Modeling of the Physiology of *D. hansenii* Using
Population-Based Search Methods for Parameter
Estimation

DULCE ISABEL VIEGAS CALÇADA

Dissertation for obtaining the degree of Master of Science in

BIOMEDICAL ENGINEERING

Examing Board

President: Prof.ª Patrícia Figueiredo
Advisor: Prof. Agostinho Cláudio da Rosa
Advisor: Prof. Domingos Manuel Pinto Henrique
Specialist: Prof. Vítor V. Lopes
Specialist: Prof. Luís C. Duarte

October 2009
Acknowledgements

Firstly, I would like to thank Prof. Vítor Lopes and Prof. Agostinho Rosa for the opportunity to develop this work. A special thanks to Prof. Vítor Lopes, for his relentless optimism and resourcefulness, for his ideas and for his advice in the writing of this thesis. To Prof. Agostinho Rosa for his constructive criticism regarding the presentation of my work and organization of this thesis. To both, for their motivation and patience with my irregular work schedule.

I would also like to thank all the people at LaSEEB for attending my presentations, offering me a more detached perspective of my work, and contributing with pertinent remarks and insightful suggestions. A special thanks to Nuno Fachada, for his help with uncooperative computational machines, to Alexandre Calapez, for his interest in the mathematical modeling and optimization portions of this work, and to Rogério Largo, for his suggestions concerning genetic algorithms.

I would like to express my deepest gratitude to Prof. Luís Duarte, for agreeing to the use of his experimental data in this thesis. Most of all, I thank him for his enthusiastic collaboration in this work, as his extensive knowledge of the physiology of D. hansenii and promptness in sharing it led to great progresses in the modeling efforts and in the biological interpretation of the results.

To Prof. Domingos Henrique, for accepting to be my co-advisor, and for the patience and kindness with which he assumed that role. Most of all, for unknowingly having inspired me to pursue a thesis in this area, by having introduced similar topics in an earlier course.

To my brother, Nuno, who bore with me in the most stressful times of this work, and to the remaining of my family, for their unconditional support. A special thanks to my friends Andreia and Nuno, for the uplifting conversations and so much more.

Finally, to Gustavo. I acknowledge the time and effort he so generously committed to the review of this thesis’ first manuscript and, more importantly, to our frequent intellectual debates. His clarity of thought and critical thinking skills spurred my own, helping me to deliver a sounder and richer work. I also thank him for his refined patience in my (not so infrequent) absent-minded days, for his listening abilities and emotional support. And for everything else.
Abstract

This work tackled the problem of finding a mathematical model capable of describing the physiology of the yeast *Debaryomyces Hansenii* (*D. hansenii*), under certain conditions. Nine alternative physiological models were developed and formulated as systems of nonlinear ordinary differential equations, describing the temporal rate of change of the concentrations of eight substances of interest, as a function of a number of parameters (26 to 31, depending on the model).

The estimation of the parameters of these models was accomplished by fitting them to experimental time courses of the concentrations of the aforementioned substances. The resulting optimization problem aimed at minimizing, with certain constraints, an objective function (OF) consisting of the sum of squared differences between the models’ predictions and the experimental data. Given the limited success of classical deterministic methods in solving this complex problem, two population-based stochastic algorithms were proposed to solve it: a genetic algorithm (GA) and particle swarm optimization (PSO). The performance of these two algorithms was tested on one model and one data group, using different settings for their operating parameters. It was found that the PSO, with settings i) resulting in a convergent behavior, with oscillation or zigzagging, and ii) privileging local search to global search, was superior to the GA.

The PSO was then used to estimate the parameters of all the models with the two available data groups, and the results showed that the best OF values obtained for all the models except one (model 2A) were statistically equivalent to the variability of the experimental data. As for the concentrations predicted by the models, they all closely agreed with their experimental counterparts (goodness-of-fit coefficients close to 1), with the exception of glycerol and, in some instances, ethanol.

Model 1D was found to be the best model, considering both the OF values obtained for the two data groups and its reduced number of parameters. It was possible to provide an interpretation for the variation of its parameters from one data group to the other, and formulate a testable hypothesis for the phenomena that took place, in terms of the different physiological pathways leading to biomass growth in the two data groups.
Keywords
Resumo

Este trabalho abordou o problema de obter um modelo matemático capaz de descrever a fisiologia da levedura *Debaryomyces hansenii* (*D. hansenii*), em certas condições. Foram desenvolvidos nove modelos alternativos, cada qual formulado como um sistema de equações diferenciais ordinárias não lineares. Este sistema descreve a taxa de variação temporal das concentrações de oito substâncias de interesse, em função de um certo número de parâmetros (26 a 31, dependendo do modelo).

A estimação dos parâmetros dos modelos foi feita ajustando-os a perfis temporais experimentais das concentrações das ditas substâncias. No problema de optimização resultante pretendeu-se minimizar, com certos constrangimentos, uma função objetivo (OF) consistindo da soma dos quadrados das diferenças entre as previsões dos modelos e as observações experimentais. Dado o sucesso limitado dos métodos clássicos (determinísticos) na resolução deste problema complexo, foram propostos dois algoritmos estocásticos e baseados em populações de potenciais soluções para o problema: um algoritmo genético (GA) e a optimização por enxame de partículas (PSO). O desempenho destes dois algoritmos foi testado para um modelo e um grupo de dados, variando os parâmetros operacionais de cada um, a fim de descobrir qual o melhor algoritmo para a resolução do problema. Concluiu-se que o PSO é superior ao GA, usando parâmetros operacionais que levem o algoritmo a convergir com oscilações ou ziguezague, e que privilegiem uma procura local a uma procura global.

O PSO foi então usado para estimar os parâmetros de todos os modelos, nos dois grupos de dados disponíveis. Os melhores valores da OF encontrados para todos os modelos excepto um (modelo 2A) revelaram-se estatisticamente equivalentes à variabilidade dos dados experimentais. Em relação às concentrações previstas pelos modelos, foi obtido um bom ajuste (coeficientes de determinação $R^2 > 1$) para os perfis temporais de todas as substâncias, com exceção do glicerol e também, em alguns casos, do etanol.

O modelo 1D foi seleccionado como melhor modelo, considerando tanto os valores da OF que obteve para os dois grupos de dados, como o seu reduzido número de parâmetros. Foi possível fornecer uma interpretação para a variação dos valores dos parâmetros de um grupo de dados para o outro, tendo-se formulado uma hipótese experimentalmente testável acerca das diferentes vias fisiológicas que levam ao crescimento da biomassa nos dois grupos.
**Palavras-chave**

Modelo Fisiológico, *Debaryomyces hansenii*, Cinética do Crescimento Microbiano, Equações Diferenciais Ordinárias, Estimação de Parâmetros, Optimização Não Linear, Algoritmos Genéticos, Optimização por Enxame de Partículas.
Contents

Acknowledgements ........................................................................................................... i
Abstract .......................................................................................................................... iii
Keywords ........................................................................................................................ iv
Resumo ............................................................................................................................ v
Palavras-chave ................................................................................................................ vi
Contents ........................................................................................................................ vii
List of Figures .................................................................................................................. ix
List of Tables ................................................................................................................... x
List of Abbreviations ......................................................................................................... xi
1 Introduction ................................................................................................................... 1
  1.1 Motivation: The Field of Industrial Biotechnology ................................................... 1
    1.1.1 The Biorefinery Concept .................................................................................... 2
    1.1.2 Relevance of Pentose-Fermenting Yeasts in an Industrial Framework .......... 3
  1.2 Background ............................................................................................................ 3
    1.2.1 Mathematical Modeling of Microbial Growth Kinetics ................................. 3
    1.2.2 Methods for Solving Nonlinear Optimization Problems ......................... 6
  1.3 Aims ....................................................................................................................... 9
  1.4 Outline of the Thesis ............................................................................................. 9
2 Methodology .................................................................................................................. 10
3 Mathematical Modeling of D. hansenii Physiology ..................................................... 13
  3.1 Preliminary Considerations .................................................................................... 13
    3.1.1 Experimental Conditions and Range of Validity of the Model .................... 13
    3.1.2 The Metabolism of Glucose, Xylose and Arabinose in D. hansenii ............ 14
  3.2 Structure of a Physiological Network for D. hansenii ........................................ 16
  3.3 The Kinetic Model as a System of Ordinary Differential Equations .................... 19
    3.3.1 Derivation of the ODE System from the Model’s Network ..................... 20
    3.3.2 Numerical Integration of the ODE System ................................................. 23
4 Parameter Estimation as an Optimization Problem ................................................... 24
  4.1 Objective Function .............................................................................................. 24
    4.1.1 Nonlinear Least-Squares .............................................................................. 24
    4.1.2 Constraints .................................................................................................. 25
    4.1.3 Safeguard for Numerical Integration Errors .............................................. 28
    4.1.4 Outline of the Objective Function ............................................................ 29
  4.2 Analysis of Experimental Data ............................................................................. 29
    4.2.1 A Measure for Data Spread ....................................................................... 30
    4.2.2 Bootstrapping the SSPD .......................................................................... 30
5 The Optimization Algorithms ..................................................................................... 32
  5.1 The Genetic Algorithm .......................................................................................... 32
    5.1.1 Fixed Settings .............................................................................................. 34
    5.1.2 Settings subject to Tuning ........................................................................... 37
  5.2 Particle Swarm Optimization ................................................................................. 40
    5.2.1 Fixed Settings .............................................................................................. 41
    5.2.2 Settings subject to Tuning ........................................................................... 42
  5.3 Tuning: Optimizing the Performance of the Optimization Algorithms ................. 44
    5.3.1 Tuning Methodology .................................................................................... 45
    5.3.2 Evaluation of Algorithm Performance ....................................................... 46
6 Results and Discussion .............................................................................................. 47
  6.1 Analysis of Experimental Variability ..................................................................... 47
  6.2 Analysis of the Tuning of the Algorithms ............................................................. 47
    6.2.1 Tuning of the GA ......................................................................................... 48
    6.2.2 Tuning of the PSO ....................................................................................... 49
    6.2.3 Comparison between GA and PSO ............................................................. 51
  6.3 Parameter Estimation Results ................................................................................ 51
    6.3.1 Analysis of PSO Performance ....................................................................... 52
    6.3.2 Analysis of the Fit ....................................................................................... 54
    6.3.3 Analysis of the Parameters of the Best Model .............................................. 58
7 Conclusions ............................................................................................................... 66
References ..................................................................................................................... 68
Appendix A Experimental Data ...................................................................................... 73
Appendix B Models and Parameters ............................................................................. 76
Appendix C Predicted Time Courses of Concentrations .............................................. 82
List of Figures

Figure 1-1. Basic components and processes of a biorefinery (taken from [4]) ................................................. 2
Figure 2-1. Block diagram showing the strategy followed in this work................................................................. 11
Figure 3-1. Simplified pathway of glucose utilization by D. hansenii ................................................................. 15
Figure 3-2. Simplified pathways of xylose and arabinose utilization by D. hansenii ........................................ 16
Figure 3-3. Models 1A, 1B, 1C and 1D. .................................................................................................................. 17
Figure 3-4. Models 2A, 2B, 2C. ............................................................................................................................. 18
Figure 3-5. Models 3A and 3B. ............................................................................................................................. 19
Figure 3-6. Diagram for the network of model 3B, with reactions and inhibitions labeled. The black arrows (solid as well as dashed) represent reactions, and the red lines ending in circles represent inhibitions. ................................................................................................................................................. 20
Figure 5-1. Flow chart for the genetic algorithm................................................................................................. 33
Figure 5-2. Flow chart for the particle swarm optimization algorithm............................................................... 40
Figure 6-1. Histograms for SSPD of experimental data in group I (left) and group II (right)......................... 47
Figure 6-2. Histograms of SSPD in data groups I and II, with each model’s best OF value superimposed. Also superimposed is the mean SSPD for each data group and a tail-symmetric 95% confidence interval for the values of the SSPD, built from the 2.5\textsuperscript{th}-percentile and 97.5\textsuperscript{th}-percentile of the data. ................................................................................................................................. 54
Figure 6-3. Predicted ethanol concentrations for data group II by model 2C (left) and model 2A (right). .................................................................................................................................................. 56
Figure 6-4. Biomass concentrations predicted by model 1B for data groups I (left) and II (right) .......... 57
Figure 6-5. Parameters obtained for model 1D in its best run, for data group I (a) and data group II (b). .................................................................................................................................................. 59
Figure 6-6. Logarithmic plot of the ratio of the parameters in group II to the parameters in group I. The axis of abscissas groups the parameters according to the reaction to which they refer, and a different color and border style is used for each type of parameter. The values of the ratios are shown in the accompanying table ...................................................................................................................................... 60
Figure 6-7. Time courses of arabinose concentration, $Ara(t)$, of (scaled) arabinose rate of consumption, $dAra/dt$, and of the corresponding (scaled) inhibition factor, for replicate B2E1 in data group I (labeled GI) and B2E10 in data group II (labeled GII). For clarity reasons, the ordinate axis is not labeled and the units of the plotted values are given in the legend. The scaling factors are the same for both data groups .................................................................................................................................. 62
Figure 6-8. Time courses of ethanol concentration, $EtOH(t)$, and of (scaled) ethanol specific rate of consumption, $\mu_{\gamma}$, for replicates B2E1 in data group I (labeled GI) and B2E10 in data group II (labeled GII). ........................................................................................................................................ 63
Figure 6-9. Specific rate of biomass production over time, plotted along with the specific rates of the reactions contributing to its value. The replicates used, were, for group I (a), B2E1, and for group II (b), B2E10. ........................................................................................................................................ 64
List of Tables

Table 3-1. Range of experimental initial concentrations of the eight species considered in this work. 14
Table 4-1. Empirical formulas and molar masses of the chemical species considered in this work..... 26
Table 4-2. Stoichiometric coefficients for the reactions in the models. ............................................. 26
Table 4-3. Upper bounds for the yields of the reactions in the models (in g/g). ................................. 26
Table 5-1. Summary of settings for the GA’s tuning simulations. ...................................................... 39
Table 5-2. Summary of settings for the PSO’s tuning simulations. .................................................. 44
Table 6-1. Mean value and standard deviation for the experimental SSPD. ..................................... 47
Table 6-2. Summary of the results of the GA’s tuning simulations (the best tuning case is highlighted). ........................................................................................................................................... 48
Table 6-3. Summary of the results of the PSO’s tuning simulations (the best tuning case is highlighted). ........................................................................................................................................... 49
Table 6-4. Summary of parameter estimation results for data groups I and II.................................. 52
Table 6-5. Mean values of each model’s best runs (as defined in the text) for the data in group I, group II and total. The most successful model is highlighted. ................................................................. 53
Table 6-6. R² goodness-of-fit coefficients for the concentrations of each species, obtained for data group I, with the best run of each model. ........................................................................................................................................ 55
Table 6-7. R² goodness-of-fit coefficients for the concentrations of each species, obtained for data group II, with the best run of each model. ........................................................................................................................................ 55
Table 6-8. Best OF value found for each model in the two data groups and their sum. The number of parameters of the models is also shown and the best model is highlighted. ............................................. 58
List of Abbreviations

*D. hansenii*  *Debaryomyces hansenii*

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>Biomass</td>
</tr>
<tr>
<td>Glc</td>
<td>Glucose</td>
</tr>
<tr>
<td>GlyOH</td>
<td>Glycerol</td>
</tr>
<tr>
<td>EtOH</td>
<td>Ethanol</td>
</tr>
<tr>
<td>Xyl</td>
<td>Xylose</td>
</tr>
<tr>
<td>XOH</td>
<td>Xylitol</td>
</tr>
<tr>
<td>Ara</td>
<td>Arabinose</td>
</tr>
<tr>
<td>ArOH</td>
<td>Arabinol or Arabinitol</td>
</tr>
<tr>
<td>ODE</td>
<td>Ordinary Differential Equation</td>
</tr>
<tr>
<td>SSPD</td>
<td>Sum of Squared Pairwise Differences</td>
</tr>
<tr>
<td>PSO</td>
<td>Particle Swarm Optimization</td>
</tr>
<tr>
<td>GA</td>
<td>Genetic Algorithm</td>
</tr>
<tr>
<td>RWS</td>
<td>Roulette Wheel Selection</td>
</tr>
<tr>
<td>SUS</td>
<td>Stochastic Universal Sampling</td>
</tr>
</tbody>
</table>
1 Introduction

This work deals with the problem of finding a mathematical model capable of describing the physiological behavior of the yeast *Debaryomyces hansenii* (*D. hansenii*), under certain conditions. The motivation for such an endeavor is discussed in section 1.1 below, where the relevance of the work developed in this thesis is explained.

The estimation of the parameters of the mathematical model constitutes an optimization problem, in which one tries to obtain the optimal fit between the available experimental data and the model’s predictions. Because the models developed here are (as it is usual for mathematical models describing biological phenomena) very complex and highly nonlinear, classical optimization techniques have limited success in solving them. Section 1.2 will provide the necessary background for this work, discussing both the mathematical modeling of microbial growth and the optimization techniques that may be used for parameter estimation. Among the numerous alternatives to classical optimization methods are the two algorithms explored in this thesis, the genetic algorithm and particle swarm optimization, which will also be introduced in this section.

In section 1.3 the aims of this work are stated, and an outline of the thesis is given in section 1.4.

1.1 Motivation: The Field of Industrial Biotechnology

The field of Industrial Biotechnology deals with the design and development of methodologies and equipment for the industrial production of goods such as fuels, pharmaceuticals, and other chemicals, from and with the aid of biological materials. In particular, Industrial Biotechnology entails the industrial use of (genetically modified or not) microorganisms, such as yeasts or bacteria, to manufacture certain products, which if obtained through chemical processes, would be too costly, time-consuming and/or unpractical. Examples of Industrial Biotechnology applications range from the long-standing usage of baker’s yeast (*Saccharomyces cerevisiae*) in baking and fermentation of alcoholic beverages, to the more recent employment of genetically engineered bacteria in the production of therapeutic proteins, such as insulin or antibodies.

Although a great deal of knowledge has been accumulated on the physiology and metabolic pathways of the most commonly used microorganisms (e.g. the yeast *S. cerevisiae* and the bacterium *E. coli* constitute model organisms in genetic engineering and microbiology), the quest for increased industrial productivity and profitability oftentimes entails the experimentation with new, promise-showing, microorganisms. It then becomes capital to properly enlighten the physiology of the organism employed.

An additional problem in Industrial Biotechnology is that, in the majority of cases, it is not cost-effective to feed the microorganisms with a single purified substrate, and instead a naturally occurring mixture of substances is used, with some of the substances being the intended substrates and others either having no effect on, or, more importantly, hindering the performance of the organism in metabolizing the intended ones. The importance of understanding the physiology of industrially used microorganisms also reflects on the product end, since 1) it is frequently necessary, during production, to remove undesirable metabolic byproducts from the growth medium, and thus one should be able to pinpoint these undesirable products, and 2) the recovery process targets one or several particular
substances, and an incomplete physiological knowledge of the microorganism could lead to the dismissal of possibly important products, which might make the process more economically attractive.

