniki, Slovenia, a development center for electro-industry. During her PhD studies, she received a three-months scholarship by the German Academic Exchange Service for student internship at the Chair for Computation in Engineering, Technical University of Munich, Munich. She also collaborated on the EU funded project PHAGOSYS (Systems biology of phagosome formation and maturation – modulation by intracellular pathogens).

Her research is in the field of parameter identification of nonlinear dynamic systems, combinatorial and numerical meta-heuristic optimization, parallel computing and includes study, development and application of different optimization algorithms. The application is mainly focused on case studies in ecological modeling and computational biology. She has published (and submitted) her work in several peer-reviewed journals as well as presented it at several international conferences and workshops in the areas of data mining, optimization, parallel computation, ecological modeling and computational biology. Related to her current employment, the research is focused on building models for predicting operational life-time of vacuum cleaner motors.