The importance of these microorganisms resides, from the industrial point of view, in their ability to degrade (i.e. metabolize), while growing, a variety of organic compounds, originating desirable products. One very promising approach to the industrial application of microorganisms is the so-called biorefinery [1].

1.1.1 The Biorefinery Concept

Currently at the early stages of its development, a “biorefinery” is “a facility (including equipment and processes) that: a) converts renewable biomass into biofuels and biobased products; and b) may produce electricity.” [2]. Biorefineries deal with the conversion of a wide spectrum of biomass - organic matter available on a renewable basis (e.g. wood and wood waste) - into power, fuels, valuable chemicals and other biomaterials, much like the long-established petroleum refinery turns crude oil into more useful petroleum products.

Moreover, as Briens [3] explains, “A detailed review of the many processes that can convert biomass into fuels and chemicals shows that no individual process is without drawbacks. As a result, it is recommended that a biorefinery is the best solution to combine and integrate various processes to maximize economic and environmental benefits, while minimizing waste and pollution.”. Therefore, a key feature of biorefineries is that, by producing multiple products, they are able to take advantage of the various components of biomass, thus 1) maximizing the value derived from biomass feedstock, and also 2) producing (surplus) energy. A schematic depiction of the processes that a biorefinery encompasses is shown in Figure 1-1 below.

Microorganisms such as yeasts and bacteria (also designated biomass themselves) may prove to be the workers of choice for raw biomass utilization in biorefineries, with the ability of efficiently utilizing the most common biomass materials, being regarded, among other factors, as a desirable characteristic, when choosing the microorganism to perform the process.
One particularly attractive type of biorefinery is the ligno-cellulosic biorefinery, which uses ligno-cellulosic materials as renewable biomass. Ligno-cellulosic materials are plant biomass that is composed of cellulose, hemicellulose, and lignin. They are naturally abundant and inexpensive, representing more than 70% of the total available plant biomass [5], and enable a great variety of products to be obtained. Besides containing high amounts of glucose, these materials are also extremely rich in pentoses (five carbon sugars), and thus, their industrial processing may benefit from the utilization of microorganisms which are able to naturally ferment these sugars.

1.1.2 Relevance of Pentose-Fermenting Yeasts in an Industrial Framework

This is the case of the yeast *D. hansenii*, which, in addition to glucose, is capable of fermenting the pentoses which are most abundant in ligno-cellulosic materials, xylose and arabinose (together making for up to 25% of ligno-cellulosic dry mass [1]). This yeast is capable of producing high yields of xylitol, a polyol with many applications in the food industry (e.g. in chewing gums, candy, soft drinks), in oral hygiene products, and as a sugar substitute for diabetic patients [6]. Xylitol may constitute an alternative added-value product to ethanol (which is emerging as a commercial biofuel), for the valorization of pentose sugars.

Besides being able to ferment pentoses, *D. hansenii* also possesses several other physiological features which are desirable from an Industrial Biotechnology point of view, namely: 1) it is osmotolerant, enduring salt concentrations which are deadly to a great number of other microorganisms, and 2) it is chemotolerant, i.e., it is able to grow in very high concentrations of substrates, and can metabolize ethanol, which for other species is effectively toxic, and must be regularly removed from the industrial reactor [7]. The employment of *D. hansenii* in the context of ligno-cellulosic biorefineries thus results in the economical valorization of ligno-cellulosic materials, and may soon become a competitive alternative to the chemical synthesis of xylitol.

1.2 Background

1.2.1 Mathematical Modeling of Microbial Growth Kinetics

Microbial growth kinetics is, by definition [8], the relation between the specific growth rate of a microbial population and the concentrations of the substrates available to that population. Kinetic models for microbial growth may be of several types. A possible classification system for these models is based on whether or not the model accounts for the existence of structure, both at the cell level and at the population level. According to Nielsen [9], structure at the cell level means that individual intracellular reactions are taken into account and that biomass is structured into two or more variables, while structure at the population level means that population segregation, i.e. the fact that not all cells are in the same state, is taken into account. A simpler model will not consider the existence of structure on either level, while a more complex model will.

The modeling efforts undertaken in this work belong to the class of unstructured and unsegregated models. They are designated physiological models because they are meant to capture the physiological behavior of the microorganism with a high level of abstraction from the physical and biochemical phenomena that originate that behavior. The mathematical formulation of this type of model will be based on the well known and widely used Monod equation for microbial growth kinetics.
1.2.1.1 The Monod Model for Microbial Growth Kinetics

This equation was proposed by Jacques Monod as early as 1949, and it relates the growth rate, $\mu$, of biomass ($X$) to the concentration of a single growth-limiting substrate ($S$), via a hyperbolic expression. The Monod equation contains two parameters: $\mu_{\text{max}}$, the maximum specific growth rate of the microbial population, and $K_S$, the half-saturation constant, whose inverse is thought to be indicative of the affinity of the microorganism for that particular substrate\(^1\). Mathematically:

$$\frac{dX}{dt} = \mu X \quad \text{with} \quad \mu = \frac{\mu_{\text{max}} S}{S + K_S} \quad (1.1)$$

Despite its similarities to the Michaelis-Menten model for enzyme kinetics, the Monod model is an empirical law, not a theoretically derived result. For both models, the relationship between the specific rate $\mu$ and the substrate concentration $S$ is described by a saturation-type curve. This means that, at high substrate concentrations, the microbial population will grow at a maximum rate, $\mu_{\text{max}}$, independent of substrate concentration. Because the units of $K_S$ and $S$ are the same (any units of concentration), the units of $\mu$ and $\mu_{\text{max}}$ are also the same – the inverse of time.

As for substrate utilization kinetics, noting that the rate of substrate consumption may be written as

$$\frac{dS}{dt} = \frac{dS}{dX} \frac{dX}{dt} \quad (1.2)$$

one may define

$$\left| \frac{dX}{dS} \right| = \text{constant} \equiv Y_{X/S} \quad (1.3)$$

with $Y_{X/S}$, the yield coefficient (usually expressed in g/g), being a measure of the conversion efficiency of substrate $S$ into cell material. The total rate of change of substrate per unit time then becomes

$$\frac{dS}{dt} = -\frac{1}{Y_{X/S}} \mu X = -\frac{1}{Y_{X/S}} \frac{\mu_{\text{max}} S}{S + K_S} X \quad (1.4)$$

which may also be written as

$$\frac{dS}{dt} = -q_S X = -q_{S,\text{max}} \frac{S}{S + K_S} X \quad \text{with} \quad q_S = \frac{\mu}{Y_{X/S}} \quad (1.5)$$

where $q_S$ is the specific rate of substrate utilization, i.e., the amount of substrate consumed (in units of e.g. mass, concentration), per unit time and unit amount of biomass.

Besides the Monod model, other models exist for the kinetics of microbial cell growth [9]. Most are simply extensions of it, accounting, for example, for the

i) fact that cells may need substrate even when they are not growing (that is, they need a minimum amount of substrate for energy maintenance purposes) [8], or the

\(^1\) Some authors hold that the affinity of the microorganism for a given substrate should instead be represented by the ratio $\mu_{\text{max}}/K_S$, termed specific affinity, since both a significant amount of experimental data [8] and some theoretical derivations of the Monod equation [10] predict an increase of $K_S$ with $\mu_{\text{max}}$, with their ratio remaining approximately constant.
ii) inhibitory effect that a high concentration of biomass may have on its own growth rate (Contois model).

Some generalizations of the Monod model have also been developed [11], which, for certain conditions, reduce to the Monod equation [8]. All the aforementioned models are at least as complex as, and, most often, more complex than the Monod model, that is, they require more parameters for the description of microbial growth kinetics.

It is important to notice that the Monod model only describes microbial growth kinetics in the presence of a limiting substrate, that is, when all other substrates that the cell may require to grow are present in excess. It may not be directly applied when one wishes to model, for example, the kinetics of cell growth on multiple substrates (limiting or not), or the kinetics of the formation of a given product during cell growth. The extension of the model to cover these cases is not straightforward and the success of the several existing alternatives [12;13] seems to depend on the particular set of experimental data on which they are tested [12].

Equations (1.1) and (1.4) in the Monod model constitute a nonlinear system of ordinary differential equations, with two variables ($X$ and $S$) and three parameters ($\mu_{\text{max}}$, $K_S$ and $Y_{X/S}$). These parameters may be estimated by fitting the model to available experimental data.

1.2.1.2 Fitting the Model to Experimental Data

The estimation of the parameters $\mu_{\text{max}}$, $K_S$ and $Y_{X/S}$ is accomplished by fitting the Monod model to experimental data, using either its derivative or integrated form, with the latter usually yielding better results than the former [14]. Unfortunately, there is no closed-form exact analytical expression for the solution of the system (that is, the equations cannot be analytically integrated in order to obtain the concentrations of biomass and substrate), and, thus, procedures which use the integrated form to estimate the parameters of the model have to rely on numerical integration.

The fitting of experimental data to the Monod model presents several challenges, which derive mainly from its nonlinearity and from the inadequacy of the experimental data used to fit it. Aware of the limited quality of his data, Monod himself stated that “several different mathematical formulations could be made to fit the data. But it is convenient and logical to adopt a hyperbolic equation” [8]. In fact, several results for the parameters of a model referring to the same microorganism are inconsistent, with the value of $K_S$ often varying with the value of $\mu_{\text{max}}$ [8], and the specific growth rate depending on the ratio of substrate to biomass concentration [15].

Between 1949 and the present day, the optimization of the design of experiments which are to produce data to fit the Monod model (and others), has been the subject of much research [16-19]. Though optimal experimental designs for the fitting of the Monod model have been derived, it becomes much harder to obtain such a derivation when dealing with models that are more complex, as is the case of the models developed in this work. Furthermore, although linearized versions of the Monod model can be derived, in order to simplify the parameter estimation procedure (though at the cost of losing quality in the estimates obtained) [14;20], linearization is not possible for more complex models [21].

Another difficulty of trying to fit experimental data to very complex models (i.e. models with many variables and parameters, related in a nonlinear fashion), is that the models are often over-
parameterized with respect to the available experimental data. This is a very common problem when modeling biological phenomena at large, because 1) the available models are generally complex, and 2) the amount of experimental data is limited, due to the measurements being expensive and/or disruptive to the experiment. Over-parameterization renders the fitting exercise both computationally heavy and unreliable, since the set of parameters which predict the experimental data are most often not unique [22].

Fitting the predictions of a mathematical model to experimental data entails minimizing the difference (in a broad sense) between the former and the latter, constituting an optimization problem. When the model is linear in the parameters, the fitting is achieved by ordinary linear regression, minimizing the sum of squared residuals between the experimental data and the predictions of the model (least squares criterion). However, when the model is nonlinear in the parameters, a nonlinear least squares problem results. The solution for a problem of this kind can’t be found in a single step, and iterative procedures have to be used instead.

1.2.2 Methods for Solving Nonlinear Optimization Problems

The goal in an optimization problem is to find the minimum of a function \( f \), designated cost function or objective function (OF), which measures the difference between the experimental data and the predictions of the model that one is trying to fit. In mathematical notation, given a function \( f \), from some set \( D \) to the real numbers, one is searching for a value, or vector of values, \( \theta_{\text{opt}} \), contained in \( D \), such that

\[
 f : D \mapsto \mathbb{R} \\
 f (\theta_{\text{opt}}) \leq f (\theta), \quad \forall \theta \in D
\]  

(1.6)

where vector \( \theta_{\text{opt}} \) contains the optimal parameters for the model. According to this formulation, vector \( \theta_{\text{opt}} \) will have to be a global minimum of function \( f \), but one could also be interested in finding a local minimum, in which case equation (1.6) would become

\[
 f (\theta_{\text{opt}}) \leq f (\theta), \quad \forall \theta \text{ near } \theta_{\text{opt}} : \theta \in D
\]  

(1.7)

When trying to fit a mathematical model to experimental data, function \( f \) will usually be the sum of squared residuals, that is, the sum of the squares of the differences between the model’s predictions and the experimental observations. This method of fitting is thus designated method of least squares: given a set of \( m \) data points, \( x_1, y_1, x_2, y_2, \ldots, x_m, y_m \), and a mathematical model described by a function \( y = g(x, \theta) \), one is seeking the set of parameters \( \theta \) which minimize

\[
 f(\theta) = \sum_{i=1}^{m} (y_i - g(x_i, \theta))^2 \quad \text{with} \quad \theta = \theta_1, \theta_2, \ldots, \theta_n, \quad m \geq n
\]  

(1.8)

Minimizing a function corresponds to finding the root of its derivative, and when function \( g \) is linear in the parameters (linear least-squares problem), setting the derivatives of \( f \) (with respect to the \( n \) parameters) to zero yields an expression for the values of the optimal parameters. On the contrary, when \( g \) is nonlinear in the parameters (nonlinear least-squares problem), there is no such formula. Instead, an initial estimate for the values of the parameters must be provided. This initial estimate is then iteratively refined, until convergence of the optimization method is attained. The method most
commonly used for solving nonlinear least-squares problems is the Levenberg-Marquardt algorithm [23], which is an iterative procedure that alternates between the Gauss-Newton algorithm and the gradient descent method. This is, however, a local optimization algorithm, since it will find the optimum closest to the initial guess provided for the parameters, and may only converge if this initial guess is close to an optimum. In practice, therefore, several runs of this algorithm are performed, each of them departing from a different initial guess for the parameters.

Aside from its lack of elegance, this methodology may also become problematic, especially when i) the number of parameters to be estimated is large, ii) the range of allowed values for the parameters is large, iii) an accurate initial estimate is not available for a large percentage of the parameters, and iv) it is suspected that function $f$ has many local minima. This is precisely the case when it comes to models of biological systems. Even if the state variables of the system are no more than a handful, it turns out that the number of connections between them (representing e.g. biochemical reactions) is significantly larger, and thus requires that a large number of parameters be specified. Additionally, the nonlinear form of the model, coupled with its large number of parameters, increase the probability of encountering local minima. Other more powerful global optimization methods (e.g. branch-and-bound algorithms) unfortunately require that function $f$ satisfy certain conditions, namely that either $f$ or an outer approximate function of $f$ be convex [24]. Once again, such conditions are very hard to ensure when one uses, in the computation of $f$, either a highly nonlinear and complex model, or experimental data (or both).

Given the limited success of the aforementioned deterministic optimization algorithms in the context of biological kinetic problems, this work set forth to explore the use of algorithms which include a stochastic element in their operation. These kinds of algorithms are used for particularly hard optimization problems, when the more conservative deterministic approach fails. Among the many possible alternatives (e.g. simulated annealing, stochastic hill climbing, evolutionary algorithms, swarm algorithms), two algorithms were chosen:

1. the genetic algorithm (GA), belonging to the class of evolutionary algorithms, and
2. particle swarm optimization (PSO), belonging to the class of swarm algorithms.

In addition to being stochastic, these algorithms may also be classified as metaheuristics and as population-based (or ensemble) search methods [25;26].

Metaheuristics are heuristic methods which use other heuristic methods themselves, in order to solve computational problems for which there is no satisfactory algorithm available (or when such an algorithm is not efficient). Heuristics may be viewed as rules of thumb or educated guesses about problems, or rather yet, as problem solving strategies employing this kind of empirical knowledge. They do not generally use the sophisticated, yet computationally heavy to obtain, mathematical knowledge of the problem, to guide the optimization, and are not ensured to arrive at the optimal solution. Instead, they aim at quickly arriving at a solution that is very close to the optimum. Metaheuristic algorithms normally use several operating parameters for governing the application of the heuristics, and they need to be properly adjusted in order to optimize the algorithms' performance.

Population-based metaheuristics such as the ones explored in this work have the particularity of operating on a pool of candidate solutions to the problem, rather than on a single solution. They are
iterative procedures which rely on a set of heuristic methods to generate and combine these candidate solutions at each step of the algorithm, until convergence is reached and/or a good enough solution is found among the candidates. Although these population-based metaheuristics do not guarantee that the global optimum is found, they use their pool of candidate solutions to perform a global search for that optimum, thoroughly exploring the search space of the problem, and thus increasing the probability of finding a global optimum. Moreover, the combination of candidate solutions via the heuristics used, often results in the emergence of synergistic effects, which are crucial for an efficient operation of the algorithm. A short description of the GA and PSO algorithms is presented in the next two paragraphs, leaving the detailed explanation of the particular implementations of these algorithms used in this work to Chapter 5.

The genetic algorithm was first introduced by John Holland in 1975 in his book “Adaptation in Natural and Artificial Systems” and belongs to the wider class of evolutionary algorithms, which also includes the related techniques of evolutionary programming, evolution strategies and genetic programming. The GA takes inspiration from genetic phenomena as they are related to evolution. Its pool of candidate solutions to the problem is called the population, and the individual solutions are designated individuals. An individual usually possesses a single chromosome (though in some variants of the GA it may be composed of two chromosomes), and chromosomes are made up of individual genes. In classic versions of the GA, the phenotype of an individual (the actual value of the solution) has to be decoded from its genotype (the chromosome representing the solution, e.g. in binary coding), but this decoding step is not necessary when the individuals are represented using real values. The genetic algorithm starts with a random initial population, and simulates the evolution of this population over a number of generations, which correspond to the iterations of the algorithm. At each generation, a number of individuals are selected for reproduction, according to their fitness. The fitness of an individual measures how good the corresponding candidate solution is, and is therefore computed via the objective function defined for the particular problem being solved. Reproduction is carried out by means of crossover and mutation, which constitute the main heuristics used by the GA. In crossover, a portion of the chromosomes of a pair of individuals is swapped between them, while in mutation small random changes are introduced in a chromosome. The population at the next generation is composed of the offspring of the individuals selected for reproduction. Although the emphasis has been classically placed on crossover, rather than on mutation, it is true that some problems are more easily solved using only mutation, while others depend crucially on the use of the crossover operator [27], the success of the approach also depending on the size of the population [28]. Many variants of the simple GA, including the introduction of additional heuristics, have been proposed since 1975 [29-33], and much work regarding the controversial theoretical basis of its operation as an optimization algorithm (e.g. the schema theorem and the building block hypothesis) continues to appear in the literature [28;34-40].

Particle swarm optimization is an algorithm inspired by the natural phenomena of bird flocking, firstly proposed by Kennedy and Eberhart in 1995 [41]. It is part of a class of algorithms based on naturally occurring swarming behavior, examples of which include ant colony optimization, the bees algorithm, or the more recent firefly algorithm. It is a stochastic iterative procedure which works on a pool of candidate solutions to a problem, called the swarm (or the population). The individual candidate solutions are designated particles. Each particle is a vector containing the coordinates for the position of the particle in the search space, and thus, contrary to what happens with the binary GA, there is no
The distinction between the search space of the algorithm and the search space of the problem. The PSO departs from a random initial swarm of particles, evolving it over a number of epochs (or iterations), and using a set of rules (the heuristics) to update the positions of the particles at each iteration. Thus, also in opposition to the GA, there is no selection process, as the update rules are applied to all particles. At each iteration, the objective function values of the particles are computed, and the global best particle in the swarm (the one with the lowest objective function value) is found. A record of the best position found by every particle (local bests) is also kept. The update rule of the velocities takes into account both of these quantities, also called the social and cognition components of the algorithm. Each particle is then “pulled”, with a given force, towards the best position it has found so far, and, with a different force, towards the best global position found by the swarm. Indeed, as Poli notes [42]: “we can interpret the components (...) as attractive forces produced by springs of random stiffness, and we can approximately interpret the motion of a particle as the integration of Newton’s second law.” Much less variants exist for the PSO algorithm [43-46] than for the GA, and theoretical analyses of the convergence properties of this algorithm are also available [43;47-51].

In recent years, both the GA and the PSO (as well as hybrid algorithms, combining the two) have been applied to parameter estimation problems, in the context of microbial growth kinetic models [52;53], metabolic models [54], models of chemical reactions [55;56], and models of biological reactors [57]. An exhaustive and interesting study is that by Drager et al [54], in which eight different optimization strategies (including the GA and the PSO, as well as other metaheuristics) are compared, in the context of the estimation of the parameters of a complex metabolic model for C. glutamicum. The results of this study show that the PSO is one of the best optimization strategies.

1.3 Aims

The primary aim of this work is to develop a mathematical model capable of describing the physiology of the yeast D. hansenii under certain conditions. This includes the estimation of the parameters of the model, and will allow the formulation of a hypothesis concerning the reason for the distinct physiological response of D. hansenii in two sets of experimental conditions.

A secondary aim of this work is to explore the performance of two population-based stochastic optimization algorithms, the genetic algorithm and particle swarm optimization, in the estimation of the parameters of the model.

1.4 Outline of the Thesis

Besides the present Introduction, this thesis is composed of six more chapters: chapter 2, designated Methodology, provides an outline of the strategy followed to solve the problem; chapter 3, Mathematical Modeling of D. hansenii Physiology, elaborates on the construction of the physiological model for this yeast; chapter 4, Parameter Estimation as an Optimization Problem, defines the optimization problem that must be solved in order to estimate the parameters of the physiological model; chapter 5, The Optimization Algorithms, describes the algorithms used to solve the aforementioned optimization problem; chapter 6, Results and Discussion, presents and analyzes the main results of this work; and finally, chapter 7, Conclusions, rounds up the main achievements of the work developed for this thesis.
2 Methodology

The aim of this chapter is to provide a succinct overview of the methodology followed in this work. In Figure 2-1 a diagram for this methodology is shown.

In order to achieve the primary aim of this work (to obtain a model for *D. hansenii* physiology), two main problems must be addressed:

1. The mathematical formulation of a physiological model for *D. hansenii*, and
2. The estimation of the parameters of the physiological model.

The first problem is tackled in Chapter 3 of this thesis. It leads to the development of nine distinct physiological models for *D. hansenii*, each formulated as a system of ordinary differential equations, depending on a set of parameters.

The second problem is examined in Chapter 4 of this thesis. The parameter estimation procedure is defined as an optimization problem, in which one tries to minimize the discrepancy between the predictions of the model and the experimental data. The objective function (OF) of this minimization problem is defined, and the variability of the experimental data is analyzed, in order to obtain a criterion against which to compare the result of the optimization. The available experimental data is divided in two groups: group I refers to 3 experiments carried out under standard experimental conditions, and group II to 4 experiments carried out in the presence of inhibitory substances.

The complex nature of the optimization problem in question leads to the proposal of two different population-based stochastic optimization algorithms for its resolution. They are the genetic algorithm (GA) and particle swarm optimization (PSO), and a detailed description of their operation is provided in Chapter 5 of this thesis.

The proposal of two algorithms reflects the secondary aim of this work, which is to compare their performance on this particular problem, thus finding the algorithm which is more appropriate for solving the problem. Both the GA and PSO have adjustable operating parameters, which greatly influence their performance, and thus, a meta-optimization problem arises as a preliminary step in the methodology. This problem consists of finding the best operating parameters for each of the algorithms, and it is usually termed tuning (of the algorithm) [38;58;59], precisely because the algorithm operating parameters are being adjusted to the problem being solved. For practical reasons, and because the models and data groups are similar among themselves, only one model and data group are used to tune the algorithms. Due to the stochastic nature of these algorithms, 20 independent runs are obtained for each of the 12 sets of operating parameters tested.

After obtaining the best settings for the GA and the PSO, one chooses the best overall algorithm, and uses it to estimate the parameters of the nine models, against both of the data groups (also obtaining 20 independent runs for each model).
CHAPTER 3 Mathematical Modeling of *D. hansenii* Physiology
- 9 alternative models with different structures are proposed.
- Each model is encoded as a system of ordinary differential equations, with a certain number of parameters.

CHAPTER 4 Parameter Estimation as an Optimization Problem
- The objective function of the optimization problem is defined.
- The experimental data (2 groups of data) are analyzed to obtain a criterion to evaluate the optimization's result.

CHAPTER 5 The Optimization Algorithms
- The Genetic Algorithm (GA) is described.
- Particle Swarm Optimization (PSO) is described.

Meta-Optimization: Tuning of the Algorithms' Operating Parameters
- Only one data group (data group I) and one model (model 3A) are used in this procedure.
- For both the GA and the PSO, 12 tuning cases are defined, corresponding to different settings of the algorithms' operating parameters.
- Because the algorithms are stochastic, each tuning case is run 20 times, and measures of performance are recorded.

<table>
<thead>
<tr>
<th>GA</th>
<th>PSO</th>
</tr>
</thead>
<tbody>
<tr>
<td>The influence of the crossover fraction and of the mutation and crossover functions is explored.</td>
<td>The influence of the acceleration constants and inertia weight is explored.</td>
</tr>
</tbody>
</table>

Best Settings for the GA  Best Settings for the PSO

Best Algorithm with Best Settings

Parameter Estimation for all the proposed Models
- The estimation of parameters for all the models and both data groups is carried out using the best algorithm with the best operating parameters.
- As before, the algorithm is run 20 times for each model.
- The evaluation of results entails analyses of the performance of the algorithm for each model, of the goodness-of-fit of the models' predictions, and of the parameters obtained.

Best Model(s)

Figure 2-1. Block diagram showing the strategy followed in this work.
The evaluation and discussion of the results of parameter estimation is done in Chapter 6, and includes the analysis of:

i) the performance of the algorithm for each model, considering the OF values in the 20 runs obtained, and

ii) the goodness-of-fit of the models' predictions.

As for the choice of the best model (or models), it will be based on two criteria:

1. the sum of the best OF values obtained by each model for the two data groups, and

2. the number of parameters of the model.

The first criterion is, of course, a measure of the agreement of the predictions of the model with the experimental data. Both data groups are considered in this measure, because if a model which has found a good optimum for the standard data group, is still able to find a good optimum for the data group containing the inhibitors, then that model is flexible enough to appropriately capture the dynamics of this effectively more complex new system, without adding any extra parameters.

The second criterion is related to the Occam’s razor principle: if a model with a certain number of parameters is as effective (or more effective) at describing the experimental data as another, possessing a larger number of parameters, then the former is preferable to the latter.

After the best model is chosen, a comparison of the parameters obtained for this model on the two data groups is made. Although the concentrations of the inhibitory substances present in the experiments of data group II are not taken into account in the model, their effects on the dynamics of the species modeled should come across as a change in the values of the parameters of the model. Since the parameters have a biological meaning, the observed variations in their values may provide clues as to what is the response of the microorganism to the inhibitors.
3 Mathematical Modeling of *D. hansenii* Physiology

This chapter focuses on obtaining a set of alternative physiological models capable of describing the experimental time courses observed for the concentrations of eight species: biomass (X), glucose (Glc), glycerol (GlyOH), ethanol (EtOH), xylose (Xyl), xylitol (XOH), arabinose (Ara) and arabitol (ArOH) in fermentation profiles of *Debaryomyces hansenii*.

Section 3.1 exposes and discusses the experimental and metabolic knowledge that needs to be taken into account in order to obtain an adequate physiological model for *D. hansenii*, and which will also help in defining the conditions under which the application of such model is valid.

In section 3.2 several alternative networks for the physiological model of *D. hansenii* are proposed and examined. The main reason for testing several models is that, in the spirit of the Occam’s razor principle, one is interested in finding the simplest model that gives an adequate description of the data. In this context, the simplest model is the one that specifies the smallest number of parameters and/or the smallest number of relations between the species. By using a set of distinct models, one is testing which physiological phenomena are significant enough to be included in the model, and which are less important and do not need to be taken into account.

Finally, section 3.3 details how the model’s network is translated into a system of ordinary differential equations. This system shall describe the temporal rate of change of the concentration of each species, as depending on their own concentration, on the concentration of other species, and on a set of parameters, which one wishes to estimate.

3.1 Preliminary Considerations

This section is divided into three parts. Section 3.1.1 discusses the settings under which the experimental data used in this work were obtained, defining what the model must capture and under which conditions its application is valid. Section 3.1.2.1 briefly examines the metabolic behavior of *D. hansenii* in regard to its utilization of the sugars glucose, xylose and arabinose. The considerations in these two subsections will form the basis for the construction of a physiological network for *D. hansenii*, undertaken in section 3.2.

3.1.1 Experimental Conditions and Range of Validity of the Model

In the experiments used in this work [60], *D. hansenii* was grown in completely chemically defined media, with a sugar composition designed to simulate the composition of concentrated, pentoses-rich, hemicellulosic acid hydrolysates, a potential industrial feedstock for this microorganism. The cells were cultured in “batch mode” [61], meaning that they were provided with a specific amount of each of the sugars (and other nutrients such as nitrogen, phosphorus, magnesium, sulfur, trace elements and vitamins) at the beginning of the experiment, and only at the beginning of the experiment. The three sugars supplied were D-glucose (a hexose), D-xylose and L-arabinose (pentoses). The experiments were carried out aerobically in cotton-capped Erlenmeyer flasks for a total of 168 h, with samples being taken at regular intervals (10 samples for each experiment). These samples were used to obtain

---

2 The cells were previously pre-cultured to the point when exponential cell growth was observed, so as to ensure that the lag phase of growth would be minimized in the main experiment.
the values of the concentrations (in grams per liter, g/L) of the three sugars – glucose, xylose and arabinose –, of the metabolic products – the alcohols ethanol, glycerol, xylitol and arabinose -, and also to quantify biomass dry weight (also in g/L). The analytical techniques used to this end were high-performance liquid chromatography (HPLC), for sugars and alcohols, and gravimetric analysis after centrifugation, for biomass dry weight.

With the exception of these samples, there was no addition to and no withdrawal from the culture, of biomass, nutrients, or culture broth (except for substances on the gaseous phase, which were able to diffuse in and out of the reactor). The temperature was kept constant and the pH of the medium was partially buffered, with available pH measurements showing a pH decrease throughout the experiment (with values ranging from 5.67 at the beginning of the experiment, to 2.47 at the end). The dissolved oxygen levels were not measured, and thus, despite its appreciable effect on the physiological behavior of the organism, oxygen cannot be included in the model.

One important characteristic of the experiment is that it ended when the cells exhausted all the available nutrients, i.e. when the concentration of arabinose was zero. Therefore, the model only needs to take into account biomass growth and not biomass death/decay, since in the presence of nutrients no net biomass decay was observed in the data. Besides biomass, all the other species measured are extracellular - they are either the primary substrates provided to the cells (the sugars glucose, xylose and arabinose) or products that the cells were able to excrete to the extracellular medium and later reassimilate (the alcohols glycerol, ethanol, xylitol and arabinol).

The main restrictions regarding the validity of application of the models developed for D. hansenii physiology in this work, concern the initial concentrations of the eight species considered. In a strict sense, the conditions under which the models are valid correspond to the experimental range of initial concentrations for these species, that is, to the initial concentrations of these species in the different experiments considered in this work. They are shown in Table 3-1.

<table>
<thead>
<tr>
<th>Species</th>
<th>Initial Concentration (g/L)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Lowest Value</td>
</tr>
<tr>
<td>Glc</td>
<td>19.59</td>
</tr>
<tr>
<td>Xyl</td>
<td>55.7</td>
</tr>
<tr>
<td>Ara</td>
<td>18.87</td>
</tr>
<tr>
<td>X</td>
<td>1.5</td>
</tr>
<tr>
<td>GlyOH, EtOH, XOH, ArOH</td>
<td>0</td>
</tr>
</tbody>
</table>

Although no claims can be made regarding model validity outside of this range of initial concentrations, one can speculate that the model may still be valid for a wider range of initial concentrations of biomass and the three sugars, provided that the ratios of sugar concentration and the ratio of each sugar to biomass concentration are kept similar to those in the present experiments.

### 3.1.2 The Metabolism of Glucose, Xylose and Arabinose in D. hansenii

This section examines the utilization by D. hansenii of glucose, in subsection 3.1.2.1, and of the pentoses xylose and arabinose, in subsection 3.1.2.2, defining, in subsection 3.1.2.3, the inhibitory effect that some of these sugars exert on the utilization of the others.
Because the goal of this section is to set forth the necessary background for the definition of the structure of the physiological network of *D. hansenii* (enlightening the phenomena that such a model must be able to predict), the approaches followed on the different models will be mentioned, after a general exposure of the background has been accomplished.

### 3.1.2.1 Glucose Metabolism

Glucose is metabolized by *D. hansenii* through the glycolysis pathway. At the fourth step of this pathway, glyceraldehyde-3P is formed, and may be transformed into glycerol and excreted out of the cell. In the last step of glycolysis, pyruvate is formed. Pyruvate may be further transformed into acetaldehyde and later ethanol, which the cell also secretes; alternatively, if oxygen is available, it may enter the tricarboxylic acid cycle (which provides redox potential and precursors for many cell components), resulting in the formation of biomass.

The excreted products, glycerol and ethanol, are reassimilated by the cell. Glycerol is transformed back into glyceraldehyde-3P and continues to be degraded in the course of glycolysis, while ethanol is transformed back into acetaldehyde and later pyruvate, leading once again to biomass growth. A simplified graphical depiction of glucose utilization by *D. hansenii* may be found in Figure 3-1 below.

![Figure 3-1. Simplified pathway of glucose utilization by *D. hansenii*.](image)

Some of the models for the metabolism of ethanol produced in this work will explore the possibility of not taking into account one particular prediction of this network: the production of ethanol from glycerol. The reason for this is that the experimental concentrations of glycerol are extremely low, and thus the amount of ethanol produced from the small amount of glycerol that reenters the cell, could be considered small enough to be negligible, in the present context.

Other models, on the other hand, will explore the possibility that ethanol is not directly produced from glucose, and that it is only formed through glycerol. This is also consistent with the low glycerol concentrations observed, provided that the rate of conversion of glucose to glycerol and of glycerol to ethanol are high, and, simultaneously, the rate of conversion of ethanol to biomass is lower (leading to higher concentrations of ethanol than glycerol).

### 3.1.2.2 Xylose and Arabinose Metabolism

Once inside the cell, both D-xylose and L-arabinose are reduced by *D. hansenii* to the polyols xylitol and arabitol, respectively. Xylitol and arabitol may be secreted by the cell, but are later reassimilated and transformed into xylulose-5P. This metabolite then suffers further transformations and leads to the growth of biomass, mainly by entering the pentose phosphate pathway and the glycolytic pathway (and thus also being able to lead to ethanol production).

In *D. hansenii* the enzyme xylose-reductase (E.C.1.1.1.21), which reduces the D-xylose to xylitol, is indistinguishable from the aldose-reductase which reduces L-arabinose to arabitol, and its action on
D-xylose may result in the formation of arabitol instead of xylitol. This was subtly visible in the experimental data, with a slight arabitol accumulation occurring before any L-arabinose consumption had been detected. It shall be seen in section 3.3 that this knowledge can be incorporated into the model, leading to the rate equation for xylose consumption taking a form which is significantly different from the rate equation for glucose consumption, despite the fact that both substrates lead to the appearance of two products. Figure 3-2 below shows a simplified diagram for xylose and arabinose utilization by *D. hansenii*.

![Figure 3-2. Simplified pathways of xylose and arabinose utilization by *D. hansenii.*](image)

### 3.1.2.3 Inhibitory Effects: Catabolite Repression

Though there is evidence that xylose is consumed simultaneously with glucose, it is also apparent that the consumption of the former is slightly delayed in the presence of the latter [60;62]. Arabinose consumption is inhibited by the presence of both xylose [62] and glucose [60]. However, when these three sugars are present, it is only necessary to model two out of these three inhibitions: the one exerted by glucose on xylose consumption, and the one exerted by xylose on arabinose consumption. This is because when the three species are present, the fact that glucose inhibits arabinose consumption is implicit in the facts that glucose inhibits xylose consumption and xylose inhibits arabinose consumption.

One other important inhibitory effect is that of xylose on xylitol consumption, which is partially responsible for the high accumulation of xylitol achieved by *D. hansenii*. A further biochemical explanation is related to the occurrence of a redox imbalance inside the cell under oxygen limited conditions, leading to an inhibition of the activity of the enzyme which degrades xylitol (xylitol dehydrogenase), and hence to its accumulation [63].

### 3.2 Structure of a Physiological Network for *D. hansenii*

In developing any kind of network, one must in general define two types of structural components of the network: its nodes and the connections between them. In the particular case of the physiological model developed for *D. hansenii*, the nodes of the network which specifies the model represent the species (either simple chemicals or biomass) whose concentrations time courses one wishes to predict, and a connection between given nodes represents a reaction in which the species on the upstream node is consumed and the species on the downstream node is produced.
It is important to notice that a loose definition of the term “reaction” is used here, as it does not refer to any particular enzyme-catalyzed reaction in a metabolic pathway of the microorganism, but simply indicates that a given species is produced from another. Indeed, the metabolic pathways of *D. hansenii* will not be thoroughly analyzed here, although some rather detailed descriptions of particular enzymatic reactions will be taken into account in the development of the model.

The models herein proposed are comprehensively divided into three groups, each comprising two to four distinct models. The main differences among the groups concern the manner in which glucose metabolism is modeled, and the degradation of the polyols produced from the pentoses, namely: in some models polyols are fully transformed into ethanol and it is ethanol which is transformed into biomass, while in others, ethanol production from polyols is not predicted, and the polyols may only be assimilated into biomass.

The networks belonging to group 1 can be seen on Figure 3-3 below.

![Figure 3-3. Models 1A, 1B, 1C and 1D.](image)

This is the largest and most heterogeneous group of all, containing four different models for the network: 1A, 1B, 1C and 1D. Network 1A models glucose metabolism in a fully sequential fashion: glucose is converted to glycerol, which is converted to ethanol, which is converted to biomass. It introduces an inhibition, exerted by glucose on ethanol consumption, which was not mentioned in the previous section. The introduction of this inhibition is intended at predicting a higher accumulation of ethanol (less ethanol should be consumed if glucose is available), since preliminary results showed low ethanol accumulation without this inhibition. As for the metabolisms of xylose and arabinose, they are as explained in section 3.1.2.2, with the simplification that the production of biomass from the polyols, is not direct, but entails the production of ethanol as an intermediate species. The reason why the conversions of xylose into xylitol and arabitol are displayed as two dashed lines will be explained in section 3.3.
Network 1B is analogous to 1A, except for two things: 1) there is no inhibition exerted by glucose on the consumption of ethanol, and 2) glucose metabolism is no longer fully sequential, since an additional reaction, from glucose directly to biomass, is included. Network 1C is also analogous to 1A, except for the facts that 1) there is no inhibition exerted by glucose on the consumption of ethanol, and 2) an additional reaction, from glucose directly to ethanol, is predicted (instead of the direct reaction from glucose to biomass, defined in network 1B).

In network 1D, glucose only directly produces glycerol, which is then converted to either biomass or ethanol. Xylitol and arabitol may also produce either ethanol or biomass. The inhibitory effect of xylose on xylitol consumption was also deleted, since preliminary results indicated that this effect was very weak.

The networks in group 2 are shown in Figure 3-4 below.

The main feature of this group of models is that it divides ethanol into two conceptually distinct pools: \( \text{EtOH}_{\text{H}} \), originating the hexose (Glc), and \( \text{EtOH}_{\text{P}} \), originating from the pentoses (Xyl and Ara). The sum of the concentrations of these two pools of ethanol corresponds to the concentration of the actual species, ethanol (\( \text{EtOH} \)). This artificial division was made because preliminary results showed poor modeling of ethanol concentration. It was speculated that these results could be due to the conversion of ethanol to biomass being required to occur at a very high rate, in order to account for the early steep biomass growth. Since the conversion of the pentose-originating ethanol to biomass would not be required to be as fast (as biomass growth is less steep in the remainder of the experiment), this division would be able to predict more ethanol accumulation, thus resulting in a better modeling of the actual ethanol time course. With the exception of the introduction of these two artificial species, networks 2A, 2B and 2C, are analogous to networks 1A, 1B and 1C, respectively. The only difference is in the inhibition exerted by glucose on the conversion of ethanol to biomass, which is present in model 1A but is not in model 2A.
The two final models, 3A and 3B, belonging to group 3, are displayed in Figure 3-5 below.

![Figure 3-5. Models 3A and 3B.](image)

Network 3A models glucose metabolism in a parallel fashion. The only simplification made, with respect to the diagram shown in Figure 3-1, is that glycerol does not produce ethanol, which was justified in section 3.1.2.1. As for xylose and arabinose, the model is as explained in section 3.1.2.2, with the further assumption that xylitol and arabitol are only used to produce biomass and do not lead to ethanol formation. Network 3B simplifies glucose metabolism, making the assumption that biomass is not produced directly from glucose – glucose is transformed into glycerol or ethanol, which are secreted, reassimilated, and only then converted to biomass. In all other respects, this network is equivalent to network 3A.

A complete list of the reactions and inhibitions considered in each model may be found in Appendix B.

### 3.3 The Kinetic Model as a System of Ordinary Differential Equations

The Monod equation, relating the rate of biomass growth with the concentration of a single limiting substrate, is, in its origin, an empirical law. Its derivation was not made based on any mechanistic considerations, regarding the biochemical phenomena underlying microbial growth. Nevertheless, the parameters that the equation specifies do lend themselves to a biologically meaningful interpretation and, as such, several attempts [11] have been made to obtain a theoretical derivation to this law. Such derivations include, for example, arguments based on the thermodynamics of microbial growth [64], or on the consideration that there is a significant temporal gap between the uptake of substrate by the cells and the resulting increase of total microbial cell mass, with the Monod equation arising as an approximation of the original delayed differential equation used to describe this process [10].

A theoretical derivation of the Monod model could also be useful in obtaining a sound extension of this model to be applied when multiple substrates are present, which, besides being a more realistic setting, both in nature and in industrial applications, turns out to be much more elusive than that of a single limiting substrate. In fact, in the present work, there are three primary substrates – glucose, xylose and arabinose –, whose consumption by the microorganism leads both to growth and to the formation and secretion of a series of products – ethanol and the polyols glycerol, xylitol and arabitol. Furthermore, these products are able to reenter the cell and are thereon reincorporated in the microorganism’s metabolism, serving as secondary substrates and thus leading once again to biomass growth.

Although there exist well-established methods for modeling microbial growth on more than one substrate [12], these methods are only applicable when all the substrates present are limiting
substrates [65], that is, when the cell requires all substrates to be present in order to grow. Because this is not the case with D. hansenii and the array of substrates with which it is fed in the present experiments, a different approach had to be followed in this work.

This approach was based on the simplifying assumption that the rate of change of a given species may be written as a sum of terms, accounting for each and every reaction in which it participates (as substrate or product). Each of these terms, or individual reaction rates, was assumed to take the form of the Monod equation. The kinetic model developed for the physiology of D. hansenii is thus formulated as a system of ordinary differential equations (ODE system), with as many equations as there are species. This system describes the rate of change of the concentrations of each of these species throughout the duration of the experiment (168 hours). In the context of dynamical systems' formalism, these species are called the state variables, since the values that they take define, at each time instant, the state in which the system is found.

3.3.1 Derivation of the ODE System from the Model’s Network

The general methodology followed for the derivation of this system of ODEs will be presented, with model 3B (depicted in more detail in Figure 3-6) serving as an illustrative example. The differential equations for the remaining models are similar to the ones derived here and may be found in Appendix B.

![Figure 3-6. Diagram for the network of model 3B, with reactions and inhibitions labeled. The black arrows (solid as well as dashed) represent reactions, and the red lines ending in circles represent inhibitions.](image)

The derivation of the system of differential equations corresponding to the network diagram of a given model proceeds as follows:

i) For each of the reactions Rj in the diagram, with j ranging from 1 to 8 (see Figure 3-6), a specific rate, \( \mu_j \), with the hyperbolic form of the Monod equation, is defined:

\[
\mu_j = \mu_{\text{max}, j} \frac{S}{S + K_{S,j}}
\]  

(3.1)
where \( \mu_{\text{max},j} \) is the maximum specific rate of the reaction, \( K_{S,j} \) is the half-saturation constant, and \( S = S(t) \) is the concentration of the species which is consumed in the reaction (the substrate), at time \( t \).

ii) If the reaction has only one product (solid black line), then a yield parameter, \( Y_j \), corresponding to the ratio between the amount of product formed to the amount of substrate consumed:

\[
Y_j = \left| \frac{dP}{dS} \right|_{\text{by}}
\]

(3.2)

iii) If the reaction has several \((n>1)\) products (with \( n \) dashed black lines branching from its main solid black line, as e.g. reaction R5 in Figure 3-6), then \( n \) yield parameters, \( Y_{j,k} \), are defined:

\[
Y_{j,k} = \left| \frac{dP_k}{dS} \right|_{\text{by}} \quad k = 1, \ldots, n
\]

(3.3)

iv) If reaction \( R_j \) is inhibited by a particular species, \( I \), then its specific rate becomes:

\[
\mu_j = \mu_{\text{max},j} \frac{S}{S + K_{S,j}} \frac{K_j}{K_j + I}
\]

(3.4)

where \( I \) is the concentration of the inhibitor and parameter \( K_i \) quantifies the strength of the inhibition. A very high \( K_i/I \) ratio makes the inhibition factor tend to 1, thus indicating a weak inhibitory effect (the value of the specific rate \( \mu_i \) is only slightly affected), while a very low \( K_i/I \) ratio results in the inhibition factor tending to \( K_i/I \), indicating a strong inhibition (the value of the specific rate \( \mu_i \) is significantly lowered).

v) For each species \( S_m \) in the network (\( m \) ranges from 1 to 8 in model 3B in Figure 3-6), its rate of change is defined as a sum of negative terms, corresponding to the reactions in which it is consumed, and positive terms, corresponding to the reactions in which it is produced:

\[
\frac{dS_m}{dt} = \left( -\sum_i \mu_i + \sum_j \sum_k Y_{j,k} \mu_j \right) \cdot X
\]

(3.5)

where \( C \) is the set of all reactions in which \( S_m \) is consumed, \( F \) is the set of all reactions in which \( S_m \) is formed, and \( k \) is, as defined above, each of the \( n \) possible subdivisions of reaction \( j \) (if \( k=0 \) then \( Y_{j,0} \) is simply written \( Y_j \)).

This is thus an additive model, in the sense that the rate of change of the concentration of each species is described as a sum of terms, each accounting for the effects that the individual rates of the reactions in which the species participates have on its total rate of change. The application of this methodology to model 3B yields the eight equations below. In the context of glucose metabolism, one

---

3 For practical reasons, the numbering shown in Appendix B for the \( Y \) of each branching reaction is not the same as the one herein defined; for this particular model, \( Y_{5,1} \equiv Y_5 \) and \( Y_{5,2} \equiv Y_5 \).
obtains the following three equations, describing the temporal rate of change of glucose, glycerol and ethanol:

\[
\frac{d\text{Glc}}{dt} = \left[ -\mu_{\text{max},1} \cdot \frac{\text{Glc}}{\text{Glc} + K_{\text{Glc},1}} - \mu_{\text{max},2} \cdot \frac{\text{Glc}}{\text{Glc} + K_{\text{Glc},2}} \right] \cdot X
\]  

\[
\frac{d\text{GlyOH}}{dt} = \left[ Y_1 \cdot \mu_{\text{max},1} \cdot \frac{\text{Glc}}{\text{Glc} + K_{\text{Glc},1}} - \mu_{\text{max},3} \cdot \frac{\text{GlyOH}}{\text{GlyOH} + K_{\text{GlyOH},3}} \right] \cdot X
\]  

\[
\frac{d\text{EtOH}}{dt} = \left[ Y_2 \cdot \mu_{\text{max},2} \cdot \frac{\text{Glc}}{\text{Glc} + K_{\text{Glc},2}} - \mu_{\text{max},4} \cdot \frac{\text{EtOH}}{\text{EtOH} + K_{\text{EtOH},4}} \right] \cdot X
\]

The reactions in which these species participate are not branched nor inhibited, and thus they are simpler than the remaining ones. For the metabolisms of the pentoses, xylose and arabinose, four equations are obtained:

\[
\frac{d\text{Xyl}}{dt} = -\mu_{\text{max},5} \cdot \frac{\text{Xyl}}{\text{Xyl} + K_{\text{Xyl},5}} \cdot \frac{K_{\text{Xyl},1} + \text{Glc}}{K_{\text{Xyl},1} + \text{Glc}} \cdot X
\]

\[
\frac{d\text{XOH}}{dt} = \left[ \frac{Y_5 \cdot \mu_{\text{max},6}}{\text{Xyl} + K_{\text{Xyl},5}} \cdot \frac{K_{\text{Xyl},1} + \text{Glc}}{K_{\text{Xyl},1} + \text{Glc}} - \mu_{\text{max},6} \cdot \frac{\text{XOH}}{\text{XOH} + K_{\text{XOH},6}} \cdot \frac{K_{\text{Xyl},1} + \text{Xyl}}{K_{\text{Xyl},1} + \text{Xyl}} \right] \cdot X
\]

\[
\frac{d\text{Ara}}{dt} = -\mu_{\text{max},7} \cdot \frac{\text{Ara}}{\text{Ara} + K_{\text{Ara},7}} \cdot \frac{K_{\text{Ara},13} + \text{Xyl}}{K_{\text{Ara},13} + \text{Xyl}} \cdot X
\]

\[
\frac{d\text{ArOH}}{dt} = \left[ -\mu_{\text{max},8} \cdot \frac{\text{ArOH} + K_{\text{ArOH},8}}{\text{ArOH} + K_{\text{ArOH},8}} + Y_5 \cdot \mu_{\text{max},6} \cdot \frac{\text{Xyl}}{\text{Xyl} + K_{\text{Xyl},5}} \cdot \frac{K_{\text{Xyl},1} + \text{Glc}}{K_{\text{Xyl},1} + \text{Glc}} + Y_7 \cdot \mu_{\text{max},7} \cdot \frac{\text{Ara}}{\text{Ara} + K_{\text{Ara},7}} \cdot \frac{K_{\text{Ara},13} + \text{Xyl}}{K_{\text{Ara},13} + \text{Xyl}} \right] \cdot X
\]

Note the branching of reaction 5, whose specific rate is

\[
\mu_5 = \mu_{\text{max},5} \cdot \frac{\text{Xyl}}{\text{Xyl} + K_{\text{Xyl},5}} \cdot \frac{K_{\text{Xyl},1} + \text{Glc}}{K_{\text{Xyl},1} + \text{Glc}}
\]

and which is multiplied in equations (3.10) and (3.12) by two different yield coefficients, \(Y_{5,1}\) and \(Y_{5,2}\), respectively. Reactions 5, 6 and 7 are inhibited, and their specific rates include, therefore, an inhibition factor, as defined in point iv) of the methodology above.

Finally, the temporal rate of change of biomass concentration is given by

\[
\frac{d\text{X}}{dt} = \left[ Y_6 \cdot \mu_{\text{max},3} \cdot \frac{\text{GlyOH}}{\text{GlyOH} + K_{\text{GlyOH},5}} + Y_4 \cdot \mu_{\text{max},4} \cdot \frac{\text{EtOH}}{\text{EtOH} + K_{\text{EtOH},4}} + Y_5 \cdot \mu_{\text{max},6} \cdot \frac{\text{XOH}}{\text{XOH} + K_{\text{XOH},6}} \right] \cdot X
\]

This system of eight ODEs, composed of equations (3.6) to (3.13), is highly nonlinear, both in the state variables and in its 28 parameters, and there is no general analytical solution for it, that is, there is no general formula that one may use to compute the concentrations of each species from their corresponding temporal rates of change. Nevertheless, the concentrations predicted by the model at
each time instant do need to be computed, since they are to be compared to the experimental ones in the course of the optimization. With no analytical solution available, a numerical integration routine must be used to that end.

3.3.2 Numerical Integration of the ODE System

Besides there being no general analytical solution for nonlinear systems of differential equations such as the ones described above, the particular systems used in this work have an additional hurdling characteristic: they are stiff. In the context of dynamical systems, stiffness means that the phenomena described by the differential equations of the system occur in substantially different time scales. To prove that a system is stiff is only possible for the simplest cases. For more complex systems, stiffness is usually detected when one tries to integrate the system using techniques that are not designed to deal with this characteristic. Hints of stiffness may also be given by the time courses of the state variables: in the present case, careful observation of the experimental data shows, for example, that the rate of change of glucose concentration is very high early in the experiment, while arabinose concentration practically does not vary until the last third of the experiment. In other words, the dynamics of glucose consumption occur in a time scale that is much shorter than that of the dynamics of arabinose consumption.

Furthermore, when the numerical integration of this system was performed, in the course of the optimization, with a general purpose numerical integration routine (ode45, available on MATLAB®), the time required for the computation of the concentrations gradually became larger, as the estimated parameters resulted in a better fit to the experimental data. The switch to a routine specialized in stiff systems (that is, a routine that uses variable step sizes to perform the integration), solved this problem. Such a routine is implemented by the MATLAB® function ode15s [66].
4 Parameter Estimation as an Optimization Problem

This chapter defines the optimization problem that needs to be solved when estimating the parameters of the kinetic models proposed for *D. hansenii*. Section 4.1 defines the objective function, whose value is to be minimized during the optimization. Section 4.2 discusses the analysis of the experimental data that needs to be performed in order to assess its variability, and hence provide a measure of comparison for the OF value found during the optimization.

4.1 Objective Function

The objective function is, by definition, the function which the optimization procedure seeks to minimize. It is therefore the driving force of the optimization and its value, at the end of the procedure, will provide a measure of the quality of the optimization’s result.

As mentioned in the Introduction, the present problem may be formulated as a nonlinear least-squares problem, for which one tries to minimize the sum of squared differences between the predictions of the model and the experimental observations. A precise statement for this problem is given in section 4.1.1. Section 4.1.2 examines the constraints that must be included in the formulation of the problem, while section 4.1.3 defines the safeguard that had to be applied in order to accommodate the unavoidable errors resulting from the numerical integration performed inside the objective function. Finally, section 4.1.4 provides an outline of the operation of the objective function.

4.1.1 Nonlinear Least-Squares

The experimental data can be divided in two groups. Group I consists of $N=3$ sets of data, each with 80 pairs of data points, referring to the concentrations of 8 substances at 10 different time instants.

$$t_{ij}, y_{ik} \quad i = 1, 2, N \quad j = 1, \ldots, 10 \quad k = 1, \ldots, 8$$

where $y$ is concentration, $t$ is time instant, $i$ is the data set index, $j$ the time index and $k$ the species’ concentration index. Group II consists of $N=4$ sets of data, each also comprising 80 pairs of data points. A more detailed description of the experimental data and the reason why this division was made shall be provided in the next section, but, for now, it suffices to know that the sets in each group of data refer to roughly the same experimental conditions. They are, therefore, replicates of the same experiment. This is the only data used in the present work, and no kind of interpolation was performed on the actual experimental data, with the purpose of artificially increasing the number of available data points. The initial conditions that must be provided in order to solve the system of ODEs modeling the data, are the concentrations of each of the 8 species at the initial time instant, $t_{i1}=0$. These are similar but not equal among the $N$ replicates in each data group. Given a total of $N\times80$ observed data points and $N\times80$ model predictions, the least-squares criterion attempts to minimize the $N\times80$-term sum

$$\sum_{i=1}^{N} \sum_{j=1}^{10} \sum_{k=1}^{8} (y_{ijk} - f(t_{ij}, y_{ijk}, \ldots, \theta))^2$$

(4.1)

where $\theta$ is the vector of estimated parameters, $y$ and $t$ were defined above, and $f$ gives the concentration of the $k$-th species, at time instant $t_i$, for the initial conditions on replicate $i$. Function $f$ is
the result of numerically integrating of the system of ODEs describing the model, with initial conditions defined by the initial concentrations in each replicate.

4.1.2 Constraints

Besides seeking to minimize the sum of squares defined by equation (4.1) above, one must also ensure that some conditions (derived from the physical interpretation of the problem), are satisfied during the estimation of the model's parameters. These conditions are the constraints of the problem, and one possible strategy for transforming a constrained problem into a non-constrained one is to replace the constraints with appropriate penalizations for their violation [67].

Three kinds of constraints will be discussed in the next three sections. Section 4.1.2.1 defines the bounds for the domain of the problem, that is, the range of values that the parameters are allowed to assume. To enforce this type of constraint, no penalizations were applied. Instead, the initial random pool of candidate solutions created by each algorithm, was constrained to lie within the bounds, and for every operation subsequently performed on this pool, a safeguard was introduced, which forced any value that violated a given bound to assume the value of the bound. Section 4.1.2.2 discusses the penalizations applied for violation of constraints related to the values of the concentrations, which must always be non-negative. Finally, section 4.1.2.3 examines the penalization applied when a given set of parameters violates a linear constraint on their values, e.g. a constraint which states that the sum of two parameters must not exceed a given value.

4.1.2.1 Definition of Bounds for the Parameters

An important physical constraint of this problem is concerned with the range of values that the parameters are allowed to take. This type of constraint is called a bound constraint, since appropriate lower and upper bounds must be defined for the parameters, so that the parameter estimation procedure will only need to consider a certain range of biologically meaningful (or reasonable) values for the parameters. Moreover, since this is actually a restriction on the search domain of the problem, its enforcement will allow the optimization algorithm to perform a finer exploration than it would if it had to search a larger domain.

All the parameters specified by the models developed here must be non-negative. Accordingly, their lower bounds were set to a very small value of $10^{-7}$. As for the upper bounds, their definition is not so straightforward, and the methods used to obtain them shall be discussed separately for each type of parameter, in the next paragraphs.

In regard to the definition of the bounds for the yield coefficients ($Y$), two different situations should be considered: i) the situation of a reaction of formation of a product from a substrate, and ii) the situation of biomass growth on a given substrate. In the first case, the yield of the reaction should not be higher than the overall yield observed in the metabolic pathway that the microorganism uses to accomplish it. One should then take the net stoichiometric coefficients (in mol/mol) of this reaction, and appropriately multiply them by the molar masses of the substances involved, in order to obtain the upper bound for the yield of such a reaction. For a reaction converting $n$ moles of substrate $S$ into $m$ moles of product $P$, one gets:

$$ n \cdot S \rightarrow m \cdot P, \quad \text{with } n, m \text{ in mol } \Rightarrow Y_{PS} = \frac{m \cdot M_P}{n \cdot M_S} \quad (4.2) $$
where \( M_S \) and \( M_P \) are the molar masses (in g/mol) of \( S \) and \( P \), respectively. The molar masses of the species considered in this work are listed in Table 4-1.

Table 4-1. Empirical formulas and molar masses of the chemical species considered in this work.

<table>
<thead>
<tr>
<th>Species</th>
<th>Empirical Formula</th>
<th>( M ) (g/mol)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Glc</td>
<td>( \text{C}<em>6\text{H}</em>{12}\text{O}_6 )</td>
<td>180.16</td>
</tr>
<tr>
<td>GlyOH</td>
<td>( \text{C}_3\text{H}_6\text{O}_3 )</td>
<td>92.09</td>
</tr>
<tr>
<td>EtOH</td>
<td>( \text{C}_2\text{H}_5\text{O} )</td>
<td>46.07</td>
</tr>
<tr>
<td>Xyl</td>
<td>( \text{C}<em>5\text{H}</em>{10}\text{O}_5 )</td>
<td>150.13</td>
</tr>
<tr>
<td>XOH</td>
<td>( \text{C}<em>5\text{H}</em>{10}\text{O}_5 )</td>
<td>152.15</td>
</tr>
<tr>
<td>Ara</td>
<td>( \text{C}<em>5\text{H}</em>{10}\text{O}_5 )</td>
<td>150.13</td>
</tr>
<tr>
<td>ArOH</td>
<td>( \text{C}<em>5\text{H}</em>{12}\text{O}_5 )</td>
<td>152.15</td>
</tr>
</tbody>
</table>

The product formation reactions in all the models developed in this work, along with their respective stoichiometric coefficients (the \( m \) coefficients, considering \( n=1 \)), may be found in Table 4-2 below.

Table 4-2. Stoichiometric coefficients for the reactions in the models.

<table>
<thead>
<tr>
<th>Product</th>
<th>GlyOH</th>
<th>EtOH</th>
<th>XOH</th>
<th>ArOH</th>
</tr>
</thead>
<tbody>
<tr>
<td>Glc</td>
<td>-</td>
<td>2</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>GlyOH</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Xyl</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>XOH</td>
<td>-</td>
<td>2</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Ara</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1</td>
</tr>
<tr>
<td>ArOH</td>
<td>-</td>
<td>2</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

As for the second possible situation, concerned with reactions in which the consumption of a given substrate leads to biomass growth, upper bounds for their yields cannot be so neatly derived. They were therefore set to values similar or slightly higher than those found in the literature [60;62]. Table 4-3 displays the values for the upper bounds of the yields of all the reactions considered in our work.

Table 4-3. Upper bounds for the yields of the reactions in the models (in g/g).

<table>
<thead>
<tr>
<th>Product/Biomass</th>
<th>X</th>
<th>GlyOH</th>
<th>EtOH</th>
<th>XOH</th>
<th>ArOH</th>
</tr>
</thead>
<tbody>
<tr>
<td>Glc</td>
<td>0.7</td>
<td>1.0223</td>
<td>0.5114</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>GlyOH</td>
<td>1</td>
<td>-</td>
<td>0.5003</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>EtOH</td>
<td>0.7</td>
<td>-</td>
<td>-</td>
<td>1.0135</td>
<td>1.0135</td>
</tr>
<tr>
<td>Xyl</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>XOH</td>
<td>1</td>
<td>-</td>
<td>0.61742</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Ara</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1.0135</td>
</tr>
<tr>
<td>ArOH</td>
<td>1</td>
<td>-</td>
<td>0.61742</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

The upper bounds for the maximum specific rates (\( \mu_{max} \)) of each reaction in the models were obtained from the experimental data as follows.

i) For each substrate and each experimental point (except for the first), the slope of the secant line passing through that point and the one immediately before it, was computed. (For species which act both as products and substrates, only the experimental points corresponding to their role as substrates were considered.) This slope was taken to be an
approximation of the temporal derivative of the concentration of that species at that time instant. For substrate $S$:

$$\frac{dS}{dt}_{\text{ini}} \approx \frac{S}{t_j} - \frac{S}{t_{j-1}} \frac{1}{t_j - t_{j-1}}$$

(4.3)

ii) This approximate derivative was then divided by the minimum concentration of biomass experimentally observed within the time interval to which the derivative referred, resulting in several upper bound candidate values for the $\mu_{\text{max},i}$ of the reaction in question:

$$\left| \mu_{\text{max},i} \right| \leq \left| \frac{S}{t_j} - \frac{S}{t_{j-1}} \frac{1}{X} \right|, \quad X = \min X_{j-1}, X_j$$

(4.4)

iii) Since none of these candidate bounds exceeded the value of 1.5 (the maximum was 0.9 for Glc consumption), the upper bounds for all the maximum specific rates were set to 1.5.

It is important to notice that this method relies on the assumption that, for a reaction in which substrate $S$ is consumed, the value of $K_S$ is very small when compared to the substrate concentration, $S$, since

$$K_S \ll S \leftrightarrow K_S \ll 1 \Rightarrow \frac{dS}{dt} = -\mu_{\text{max}} \frac{1}{1 + K_S / S} X = -\mu_{\text{max}} X$$

(4.5)

and thus $|\mu_{\text{max}}| = |dS/dt|/X$. While this may be a valid assumption for the substrates present in high concentrations and/or for which the microorganism has a high affinity (e.g. glucose, xylose, arabinose), it might not be for species with smaller concentrations (e.g. glycerol and ethanol). Nevertheless, the values of the candidate upper bounds obtained for these species using equation (4.4) were more than one order of magnitude smaller than the ultimately chosen upper bound of 1.5.

As for the half-saturation constants ($K_S$), it is very difficult to find reliable results in the literature concerning their values, for the various substrates considered in the present model. Although these values usually fall between the orders of magnitude of $10^{-1}$ to $10^{0}$ (with some values being closer to $10^{1}$), it is a matter of fact that they tend to depend on the value of the maximum specific growth rate, as well as on the initial concentration of substrate provided to the microorganism. The values of the half-saturation constants were thus set to the maximum experimentally observed concentration of the species being consumed in the corresponding reaction.

The inhibition constants ($K_I$) were the parameters allowed the largest range of values. This was mainly due to the fact that the range of reported $K_I$ values for various microorganisms and substrates is also very large [13], and sometimes they refer to different kinds of inhibitions. Also, it was reasoned that, by intentionally allowing a greater range of values for $K_I$, it would be interesting to observe if extremely high or extremely low values would be privileged by the algorithm. An extremely high value for $K_I$ would indicate that the inhibitory effect it refers to is extremely weak, and may thus be removed from the model without loss of accuracy. The upper bounds for the inhibition constants were thus set to 15 times the maximum experimentally observed concentration of the inhibitory species.

### 4.1.2.2 Penalization for Negative Concentrations

One very straightforward constraint of this problem is that the concentrations of each of the species must always be non-negative. Thus, a penalization must be applied when the model predicts negative
concentrations. However, a small negative value should not be as penalized as a large one, and thus, the penalization function for this constraint should also take into account the absolute value of the negative concentration. The total penalization, $\Phi$, is obtained by summing the individual penalization terms $\varphi(y_k)$ over all concentrations $y_k$. If $y_k$ is greater than or equal to zero, the penalization is zero; if it is negative, then the penalization corresponds to the square of its value multiplied by a factor of $10^6$:

$$\Phi = \sum_{k=1}^{N_k} \varphi(y_k) \quad \text{with} \quad \varphi(y_k) = \begin{cases} 10^6 \times y_k^2, & y_k < 0 \\ 0, & y_k \geq 0 \end{cases}$$

(4.6)

If, for example, $y$ falls between $-10^{-3}$ g/L and $-10^{-2}$ g/L, the corresponding penalizations will fall between 1 and $10^6$, respectively. But, if $y = -10$ g/L, the penalization will be as high as $10^6$. The value of $10^6$ for the multiplicative factor that defines the magnitude of the penalization was chosen so that the penalizations for acceptable negative values were close to the average of the sum of squared differences between equivalent experimental points, making this kind of “error” comparable to the least-squares criterion defined above.

4.1.2.3 Penalties for Violation of Linear Constraints

The penalization methodology was also applied to a different kind of constraint, stating that the sum of the yields of two branched reactions (see Chapter 3, section 3.3.1) must not be greater than the maximum of the two upper bounds for the yields of these reactions. For each pair of such reactions, with $\theta_i$ being the yield of reaction $i$, and $T$ being the maximum of the upper bounds for these yields (defined in Table 4-3), the penalization for the violation of this linear constraint, $\psi_{\theta_i}$, is given by

$$\psi_{\theta_i, \theta_j} = \begin{cases} 10^6, & \theta_i + \theta_j > T \\ 0, & \text{otherwise} \end{cases}$$

(4.7)

This constraint is applicable to the four branched reactions considered in this work, with $T = 1.0135$ for the conversion of xylose to either xylitol or arabitol, and $T = 1$ for the remaining 3 reactions (the conversion of glycerol to either ethanol or biomass, the conversion of xylitol to either ethanol or biomass, and the conversion of arabitol to either ethanol or biomass). For models containing more than one such constraint, the individual penalizations $\psi_{\theta_i}$ are summed, yielding a total penalization $\Psi$ for the violation of all linear constraints:

$$\Psi = \sum_{i} \psi_{\theta_i, \theta_j} \theta_i, \theta_j$$

(4.8)

4.1.3 Safeguard for Numerical Integration Errors

One additional issue should be addressed when defining the objective function for this particular problem. Since the output from the model is obtained via numerical integration of a system of ODEs, it should be considered that the numerical procedure may not be able to perform soundly, yielding spurious results, and thereby rendering the value of the objective function impossible to compute.

Considering that the values of the parameters which cause this kind of behavior on the part of the numerical solver constitute poor estimates of the parameters, it is safe to apply a penalization in the instances when the solver fails to provide results for the model’s predictions. When this happens, the values of the concentrations, which would normally have been computed by numerical integration, are
given the values of $10^6$ (g/L) at each time instant, yielding a very high value for the sum of squares, and thus driving the search away from the particular combination of parameters responsible for the failure. On a more basic level, this makes the objective function failsafe, since it will always be able to return a value to the optimization algorithm. If this safeguard were not included, the failure to perform the integration would result in the abrupt and premature halt of the algorithm.

### 4.1.4 Outline of the Objective Function

The objective function, implemented in MATLAB®, takes, as its sole argument, a vector $\mathbf{\theta}$ of estimated parameters, and returns an error, which measures how well the model, when substituted with these parameters, fits the experimental data. The error is computed as follows:

i) The penalization $\Psi$ for the violation of the linear constraints is applied.

ii) The estimated parameters, along with the corresponding experimental initial conditions, are used to numerically solve the system of ODEs describing the model (using the MATLAB® routine `ode15s`). This is done $N$ times, using the $N$ sets of initial conditions made available by each of the $N$ replicates in the data group considered. $N$ solutions for the $N$ systems are obtained, each consisting of the concentrations of the 8 species at the 10 time instants. For each of the $N$ instances of the system, if the numerical solver fails to compute a solution, the safeguard defined in section 4.1.3 is applied.

iii) The sum of squared differences between the experimental data and the model's predictions is computed.

iv) The penalization $\Phi$ for negative concentrations is applied.

v) Finally, the error components in i), iii) and iv) are all summed, with the total error $E$ being

$$E = OF\mathbf{\theta} = \sum_{i=1}^{N} \sum_{t=1}^{10} \sum_{s=1}^{8} y_{ist} - f^i_{t} - t_{yi} - y_{hi}, \ldots, \mathbf{\theta}^2 + \Phi + \Psi$$  \hspace{1cm} (4.9)

### 4.2 Analysis of Experimental Data

The experimental data used in this work were obtained from a series of experiments described in the paper by Duarte et al. [60]. Seeking to ascertain the effects of three putatively inhibitory substances in the physiology of *D. hansenii*, the authors ran a total of 22 experimental assays (including replicates), from which two groups were chosen for this work. The first group, group I, corresponds to three experiments carried out in the same experimental conditions, with no inhibitory substances present (reference assay), while the second group, group II, refers to four experiments, also carried out in the same experimental conditions, but with all three inhibitory substances present. They both can be found in Appendix A. Although a fourth replicate was available for the first group, it was not considered due to the atypical time course profiles observed for some of the substances. The remaining experimental conditions were discussed in the previous chapter. Because the present optimization problem deals with fitting these experimental data to the predictions of a mathematical model, one must obtain a measure of the variability or spread of the former, in order to provide a criterion of

---

When faced with the concern over this problematic replicate, the first author of this paper supported the decision of leaving it out of the analysis conducted in this work, since no plausible biological explanation was available for the extreme behavior observed.
success to the optimization procedure. Naturally, this measure should be comparable to the measure being minimized during the optimization, that is, to the output of the objective function. Such is the aim of this section.

4.2.1 A Measure for Data Spread

The measure chosen to assess experimental data variability in this work was based on the sum of squared pairwise differences between equivalent data points, as discussed by Heffernan [68]. The sum of squared pairwise differences constitutes, like the variance or standard deviation, a measure of the spread in a finite data set, but while the latter compares each sample value to the mean of the sample, the former compares all the pairs of distinct sample values. Heffernan defined this measure, \( D \), for a sample \( X = x_1, x_2, \ldots, x_n \) as follows:

\[
D = \left[ \frac{1}{n(n-1)} \sum_{i \neq j}^{n} (x_i - x_j)^2 \right]^{1/2} \quad (4.10)
\]

It should be noted that \( D \) constitutes a more pessimistic estimate of the spread of the data than the standard deviation \( S \) (in fact, it can be shown that \( D = \sqrt{2}S \)). The modified version of \( D \) used in this work, henceforth designated SSPD, does not involve normalization with respect to the sample size, nor does it take its square-root. For a sample \( X = x_1, x_2, \ldots, x_n \), the SSPD is defined as follows:

\[
SSPD = \sum_{i < j}^{n} (x_i - x_j)^2 \quad (4.11)
\]

In the present case, the \( N \) replicates in each data group yield samples of size \( N \) for each measurement instant and each measured substance concentration. There were 10 measurement instants, in each of which the concentrations of 8 species were measured, and thus, one must sum over all of the 80 samples of size \( N \), in order to obtain a global measure of the spread of the experimental data:

\[
SSPD = \sum_{T} \sum_{C} \sum_{i}^{N} (x_i - x_j)^2 \quad (4.12)
\]

where \( T \) is the set of measurement instants, \( C \) the set of measured species, \( N \) is the number of replicates (\( N=3 \) for group I and \( N=4 \) for group II), and \( x_i \) is the concentration of a given species, at a given instant, in replicate \( i \). The SSPD provides, for each data group, an estimate of the spread of its data. However, it should be more informative to obtain a distribution for the SSPD, characterized by a mean value and a standard deviation. Typically, in order to do so, one would have to obtain additional comparable data groups, compute the SSPD for each of them, and gather the statistics of interest. Since additional data are not available, these statistics may be computed in a very simple and reasonably accurate fashion, by using a technique called bootstrapping.

4.2.2 Bootstrapping the SSPD

The bootstrapping is a modern, computer-intensive approach to statistical inference, belonging to the class of resampling methods, and which is able to provide estimates that would otherwise be difficult to obtain. In the context of bootstrapping, the statistical properties of an estimator are obtained (or
estimated) by sampling from an approximate distribution, which is usually the empirical distribution of the observed/experimental data.

Under the assumption that the population sampled is i.i.d., this is done by building a number of artificial resamples of the experimental data, in which each sample is randomly drawn (with equal probability) from the original data, and then placed again in the original pool of experimental data (random sampling with replacement). The many resamples constitute, therefore, artificial samples of the population, or samples that might have been observed, had more samples been obtained (more experiences been performed in the present case). They allow one to calculate several values for the same statistic (here, the SSPD) and hence obtain a notion of its distribution. In the present case, the samples, in the sense of the previous paragraph, were considered to be not the experimental concentrations themselves, but the individual squared pairwise differences between equivalent experimental points, \((x_i - x_j)\). The number of distinct squared pairwise differences for each data group is computed by multiplying the number of points in each replicate (always 80), by the number of pairs of distinct replicates (2-combinations) that may be formed within the group. For a group containing \(N\) replicates, the number of \(k\)-combinations is

\[
C^N_k = \binom{N}{k} = \frac{N!}{k! (N-k)!}
\]  

(4.13)

Given \(k=2\) and \(N\) equals 3 or 4, respectively, for groups I and II, one concludes that there are 3 such combinations for group I, and 6 for group II. The total number of different pairwise differences that must be summed in order to obtain the SSPD is thus 240 for group I, and 480 for group II. However, when performing the optimization, the pairwise differences between the predicted concentrations and the experimental ones are 3 in the case of group I, but only 4 (not 6) in the case of group II. Thus, for group II, the experimental SSPD and the error value returned by the OF are not directly comparable. This can be easily fixed if one only takes 320 (and not 480) pairwise differences from the experimental pool of 480 such differences. In general, one should take \(N\times80\) samples with replacement from a pool with a total of \(C^N_k \times 80\) squared pairwise differences available. The bootstrapping was implemented via an iterative cycle as follows:

i) A variable, SSPD, corresponding to the SSPD of each resample was initialized to zero;

ii) For a total of \(N\times80\) times, a squared pairwise difference was randomly chosen, with uniform probability, from the pool of \(C^N_k \times 80\) experimental samples, and its value summed to the current value of SSPD, thereby updating it.

At the last iteration, the value of the variable SSPD is the value of the SSPD for the particular set of differences from which it was formed. The above procedure was repeated \(10^5\) times, recording each value obtained for the SSPD, and computing, in the end, the mean and standard deviation of the \(10^5\) SSPD samples obtained. The greatest advantage of bootstrapping over other statistical analytical methods is its great simplicity. Also, with ever increasing computational power, it becomes easier to produce the very large number of bootstrap samples needed to reliably obtain the statistical properties of an estimator. Its main disadvantage is that it tends to produce optimistic results, perhaps due to the fact that important assumptions about the data are implicitly being made (concerning e.g. the independence of samples). In the present case, for example, the pairwise differences relative to a given species may not be independent from the pairwise differences relative to the other species.
5 The Optimization Algorithms

This chapter describes the algorithms used to solve the optimization problem, as well as the meta-
optimization carried out in order to optimize their performance.

Section 5.1 describes the genetic algorithm. After an overview of its overall operation is presented, two
distinct subsections examine the detailed settings used for its operating parameters: the first
subsection deals with operating parameters which were summarily chosen and kept fixed, and the
second deals with operating parameters which were subject to optimization, in the context of the
tuning procedure. Section 5.2, describing particle swarm optimization, has an analogous organization.

Section 5.3 discusses the tuning procedure, or meta-optimization of the algorithms, in terms of the
methodology followed and the measures of algorithmic performance used.

5.1 The Genetic Algorithm

The genetic algorithm used in this work was implemented using the Genetic Algorithm and
Direct Search Toolbox™ (GADS) available in the commercial software program MATLAB®. This
toolbox has a rather complete and flexible implementation of the simple genetic algorithm, which
allows the user to write its own custom functions for the main GA operators (selection, crossover and
mutation), while also providing built-in implementations of the most commonly used functions for these
operators. A remarkable limitation is that, contrary to what happens in the classic GA algorithm,
mutation is not performed after crossover but alternatively to it.

A basic call to ga, the main function of the GADS toolbox, has the following syntax:

\[
[x \text{ fval}]=\text{ga}(\text{@objectiveFunction},\text{nvars},\text{options})
\]

The values returned are \(x\), the solution vector containing the parameters found by the algorithm, and
\(\text{fval}\), the objective function value corresponding to those parameters. One may also ask the
algorithm to output, for example, the reason why the optimization terminated or the seeds it used for
random number generation in the course of the optimization (the possession of which makes the
simulation reproducible). The input parameters are a handle to the user-defined objective function,
@objectiveFunction, the number of parameters or variables of the problem, \(nvars\), and a
structure options containing the settings or operating parameters of the algorithm (if options is not
specified the default settings are used). The options structure contains specifications such as
population size, stopping criteria and display mode, which shall be described in more detail in the next
subsections.

The flow chart in Figure 5-1 graphically illustrates the overall operation of the algorithm. Firstly, the
algorithm creates a random initial population, uniformly distributed over the allowed range of values for
the parameters, which is passed to the algorithm through the options structure. Once the initial
population is created, the loop of iterations (i.e. generations) of the algorithm begins.
At each generation, each individual in the current population is assigned a score (or fitness value), as evaluated by the objective function. The selection function then chooses the parents of the next offspring, according to these scores. The composition of the new population is determined by two parameters: 1) the number of elite children, which are, by definition, the best individuals in the current population which are to survive intact onto the next population, and 2) the crossover fraction, which is the fraction of the new population that is to be generated by crossing over parents from the current population. Mathematically, if \( n \) is the size of the population, \( x \) is the number of elite children and \( a \) is the crossover fraction, then the number of crossover children, \( y \), is

\[
y = \text{round} \ a \times n - x
\]

and the number of mutation children, \( z \), is, therefore

\[
z = n - x - y
\]

The number of elite children is constant and equal to 1 across all simulations, while the crossover fraction is one of the parameters to be tuned. The total number of parents, \( m \), needed to form the next generation is

\[
m = 2y + z
\]

It is important to notice that any one individual may be selected as a parent more than once, and may reproduce by different methods (crossover or mutation), the chances of this happening increasing with increasing fitness. Also, the best individual in the current population, that is, the best solution found so far, is never discarded, as it constitutes the elite child in the new population, independently of the selection process.

The crossover and mutation operators generate the new offspring, making sure that every individual falls within the allowed range of values for the parameters. With the new population fully formed, the algorithm checks if any of its stopping criteria has been met. The main stopping criterion of the GA is the number of generations the algorithm is allowed. Additionally, a maximum number of generations
the algorithm is allowed without showing improvement, the stall generations limit, was defined (see section 5.1.1.4). If a stopping criterion has been met, the algorithm stops and returns the solution to the problem, that is, the individual with the best (minimum) fitness value in the current population. If not, a new population is created and the loop continues until a stopping criterion is met.

In the next subsections, each major feature of the genetic algorithm is described in great detail, with the settings which were fixed being separately considered from the ones which were subject to tuning.

5.1 Fixed Settings

5.1.1 Problem Representation

The most straightforward way to represent an individual is, in the context of this problem, as a vector of real numbers, corresponding to the parameters of the system of ODEs describing the model. This representation corresponds to a haploid population, that is, a population in which each individual has only one chromosome (and thus the individual is, for all practical purposes, the chromosome). The genes in the chromosome are the entries of the vector which defines it. With this representation, there is no need for a decoding step, since the genotype and phenotype of each individual are one and the same, i.e., the domain of the problem is the same as the domain of search of the algorithm (which is not the case with binary representation). This representation was chosen because it seems rather natural and simple, easy to implement and presents no obvious drawbacks.

A population of individuals, or candidate solutions to the problem, will therefore correspond to a matrix whose rows are the individuals. The number of rows is then the size of the population, and the number of columns is the number of parameters of the problem.

The appropriate number of individuals in the population is not as straightforward to derive as the representation of an individual. It seems that it should depend on the number of parameters to be estimated, with a larger number of parameters calling for a larger number of individuals. Indeed, since a higher-dimensional search space allows for a larger number of combinations of parameters, a larger population would, with greater probability, be able to more thoroughly span the search space. Such a population would be more diverse, and, thus, more probably contain an individual close to the optimal solution. On the other hand, the larger the population, the longer it takes for the algorithm to compute the objective function values of all the individuals, as well as to perform the operations of crossover and mutation at each generation. This will, in turn, have an effect on the total amount of time needed to find an optimal solution. Conversely, a smaller population requires less computational time at each generation, but might provide insufficient coverage of the search space, leading to an increase in the number of generations needed to find an optimal solution (or even to finding a clearly suboptimal solution, due to severe restrictions in terms of search space exploration).

Typically, genetic algorithms use population sizes ranging from 50 to 200 [69], although numbers as high as 4000 have been employed, and some authors have even found that, in particular problems, a population size of less than 16 is the most appropriate choice [70]. A reasonable rule for determining a lower bound for the population size is that there must be at least as many individuals as there are parameters. Since the models developed in this work have between 26 and 31 parameters, a population size was 150 individuals (about five times the number of parameters in each model) was chosen for all models. Given the time complexity of the objective function, with three or four instances
of numerical integration being performed per individual, a much larger population would render the algorithm too slow.

5.1.1.2 Fitness Function

The fitness function provides, by definition, a measure of how fit a given individual is, or, put in another way, how good a candidate solution to the problem is. Individuals with high fitness values thus represent good solutions, close to one’s goal. Most optimization problems, though, are usually stated in terms of function minimization, with an objective function (or cost function) providing a measure of how far from the goal one is.

The transformation of an objective function into a fitness function is not troublesome, since one simply needs to take the symmetric of the objective function and add a positive number (constant or variable), to ensure that the fitness measure is nonnegative in all instances [67]. This was, however, not necessary in the present case, since the MATLAB implementation of the GA treats the fitness function as though it was a cost function, seeking to minimize it.

5.1.1.3 Fitness Scaling and Selection Operator

The process of selecting the individuals of the current population which will get to reproduce usually comprises two steps: 1) the classification of the individuals in the population, involving the definition of the probability with which a given individual will be selected for reproduction, or the expected number of times such selection will occur, and 2) the actual sampling of the population, a stochastic procedure based on the aforementioned probabilities.

Although the probability of reproduction was, in classical GA literature, computed in a fitness proportionate manner, numerous works [71;72] have indicated that scaling the fitness values is a far better choice. This scaling helps control the selection pressure over the course of the algorithm, preventing the top individuals from taking over the population early on in the simulation. The scaling method adopted in this work is based on the rank of the individuals in the population [73;74].

This very common scaling method constitutes the default scaling method in the GADS toolbox, and is implemented by the function fitscalingrank. This function sorts the raw scores of the individuals, thereby ranking them. The rank of the fittest individual, i.e. the one with the lowest fitness value, is taken to be 1, while the rank of the less fit individual corresponds to the size of the population. The individuals’ ranks are then scaled, yielding an expectation value for each individual, which corresponds to its expected number of children. If \( m \) is the number of parents for the next generation and \( n \) is the size of the population, then the expectation of an individual of rank \( i \) is

\[
\text{expectation}_i = m \times \frac{1}{\sqrt{i} \times \sum_{i=1}^{n} \frac{1}{\sqrt{i}}}
\]

(5.4)

Rank fitness scaling removes the effect of the spread of the scores, responsible in many cases for the premature convergence of the algorithm. In fact, were the scores not scaled, in the early generations a small portion of highly fit individuals would dominate the population and be granted a large number of offspring, while the rest of the population would not be allowed to reproduce, thus preventing further exploration of the search space. This type of scaling preserves the population diversity by slowing down selection pressure (the degree to which highly fit individuals are allowed many offspring), with
the possible undesired side-effect of also slowing down the finding of highly fit individuals. Nevertheless, the increase in diversity attained through ranking usually leads to more successful results than if the individuals were scaled in a fitness-proportionate manner [73;75]. Although it normally achieves a faster convergence rate, fitness-proportionate selection (and other methods which result in a high degree of selection pressure) puts the algorithm at a higher risk of premature convergence.

After fitness scaling, the selection operator takes the expectation values of each individual, and uses them to select the parents for the next generation (in what constitutes the sampling step of the selection process). Since the expectation of an individual is the expected number of children it will produce, the number of times an individual is selected as a parent is, of course, determined by this expectation value.

The method chosen for the sampling process was the so-called roulette wheel selection (RWS). This very common selection operator assigns a slice of a circular roulette wheel to each individual, the size (area) of which is proportional to the individual’s expectation. It then runs a series of selection trials (as many as the number of parents to be selected), on each of which each individual has a chance of being selected that is proportional to its expectation. In practice, RWS, implemented in the GADS toolbox function selectionroulette, creates a vector (the wheel), containing the cumulative sums of the expectations, divided, for normalization purposes, by the total number of parents. In each trial, a uniformly distributed random number between 0 and 1 is generated, and each entry in the wheel vector is checked, in order, until it is verified that this number is less than the current entry. The current entry then becomes a parent and a new trial ensues until all parents have been created.

Two other popular selection methods are stochastic uniform sampling (SUS) and tournament selection. In the first method, a line is laid out, and each parent is made to correspond to a section of the line, of length proportional to its expectation. The algorithm moves along the line in steps of equal size, and allocates, at each step, a parent from the section it lands on (with the first step being a uniform random number less than the step size). One may visualize this method as using a roulette wheel that possesses as many pointers as the number of parents. The advantage of this method is that it overcomes an important caveat of RWS: although RWS statistically results in each individual reproducing in accordance with its expectation, when it is applied to small populations (or, more accurately, when a small number of parents are being selected) that may not actually happen. SUS ensures that an individual i will reproduce at least \[ \lfloor \text{expectation}_i \rfloor \] times but no more than \[ \lceil \text{expectation}_i \rceil \] times\(^5\)[75], and so an individual of lower rank will never reproduce more than an individual of higher rank. Despite being highly recommendable when dealing with small populations, it was empirically verified that the impact of using SUS as opposed to RWS was not very significant in the present case, where the population contains 150 individuals and the number of parents ranges from 194 to 283, depending on the crossover fraction used (see equation (5.3)).

The other alternative method, tournament selection, involves two simple steps, performed on each of the parent-selecting trials: 1) a set of two or more individuals, the players in the tournament, is

\(^5\)The symbols \( \lfloor x \rfloor \) and \( \lceil x \rceil \) denote, respectively, the result of applying the functions floor and ceiling to \( x \). Specifically, floor\((x)\) is the largest integer not greater than \( x \) and ceiling\((x)\) is the smallest integer not less than \( x \). For example, for \( x=2.3 \), floor\((x)\)=2 and ceiling\((x)\)=3.
randomly picked from the population (with equal probability), and 2) the best individual among them is
granted parenthood. Tournament selection suffers from the same sampling errors as RWS [72], due to
the fact that each tournament is carried out individually and is therefore independent from every other.

5.1.1.4 Stopping Criteria
The main stopping criterion for the GA was the number of generations, \(N\), which was set to 667, for
reasons explained in section 5.3. No time criteria were defined but one other important threshold was
set: if, at any point in the optimization, it was verified that, for the last one fourth of the total number of
generations (167), the best objective function value had not changed by an amount greater than or
equal to \(10^{-25}\), the optimization was halted. This is mostly a practical criterion, designed to detect
premature convergence and help to save computational time by terminating the simulations for which
it is fully settled (according to the one-fourth-of total-number-of-generations criterion).

5.1.2 Settings subject to Tuning

5.1.2.1 Crossover Operator and Crossover Fraction
Crossover is arguably [76] the most important operator in genetic algorithms. According to the building
block hypothesis, the recombination of portions of pairs of chromosomes throughout the GA run will
result in the creation of increasingly fitter individuals. The most common methods of crossover,
classically used for bit string representations, are single-point crossover, two-point crossover and
uniform crossover. In the first method, a single crossover point is selected on both parents, and the
two resulting sets of genes in either parent are swapped between the two of them, resulting in the
creation of two new individuals (the children). The second method is analogous to the first one, with
two crossover points being selected instead, and the genes between them in either parent being
swapped (again, two new individual are formed). In the last method, the genes of each parent are
compared one by one and independently swapped with a given probability (typically 0.5).

These crossover schemes are all easily adaptable to real coded genetic algorithms, but for real valued
representations of the individuals, several other schemes of crossover have been developed [77],
which numerically combine the entries in the parents. Some of them even take into account which
parent presents the best fitness value, when computing the resulting child (or children). In this work,
two crossover operators were tested: the aforementioned single-point crossover and Wright's heuristic
crossover [78] (as described by Herrera [77]), both implemented in MATLAB® GADS toolbox.

In the case of single-point crossover, the only modification in regard to the description given above is
that a pair of parents only produces one child, not two. In practice, the parents are arbitrarily labeled,
\(e.g.\) with numbers 1 and 2, and the child is composed of the entries in parent 1 up to the crossover
point, and of the entries in parent 2 beyond that point.

In heuristic crossover, the scores of the parents determine how the child is formed. If parent 1, \(p_1\), is
fitter than parent 2, \(p_2\), than the child, \(c\), is formed as follows

\[
c = \lambda \cdot p_1 - p_2 + p_2
\]

This places the child on a line containing the two parents, a small distance away from the parent with
the best fitness value, \(p_1\), and in the direction away from the parent with the worst fitness value, \(p_2\).
Parameter $\lambda$ specifies how far the child is from the fittest parent and its value was set to 1.2 (a value of 1 for this parameter results in a child which is the same as the fittest parent).

Unlike single point crossover, heuristic crossover numerically manipulates the parents, and may thus end up producing a child which falls outside the range of allowed values for one or more parameters. In order to ensure that this does not happen, it was necessary to modify the original crossover function, by checking every entry in vector $c$ encoding the child, and forcing any entries that exceeded the upper or lower bounds for the corresponding parameters, to assume the values of these bounds.

In addition to defining the crossover method, one must also define the crossover fraction, that is, the fraction of the population which reproduces by means of crossover, at each generation step. The values for the crossover fraction are customarily set very high within the interval [0, 1], and especially so in the classical bit-string representation version of the GA. Since this is one of the most crucial parameters for the algorithm’s performance, three different crossover fractions (0.3, 0.6 and 0.9) were set forth and tested in the tuning procedure. The results of the tuning simulations will, therefore, either confirm the appropriateness of the customary practice of using high crossover fraction in this particular case, or deny it, suggesting instead a more balanced use of crossover relative to mutation, or even favoring the latter over the former (crossover fractions of 0.6 and 0.3, respectively).

5.1.2.2 Mutation Operator

The aim of the mutation operator is to introduce random variations into a chromosome, and it constitutes, therefore, an instrument of diversity and innovation for the GA [79]. For the classical binary string representation of the chromosome, mutation is just a bit flip that is set to occur with a certain probability, usually ranging from 0.001 to 0.01. In the case of real-coded genetic algorithms, several mutation methods are possible, being Gaussian mutation [80] one of the most commonly used and intuitive to understand.

In Gaussian mutation, the method followed in this work, each entry in the chromosome vector is added a random number, drawn from a Gaussian distribution with zero mean and standard deviation $\sigma$ (a different random number is generated for each entry). The standard deviation of this distribution thus determines how much a given entry is expected to change. Hinterding [80] experimentally found that a value of 0.013 for $\sigma$ gave good results, but it stands to reason that this standard deviation should be related to the range of values that a given parameter (i.e. gene) is allowed to take.

Gaussian mutation is implemented in MATLAB® by the function mutationgaussian, which must be supplied with two arguments: scale (or $\sigma_1$), a vector of initial standard deviations for each of the genes in an individual, and shrink, a parameter which determines how the standard deviations vary from generation to generation, according to the following formula:

$$\sigma_{i,k} = \sigma_{i,1} \times \left(1 - \text{shrink} \cdot \frac{k}{N}\right) \quad k = 1, \ldots, N, \quad i = 1, \ldots, M \quad (5.6)$$

where $k$ is the current generation, $N$ is the total number of generations, $i$ is the individual and $M$ is the total number of individuals in the population. For example, if shrink = 0.5, then the standard deviation in the last generation, $k = N$, is half of the standard deviation in the first generation.
For this work, the standard deviation in the first generation, $\sigma_{i,1}$, was taken to be half of the range allowed to each parameter $i$. As for the shrink parameter, its value was set to 1. Moreover, two distinct schemes of mutation were devised, by introducing a minor alteration to the original mutation function. The first scheme of mutation, labeled M1, is the standard one, with $\sigma$ varying from $\sigma_{i,1}$ in the first generation to 0 in the last generation, according to equation (5.6), and the values chosen for parameters scale and shrink. This is in accordance with empirical evidence suggesting that mutation should work by promoting population diversity in the early generations (thereby providing better coverage of the search space for the crossover operator to work on), and gradually decrease its influence on the algorithm, so that it serves more of a fine-tuning or local search role in the last generations [81].

In the second scheme of mutation, M2, the standard deviation only decreases until $k = N / 2$ (that is halfway through the optimization) and remains constant thereafter, that is, equation (5.6) is substituted by equation (5.7) below.

$$
\sigma_{i,k} = \begin{cases} 
\sigma_{i,1} \times \left(1 - \text{shrink} \times \frac{k}{N}\right), & 1 < k \leq \frac{N}{2} \\
\frac{\sigma_{i,1}}{2}, & k > \frac{N}{2}
\end{cases}
$$  

(5.7)

This scheme of mutation results in $\sigma$ being half of its initial value (i.e. one fourth of the allowed range of values for each parameter), for the entire second half of the optimization, instead of steadily decreasing all the way to zero. This method was designed in the hope that it would introduce more diversity in the population in the last generations, thereby remedying any instances of premature convergence that might have occurred. Because the parameters were allowed to change more in the last generations, the algorithm would be given a better chance of escaping any local minima it might have been trapped in.

Analogously to what happens with heuristic crossover, Gaussian mutation of an individual may also result in one or more parameters being assigned values that lie outside their allowed range of variation. The solution adopted here was the same as for heuristic crossover, with any violation of bounds forcing the violating parameter(s) to go back to the bound that was exceeded. Table 5-1 below shows a summary of the settings for the 12 cases that were tested in the GA’s tuning simulations.

<table>
<thead>
<tr>
<th>Case</th>
<th>Gaussian Mutation Settings</th>
<th>Crossover Function</th>
<th>Crossover Fraction</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>M1</td>
<td>Single Point</td>
<td>0.3</td>
</tr>
<tr>
<td>2</td>
<td>M1</td>
<td>Single Point</td>
<td>0.6</td>
</tr>
<tr>
<td>3</td>
<td>M1</td>
<td>Single Point</td>
<td>0.9</td>
</tr>
<tr>
<td>4</td>
<td>M1</td>
<td>Heuristic</td>
<td>0.3</td>
</tr>
<tr>
<td>5</td>
<td>M1</td>
<td>Heuristic</td>
<td>0.6</td>
</tr>
<tr>
<td>6</td>
<td>M1</td>
<td>Heuristic</td>
<td>0.9</td>
</tr>
<tr>
<td>7</td>
<td>M2</td>
<td>Single Point</td>
<td>0.3</td>
</tr>
<tr>
<td>8</td>
<td>M2</td>
<td>Single Point</td>
<td>0.6</td>
</tr>
<tr>
<td>9</td>
<td>M2</td>
<td>Single Point</td>
<td>0.9</td>
</tr>
<tr>
<td>10</td>
<td>M2</td>
<td>Heuristic</td>
<td>0.3</td>
</tr>
<tr>
<td>11</td>
<td>M2</td>
<td>Heuristic</td>
<td>0.6</td>
</tr>
<tr>
<td>12</td>
<td>M2</td>
<td>Heuristic</td>
<td>0.9</td>
</tr>
</tbody>
</table>
5.2 Particle Swarm Optimization

The implementation of the particle swarm optimization algorithm used in this work was that of the PSOt toolbox, developed by Brian Birge [82] for MATLAB®, and publicly available online⁶. As the source code was public, minor alterations were introduced in the algorithm, because some options were not given to the user (and thus could only be set inside the source code).

A basic call to pso2, the main function of the PSOt toolbox (corresponding to the original pso_Trelea_vectorized main function with a few minor alterations), has the following syntax:

```
[sol tr]=pso2('objectiveFunction',nvars,mv,rg,minmax,P,'plot',seed)
```

The outputs are vector sol, containing the solution found for the problem (that is, the values of the optimal parameters, plus the value of the objective function for those parameters), and vector tr, which contains the best objective function value found at each iteration of the algorithm.

The inputs are, in order: a string, ‘objectiveFunction’, designating the file in which the objective function may be found, the number of variables, nvars, or dimension of the problem, the maximum velocity parameter, mv, a two-column matrix, rg, containing the lower and upper bounds for the range of allowed values of each parameter, a number, minmax, which indicates whether the objective function is to be minimized or maximized, a vector P containing additional parameters, a string, ‘plot’, with the name of a function that plots optimization statistics along the simulation, and seed, which supplies the algorithm with a particular initial swarm, when one doesn’t wish to randomly generate it. Vector P is used to provide the algorithm with its most important operating parameters, as are the swarm size, the total number of epochs used to evolve the swarm, the local and global acceleration constants, the inertia function settings and the stopping criteria.

The flow chart in Figure 5-2 illustrates the operations of the algorithm.

---

⁶ See http://www.mathworks.com/matlabcentral/fileexchange/7506
The first step is the creation of a random initial swarm, with positions uniformly distributed over the range of the parameters and velocities uniformly distributed between the supplied maximum velocity and its symmetric value, \([-mv, mv]\). After the initial swarm is created the loop of iterations or epochs of the algorithm begins. At each epoch, the objective function (OF) evaluates the particles of the swarm generated in the previous epoch, comparing the values obtained for this swarm with an up-to-date record of the global best OF value and each particle’s (local) best OF value found so far. If better global and/or local bests are found, this record is updated and these values are used in the update of the velocities, according to the following equation

\[ v_{i} k+1 = \phi k v_{i} k + \alpha_{1} \left( p_{i} k - x_{i} k \right) + \alpha_{2} \left( p_{g} k - x_{i} k \right) \]  

(5.8)

where \(x_{i} k\) and \(v_{i} k\) are the position and velocity of particle \(i\) at iteration \(k\), respectively, \(\phi(k)\) is the inertia weight at iteration \(k\), \(\alpha_{1}\) and \(\alpha_{2}\) are acceleration constants, \(\gamma_{1}\) and \(\gamma_{2}\) are uniform random numbers on the interval \([0, 1]\) and \(p_{i}\) and \(p_{g}\) are particle’s \(i\) and the swarm’s best positions, found up to iteration \(k\), respectively. The bounds for the velocities are then verified, forcing any velocities that exceed the bounds to assume the value of the exceeded bound. Finally, the positions of the particles are updated, according to

\[ x_{i} k+1 = x_{i} k + v_{i} k + 1 \]  

(5.9)

The bounds on the positions are also enforced, in an analogous manner to that used in the case of the velocities. With the new swarm fully formed, the algorithm checks if any stopping criterion has been met. The main stopping criterion of the PSO is the number of epochs the swarm is allowed (see section 5.2.1.2 for more information on this), and if this number has been reached the optimization ends and the solution is returned. Otherwise, the loop continues until it is reached.

The next subsections detail the major features of the PSO, separately considering the settings which were kept fixed from the ones which were subject to optimization (tuning). It is important to notice that the particle swarm metaphor entails much simpler and straightforwardly understandable heuristics (equations (5.8) and (5.9)) than the ones used by the genetic algorithm (selection, crossover and mutation). Thus, the next subsections will turn out much shorter than the ones presented for the GA. This certainly does not mean that less importance is attributed to particle swarm optimization, but instead shows that its simplicity and relatively early stage of development, as compared to the genetic algorithm, make it easier to offer more succinct descriptions of its operation. Also, many concepts that are common to both algorithms have already been introduced when describing the GA. They required, then, a more careful and thorough exposure, but shall not now be subject to much explanation.

5.2.1 Fixed Settings

5.2.1.1 Problem Representation and Objective Function

Unlike the genetic algorithm, particle swarm optimization was actually conceived with real-coded representation in mind. Indeed, in Physics, the position of a particle is usually represented by a vector containing its spatial coordinates, and these coordinates are most naturally real values. This was, of course, the chosen representation, with the swarm thus being coded by a matrix whose number of
rows is the number of particles in the swarm (its size), and whose number of columns is the number of parameters of the problem (its dimension).

In regard to the size of the swarm, there are also great differences between the PSO and the GA, in respect to the choices usually made. In this case too, the optimal swarm size might be problem dependent, but it has been experimentally found that the algorithm’s performance is practically insensitive to the size of the swarm, for sizes in the range [20, 160] [83]. The most popular empirical study, by Carlisle and Dozier [84], suggests that a swarm of 30 particles results in the optimal tradeoff between algorithm performance and computational cost. Since the set of benchmark functions used to obtain this result included high-dimensionality problems, with a number of parameters comparable to the number of parameters of the models developed in this work (around 30), this suggestion was readily embraced.

5.2.1.2 Stopping Criteria

The main stopping criterion for the PSO algorithm was the number of iterations (or epochs), N, for which it was allowed to run. This number was set to 3333, according to a criterion explained later ahead, in section 5.3. No other time criteria were defined but an additional threshold, analogous to the one used for the GA was set: if, at a given epoch, it was verified that, for the last one fourth of the total number of epochs (here 833), the objective function value had not changed by an amount greater than or equal to $10^{-25}$, the optimization was halted.

5.2.2 Settings subject to Tuning

5.2.2.1 Inertia and acceleration parameters

The inertia weight on equation (5.8) determines how much emphasis the algorithm places on global search vs. local search: a large inertia weight favors the former while a smaller one favors the latter [83]. A very popular approach to setting the value of this weight is to linearly decrease it through the course of the PSO run (classically starting with a value of 0.9 and ending with a value of 0.4), precisely because an initially large inertia weight will promote global search early in the run, while a more modest value will increase local search ability towards the end.

Furthermore, it has been found that a bigger inertia weight leads to a solution that is less dependent on the initial positions of the swarm, as well as in an increased capability of exploiting the search space [46]. This is analogous to what was previously stated for the GA, regarding the standard deviation of the Gaussian mutation operator, as its decrease throughout the run of the GA also served the purpose of obtaining increased local search ability towards its end (mainly by not allowing big jumps to occur as probably as they could earlier on).

As with the GA, the importance of this effect was assessed in the context of the tuning procedure, by altering the iteration at which the final inertia value was reached. To that end, two alternatives were devised. Considering that the algorithm runs for a total of N iterations, 1) the inertia weight was linearly decreased from 0.9 at the first iteration to 0.4 at iteration $7^*N/8$ (that is, very close to the end of the run), remaining constant thereafter, and 2) the inertia weight was linearly decreased from 0.9 at the first iteration to 0.4 at iteration $N/8$ (that is, earlier on in the run), also remaining constant thereafter.
The tuning simulations also focused on the acceleration constants, which play a major role in adjusting the balance between local and global search. In the literature, the local and global acceleration constants are more commonly either set equal [43;83], or the local acceleration is set higher than the global acceleration [84]. Forcing the sum of the two acceleration constants, \( \alpha = \alpha_1 + \alpha_2 \), to be fixed (for reasons explained below, in section 5.2.2.2), three alternatives were tested in this work, corresponding to fractions of local acceleration, \( \alpha_1/\alpha \), equal to \( \frac{1}{2} \), \( \frac{1}{4} \) and \( \frac{3}{4} \).

5.2.2.2 PSO Convergence Behavior and the Tuning Design

The overall design of the tuning procedure for the PSO was based on the findings of several authors [43;48;50;51], who, treating the PSO as a dynamical system defined by its update rules, set forth to study the properties of this system, by relating the values of the parameters in equation (5.8) to the convergence behavior of the algorithm. They used a deterministic version of this equation, obtained by replacing the random numbers in the original stochastic version with their corresponding expected values. Thus, although the results are not strictly valid for the stochastic version, they provide insightful suggestions on the choice of parameters which will achieve the desired behavior for the algorithm.

Using the results of Trelea [50] and Lin [51], the performance of the PSO was evaluated for three different types of convergence behavior: 1) convergence with oscillation or zigzagging, 2) convergence without oscillation or zigzagging, and 3) divergence. The differentiation of these zones of behavior for the PSO is based on the values taken by two parameters: the inertia weight, and the sum of the acceleration constants.

Convergent behavior is a priori desirable, while the presence of oscillation and/or zigzagging might prove useful for escaping local minima. Thus, half of the tuning cases (six) were assigned to this zone (it is interesting to notice that the values typically chosen for the parameters of the PSO belong in this category). Two other cases were assigned to the convergent behavior without oscillations or zigzagging, and the remaining four cases were set in the divergent zone.

In the zone of convergence with oscillation or zigzagging, the inertia weight was tuned as described in section 5.2.2.1, and the three hypotheses for the fraction of local acceleration were tested. For the other zones, the inertia weight was set constant (to 1 in the divergent zone and to 0.2 in the convergent without oscillation zone), and the fractions of local acceleration tested were \( \frac{1}{2} \) and \( \frac{3}{4} \) (since preliminary results showed these were the most promising cases).

Additionally, a limit on the particles' maximum velocity was imposed on all the test cases. Its value was set very high on the majority of the cases, since no remarks concerning this limit were made in the works defining the convergence behavior of the algorithm. Hence, this velocity threshold was regarded more as a safeguard, ensuring that the stochastic version of the algorithm did not run wild. However, it was deemed interesting to find out if a stricter limit for the maximum velocity would be capable of producing better results in the zone of divergent behavior and, thus, this hypothesis was tested. Table 5-2 below shows a summary of the settings for the 12 particle swarm optimization tuning cases.
Table 5-2. Summary of settings for the PSO’s tuning simulations.

<table>
<thead>
<tr>
<th>Swarm Behavior</th>
<th>Case</th>
<th>Initial Inertia Value</th>
<th>Final Inertia Value</th>
<th>Epoch of Final Inertia Value</th>
<th>$\alpha$</th>
<th>$\frac{\alpha}{\alpha}$</th>
<th>Maximum Velocity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Convergent w/ Oscillation or Zigzagging</td>
<td>1</td>
<td>0.9</td>
<td>0.4</td>
<td>$N/8$</td>
<td>4</td>
<td>$\frac{3}{4}$</td>
<td>$10^4$</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0.9</td>
<td>0.4</td>
<td>$N/8$</td>
<td>4</td>
<td>$\frac{3}{4}$</td>
<td>$10^4$</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>0.9</td>
<td>0.4</td>
<td>$N/8$</td>
<td>4</td>
<td>$\frac{3}{4}$</td>
<td>$10^4$</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>0.9</td>
<td>0.4</td>
<td>$7^*N/8$</td>
<td>4</td>
<td>$\frac{3}{4}$</td>
<td>$10^4$</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>0.9</td>
<td>0.4</td>
<td>$7^*N/8$</td>
<td>4</td>
<td>$\frac{3}{4}$</td>
<td>$10^4$</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>0.9</td>
<td>0.4</td>
<td>$7^*N/8$</td>
<td>4</td>
<td>$\frac{3}{4}$</td>
<td>$10^4$</td>
</tr>
<tr>
<td>Convergent w/o Oscillation or Zigzagging</td>
<td>7</td>
<td>0.2</td>
<td>0.2</td>
<td>1</td>
<td>0.2</td>
<td>$\frac{3}{4}$</td>
<td>$10^4$</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>0.2</td>
<td>0.2</td>
<td>1</td>
<td>0.2</td>
<td>$\frac{3}{4}$</td>
<td>$10^4$</td>
</tr>
<tr>
<td>Divergent</td>
<td>9</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>$\frac{3}{4}$</td>
<td>$10^4$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>$\frac{3}{4}$</td>
<td>$10^4$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>$\frac{3}{4}$</td>
<td>0.4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>$\frac{3}{4}$</td>
<td>0.4</td>
<td></td>
</tr>
</tbody>
</table>

5.3 Tuning: Optimizing the Performance of the Optimization Algorithms

Tuning the operating parameters of an algorithm usually involves running a series of computational experiments, each with different sets of values for those parameters, and monitoring the performance of the algorithm on each of them, e.g. in terms of speed of convergence and quality of the optimal objective function (OF) value found. The tuning process is computationally costly for three reasons:

i) there may be a large number of operating parameters to take into account,

ii) the cardinality of the search space for each parameter may need to be high, depending on the quality required for the result, and

iii) the interactions among operating parameters are unknown and most probably nonlinear, so one cannot find the optimal parameters in a sequential manner (i.e. one cannot first optimize one parameter, then fixate its value at the optimum found and optimize the next, and so on).

For example, if there are 5 parameters to be tuned and 3 of them may take 4 different values (numeric or otherwise), while the other 2 may take 3 different values, there are a total of $4^3 \times 3^2 = 576$ combinations of parameters to be tested. If each of the 576 test simulations takes an average of 5 hours (which is about the time the simulations in this work require), and one would additionally like to obtain a sample of 10 simulations for each test case (since the algorithms are stochastic), the total time required would be $576 \times 5 \times 10 = 28800$ hours, that is, 1200 days, or 3 years. One faces, therefore, a tradeoff between accuracy and computational time, when designing a tuning experiment. As such, the tuning process usually encompasses a preliminary, coarse-scale stage, consisting of the empirical observation of the algorithm’s performance for a small amount of time and a small set of parameter combinations. Its aim is to reduce the cardinality of the search space for each parameter, and the choice of which parameter values and combinations to observe is guided by the available literature, as well as by any known theoretical relationships between the parameters. After this preliminary stage, the tuning proceeds by rigorously testing a smaller number of parameter combinations at a finer scale, evaluating full-length runs of the algorithm. This two-step method allows for the tuning to be performed in a reasonable amount of time, with sufficiently good results. Although
there are a few guidelines in the literature regarding the choice of operating parameters for these algorithms (both theoretically derived and resulting from more or less extensive computational experiments), it turns out that their performance is usually very dependent on the problem being solved, especially so in the case of the GA. One sometimes finds contradictory results between computational studies carried out in different fields of application, or between computational studies and theoretical predictions.

The evaluation of the tuning simulations is usually done in terms of the speed of convergence of the algorithm and of the OF value towards which the algorithm has converged, but not in terms of the actual values found for the parameters of the models. Since one is presently facing a minimization problem, the OF value should clearly be as small as possible, and the algorithm is said to have converged to a given OF value when the change in this value has been very small for a large number of iterations. Monitoring the convergence behavior of the algorithm allows one to detect the common phenomenon of premature convergence, a situation wherein the algorithm very quickly heads towards a value which is a local optimum (and not the global one), and becomes “trapped” at this value, stalling, and becoming unable to further explore the search space. Likewise, it can also help to identify good convergence behavior, which is characterized by a considerably steep decrease of the best OF value in the first few iterations (as the pool of candidate solutions goes from a random initial guess to a more “sensible” set), followed by a somewhat slower and smoother decrement, until stabilization is attained. This type of behavior indicates that a good balance between the two opposite driving forces behind the algorithm’s behavior (exploration and exploitation) has been attained.

The tuning of each algorithm’s operating parameters is not, in itself, the primary aim of this work. Because of this, though nine different models were developed (and hence nine different problems had to be considered), the tuning was only applied to one arbitrarily chosen model (model 3A) and one of the data groups (data group I). This was deemed reasonable, not only because of the computational time restrictions mentioned above, but also because the nine models and the two data groups are similar to each other, and so, the settings found to be optimal for a given model-data pair, should also be optimal (or close to optimal) for another such pair. The overall goal of the tuning procedure was to find out which of the two algorithms, the GA or the PSO, is more competent for the estimation of the parameters of the nine physiological models proposed in this work. However, in order to compare the performance of the two algorithms, they should be allowed to operate at the best of their abilities, and thus, the usual goal of tuning (to find the best operating parameters of an algorithm) became, in this work, its mere “side-effect”.

### 5.3.1 Tuning Methodology

The first step of the tuning process was to choose which parameters should be tuned and which should be summarily assigned the most common values in the literature. Given the time restrictions mentioned above, this is a delicate task and a great effort was made to support the choices made on literature findings that appeared to be general (or at least applicable to this particular case).

After choosing the parameters that should be subject to tuning, the next task was to define which values to test in the tuning procedure. This choice was again grounded on available literature results, followed by extensive empirical observations of several representative values. The number of values allowed to each operating parameter varied, but, in total, the tuning experiments consisted of a set of
12 computational simulations for each algorithm, where 12 sets of parameters were tested. Although this may seem a small number of cases to test, embarking in an exceedingly exhaustive search could, in addition to being too time consuming, defeat the larger purpose of the tuning procedure by over-fitting the operating parameters to the model chosen for tuning. Each of the 12 computational experiments was run 20 times on the chosen test model and data group (model 3A and data group I), departing, in each run, from the same initial pool of candidate solutions, one for the GA and another for the PSO, randomly generated on the very first of experimental runs of each of the algorithms. In order for the results from both the GA and PSO to be comparable, and, perhaps more importantly, in order to grant each algorithm an equal opportunity at solving the problem, they were both allowed a total number of $10^5$ OF evaluations to perform the optimization. Since the number of OF evaluations performed per iteration corresponds to the size of the pool of candidate solutions, and the two algorithms use pools of different size, this resulted in a total number of 667 ($10^5/150$) iterations allowed for the GA and 3333 ($10^5/30$) iterations for the PSO. Additionally, if at any time in the course of the simulation, the OF value has not changed by more than $10^{-25}$ during one fourth of the total number of iterations, the simulation immediately comes to a halt, regardless of the number of iterations attained. No stopping criterion was imposed with regard to the best OF value attained, that is, the algorithm was not directed to stop if the best OF value was within some small distance of a predefined reference value. Although such a value does exist (it is the mean experimental SSPD, defined in the previous chapter), it shall only be used in the evaluation of the tuning results, and not as a stopping criterion of the algorithms, since the OF value should be as small as possible.

5.3.2 Evaluation of Algorithm Performance

In regard to the evaluation of the algorithms’ performance in each tuning simulation, three different measures were used to rate it [85]: 1) best objective function value at final iteration, 2) number of iterations to criterion, and 3) proportion of runs reaching criterion. The first measure is simply the minimum OF value found by the algorithm after a predefined number of iterations (which in this case was 667 for the GA and 3333 for the PSO). This is a measure of sloppy speed and it does not constitute an indicator of how close to the global optimum the algorithm is [85].

The last two measures usually require a threshold-type criterion to be set, indicating that the algorithm has reached the global optimum. The algorithm is then given a certain number of iterations (possibly larger than the one used in the first measure) to reach that threshold, and after that number of iterations has expired, its success or failure is recorded, along with the number of iterations it took to reach the criterion. In the cases where the global optimum is known (e.g. in benchmark functions used to test the performance of a novel algorithm against existing ones), it is customary to set this threshold 5% to 10% above the global optimum, and the second measure is thus an indicator of how fast the algorithm reached the region of the global optimum. In the present case, however, the global optimum is not known a priori. Thus, the criterion used for these measures was the value of the mean SSPD of the experimental data against which the optimization was being run (see Chapter 4). Also, the number of iterations the algorithms were allowed was the same as in the first measure, mainly because it was observed that, after $10^5$ OF evaluations, the algorithms had, in the great majority of cases, already converged, and thus, either the criterion had been reached or it would not be reached at all. The approximations used in the last two measures make them somewhat similar to the first one, and it will be interesting to observe if all three yield the same results.
6 Results and Discussion

This section is divided into three subsections, which correspond to the results relative to 1) the analysis of the experimental variability, 2) the tuning of the algorithms, and 3) the parameter estimation for all the proposed models and their discussion and physiological interpretation.

6.1 Analysis of Experimental Variability

The SSPD of data groups I and II, as well as its mean value, \( \overline{SSPD} \), and its standard deviation, \( \sigma_{SSPD} \), obtained, by bootstrapping (see Chapter 4, section 4.2), for these data groups, are shown in Table 6-1 below. The histograms for the distributions of the SSPD in each group can be seen in Figure 6-1.

<table>
<thead>
<tr>
<th>Group</th>
<th>SSPD</th>
<th>( \overline{SSPD} )</th>
<th>( \sigma_{SSPD} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>1188.4</td>
<td>1192.6</td>
<td>367.3</td>
</tr>
<tr>
<td>II</td>
<td>879.05</td>
<td>587.21</td>
<td>112.82</td>
</tr>
</tbody>
</table>

It is worth noticing that, despite the smaller number of terms used to calculate the SSPD for group I, its mean value is larger than that obtained for group II. This is not unexpected, since a careful observation of the experimental time courses (see Appendix A) should detect higher variability in the concentrations of xylitol, for the experiments in group I. The fact that this high variability occurs at high concentrations of xylitol has a big impact on the value of the SSPD for this data group, since the SSPD is a sum of squared differences between the data. The bootstrapping histograms show that the distribution of the SSPD value has an approximate \( \chi^2 \) shape for both data groups, with the tail to the right of the mean being slightly more populated than the tail to the left.

6.2 Analysis of the Tuning of the Algorithms

The results of the tuning of the algorithms are presented in the next three sections: section 6.2.1 deals with the results for the GA, section 6.2.2 with the results for the PSO and, finally, in section 6.2.3, the results for both algorithms are compared. Based on that comparison, a single algorithm, with its
corresponding set of optimal parameters, was deemed the most appropriate for parameter estimation and used, to that end, on all of the proposed models.

The performance measures used to evaluate the tuning were defined earlier in Chapter 5. Since only the data in group I were used in the tuning, the criterion (or error criterion) to which some of these measures refer (fourth and fifth columns in Table 6-2 and Table 6-3), corresponds to the mean of the SSPD for this data group, shown in Table 6-1, in the section 6.1.

6.2.1 Tuning of the GA

The results for the tuning of the genetic algorithm are shown in Table 6-2 below. The numbering of the tuning cases in this table corresponds to that defined in Chapter 5, Table 5-1. The analysis that follows separately examines the influence of each of the three tuning variables on the performance of the algorithm.

<table>
<thead>
<tr>
<th>Case</th>
<th>Best</th>
<th>Mean</th>
<th>Percentage Reaching Criterion</th>
<th>Iterations to Criterion (average)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1940.8</td>
<td>3241.1</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>1190.3</td>
<td>1845.4</td>
<td>15</td>
<td>592.33</td>
</tr>
<tr>
<td>3</td>
<td>1176.9</td>
<td>1653.6</td>
<td>50</td>
<td>545.4</td>
</tr>
<tr>
<td>4</td>
<td>1920</td>
<td>2579.1</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>1653.6</td>
<td>2322.1</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>6</td>
<td>1534.3</td>
<td>2598.9</td>
<td>5</td>
<td>650</td>
</tr>
<tr>
<td>7</td>
<td>2372.4</td>
<td>3004.9</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>8</td>
<td>1586.7</td>
<td>2243.8</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>9</td>
<td>1191.9</td>
<td>2043.4</td>
<td>10</td>
<td>395</td>
</tr>
<tr>
<td>10</td>
<td>2017.5</td>
<td>2884</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>11</td>
<td>1775.7</td>
<td>2609.7</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>12</td>
<td>1630.6</td>
<td>2429.1</td>
<td>0</td>
<td>-</td>
</tr>
</tbody>
</table>

6.2.1.1 Standard Deviation of the Gaussian Mutation Function

Cases 7 to 12 in Table 6-2, for which the standard deviation used in the Gaussian mutation function was kept constant in the second half of the run, perform, in general, much worse than cases 1 to 6, for which there was a steady decrease of the standard deviation along the run, until a value of zero was reached at the last generation. In fact, only when using single point crossover with a crossover fraction of 0.9 did 10% (2) of the runs satisfy the error criterion. It thus stands to reason that the increased interval of mutation, developed with the aim of obviating premature convergence, had, on the contrary, a negative effect on the algorithm's performance. This effect is probably due to the fact that a larger mutation range makes the search process more random and “jumpy”, preventing it from following a coherent path along the fitness landscape. It would be interesting to test if the original goal could be attained by decreasing the standard deviation during the first 3/4 or 4/5 of the run, instead of the first half.

6.2.1.2 Crossover Fraction

Cases 1, 4, 7 and 10 in Table 6-2, where the lowest crossover fraction (0.3) was used, were, in general, the poorest performers, in terms of both best and mean objective function value found. Moreover, none of these cases ever reached the error criterion. On the other hand, cases 3, 6, 9 and
12 (where the highest crossover fraction of 0.9 was used) were almost always the best performers. The only exception was in cases 4 and 6, where the mean value of the objective function over the 20 runs was slightly better for the lowest crossover fraction than for the highest, and both were worse than the crossover fraction of 0.6. Therefore, in this particular problem, the role of the crossover operator is crucial for the performance of the genetic algorithm and a high crossover fraction of 0.6-0.9 is preferable to a lower one.

6.2.1.3 Crossover Function

Regarding the crossover function, it was found that, for the instances with highest crossover fractions, 0.6 and 0.9, single point crossover always outperformed heuristic crossover, in terms of best and mean objective function values, and gave equal or better results, in terms of percentage of runs reaching the criterion. For the lowest crossover fraction, 0.3, heuristic crossover (cases 4 and 10) did better than single point crossover (cases 1 and 7), but none of the corresponding runs ever reached the predefined criterion. Bearing in mind that:

a) heuristic crossover numerically combines the parents, thereby producing a child who is effectively closer to the best parent, whereas single point crossover only brings together portions of the parents’ chromosomes and does not alter each individual gene, and

b) for the cases where a crossover fraction of 0.3 was used, 70% of the individuals were created by mutation at each generation,

this phenomenon may simply be an indication that the ability of heuristic crossover to use the fitness information, while operating on a population of highly heterogeneous individuals, resulted in better performance.

6.2.1.4 Best Settings

The best tuning case was, by far, case 3, where single point crossover was used, with a crossover fraction of 0.9, and the classically decreasing standard deviation in the Gaussian mutation function. The best objective function value found was 1176.9 and 50% of the runs went below the error criterion of 1192.6.

6.2.2 Tuning of the PSO

Table 6.3 below summarizes the results obtained for the tuning of the PSO algorithm, with the numbering of the tuning cases corresponding to that defined in the Chapter 5, Table 5.2.

<table>
<thead>
<tr>
<th>Swarm Behavior</th>
<th>Case</th>
<th>Best</th>
<th>Mean</th>
<th>Percentage Reaching Criterion</th>
<th>Iterations to Criterion (average)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Convergent w/ Oscillation or Zigzagging</td>
<td>1</td>
<td>1064</td>
<td>1831.9</td>
<td>15</td>
<td>272</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>841.66</td>
<td>1411.1</td>
<td>60</td>
<td>839.5</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>1002.5</td>
<td>1862.6</td>
<td>20</td>
<td>195.5</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>1186.9</td>
<td>2498.6</td>
<td>5</td>
<td>1444</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>922.38</td>
<td>1574.6</td>
<td>40</td>
<td>1721.3</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>1186.9</td>
<td>1966</td>
<td>15</td>
<td>886</td>
</tr>
<tr>
<td>Convergent w/o Osc. or Zigzagging</td>
<td>7</td>
<td>5572.3</td>
<td>10102</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>6277.1</td>
<td>10282</td>
<td>0</td>
<td>-</td>
</tr>
</tbody>
</table>
These results show that the three types of swarm behavior tested resulted in three differentiated ranges of values for best and mean objective function values. In the analysis that follows, the performance of the algorithm is related to each of the three represented types of swarm behavior, as well as with the fraction of local acceleration used.

6.2.2.1 Convergent Behavior with Oscillation or Zigzagging and the Effect of the Inertia Weight

Cases 1 to 6, where the swarm behavior is convergent with oscillation or zigzagging, were those for which better performances were achieved. While there are performance differences among these cases, they are all better than the cases 7 to 12, for which not only were the objective function values higher, but also none of the 20 runs in each case reached the criterion.

Within cases 1 to 6, the ones for which the inertia weight reached its final value earlier in the run (cases 1 to 3), performed better than the ones for which the inertia weight reached its final value later in the run (cases 4 to 6). In the latter cases, the best and mean objective function values found were higher, the percentage of runs reaching the criterion was smaller, and a greater number of iterations were needed to reach it. This suggests that, in this particular problem, the PSO performs better when the emphasis is placed on exploitation (i.e. local search, favored by a smaller inertia weight) rather than on exploration.

6.2.2.2 Convergent Behavior without Oscillation or Zigzagging

The worst tuning results were those obtained for cases 7 and 8, representing convergent behavior without oscillation. These cases were worse than the ones in the divergent behavior category (cases 9 to 12), probably because the algorithm prematurely converged to a local minimum, and was not able to escape that area of the search space, owing to its extremely low acceleration constants (see Table 5-2 in section 5.2.2.2 of Chapter 5). These cases have best and mean objective function values that are 4 to 7 times larger than those found for the best tuning cases, clearly representing a set of operating parameters that should be avoided.

6.2.2.3 Divergent Behavior and the Effect of Maximum Velocity

The zone of divergent swarm behavior resulted in a performance that was better than that of the convergent without oscillation or zigzagging behavior, but worse than that of convergent with oscillation or zigzagging behavior. Among the divergent cases, the results were better for the cases where the maximum velocity was set to a lower value (0.4 as compared to $10^4$), but were still poor when compared to cases 1 to 6. While further lowering the value of the maximum velocity might prove more effective in obtaining better performances in this divergent zone, it could also limit the swarm’s ability to escape local minima [86].

6.2.2.4 Fraction of Local Acceleration

In general, and for all swarm behaviors, the cases for which the local acceleration constant was greater than the global acceleration constant were better than those for which they were set equal.
The latter were, in turn, better than those for which the local acceleration constant was smaller than the global acceleration constant (with the exceptions of cases 4 and 6, for both of which the best objective function value found was the same). The conclusion is, once again, that exploitation (local search ability) should be favored over exploration, in order for the PSO to attain better performances.

6.2.2.5 Best Tuning Case

The best overall case was case 2, where the behavior of the algorithm is convergent with oscillation or zigzagging. In this case, the inertia weight reaches its final (lowest) value early in the run and the local acceleration constant is three times larger than the global acceleration constant. With these settings, 60% of the runs reached the criterion and 841.66 was the best objective function value found.

6.2.3 Comparison between GA and PSO

The results obtained for cases 1 to 6 in Table 6-3, referring to the PSO tuning results for a convergent swarm behavior with oscillation or zigzagging, were, in general, much better than the best GA results - much lower best and mean objective function values were reached and higher percentages of runs reached the criterion. The average number of iterations to the criterion was, in some cases, higher for the PSO, but the time the algorithm took to compute them was not, since:

a) because the number of candidate solutions processed by the PSO at each iteration is about 5 times smaller than that processed by the GA, the PSO is about 5 times faster than the GA, and

b) the average number of iterations to reach the criterion for the PSO was, in the worst case (out of cases 1 to 6), only about 3 times greater than it was for the GA’s best case (case 3, with an average of 545.4 iterations to criterion).

Cases 7 and 8 of the PSO tuning were, on the other hand, much worse than the worst GA results, whereas divergent cases (9 to 12) were poor, but comparable to the GA’s poorest results.

The overall conclusion of the tuning simulations is that the PSO is more appropriate than the GA for solving this problem, since, according to the criteria examined, and for the best tuning settings, the former yielded significantly better results than did the latter. This conclusion agrees with the results obtained by Drager et al [54], for the estimation of parameters of similar models (metabolic models described by systems of ODEs with a comparable number of parameters), as they also favor the use of the PSO (albeit in a slightly different variant) to the use of the GA (with settings similar to the present ones).

The parameter estimation procedure therefore relied on particle swarm optimization, with operating parameters given by tuning case 2 (see Table 5-2 in Chapter 5), to find the parameters of all the proposed network models.

6.3 Parameter Estimation Results

This section analyzes the results of parameter estimation for each of the proposed models and is divided in three parts. In section 6.3.1, the performance of the PSO algorithm for each model is examined, in a manner similar to that used for the tuning of the algorithms, i.e., in terms of best and mean objective function values achieved, and percentage of runs reaching the error criterion.
In section 6.3.2, each model’s best run is compared to the experimental data, both in terms of objective function value, and in terms of the goodness-of-fit of the concentrations predicted.

Finally, in section 6.3.3, the parameters found for the best model are analyzed, and a comparison is made between the values obtained for data group I and data group II.

### 6.3.1 Analysis of PSO Performance

The results of the parameter estimation procedure are given in Table 6-4, in terms of the best and mean objective function values found, as well as of the percentage of runs which reached the criteria established for groups I and II (shown in Table 6-1).

<table>
<thead>
<tr>
<th>Model</th>
<th>Data Group I</th>
<th>Data Group II</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Best</td>
<td>Mean</td>
</tr>
<tr>
<td></td>
<td>1A</td>
<td>1106.1</td>
</tr>
<tr>
<td></td>
<td>1B</td>
<td>840.85</td>
</tr>
<tr>
<td></td>
<td>1C</td>
<td>1006.7</td>
</tr>
<tr>
<td></td>
<td>1D</td>
<td>911.49</td>
</tr>
<tr>
<td></td>
<td>2A</td>
<td>1113.6</td>
</tr>
<tr>
<td></td>
<td>2B</td>
<td>966.05</td>
</tr>
<tr>
<td></td>
<td>2C</td>
<td>1119.9</td>
</tr>
<tr>
<td></td>
<td>3A</td>
<td>997.34</td>
</tr>
<tr>
<td></td>
<td>3B</td>
<td>922.28</td>
</tr>
</tbody>
</table>

Before a more detailed analysis is made, it is interesting to observe that the results obtained here for model 3A in data group I, were not as good as those obtained for this same model and data group in the PSO’s tuning simulations (see Table 6-3). This may be attributed to the fact that the single initial population used for the tuning simulations was, by chance, more favorable to this model, whereas the use of multiple random initial populations, as was the case in the second part of this work, cancelled that chance effect. On the other hand, models 1B, 1D and 3B were able to achieve, in terms of best objective function value found, results comparable to the ones achieved by model 3A in the tuning simulations. This indicates that there was no over-fitting of the operating parameters of the algorithm to the particular model used for tuning.

It is apparent, from the results on Table 6-4, that, in terms of meeting the error criterion based on experimental variance (the SSPD), the optimization gave better results for the data on group I, as compared to the data on group II. While in the former, the percentage of runs reaching the criterion varied from 10% to 45% (2 to 9 runs), depending on the model, in the latter, only two of the models (1C and 2C) ever reached the criterion, and, in each case, in only 1 out of 20 runs. This can be explained by the smaller amount of variance in the experimental data belonging to group II, which is also responsible for the lower objective function values obtained for this group.

Since the PSO algorithm is a stochastic procedure, the total error of the optimization should be composed of a portion pertaining to the variance of the algorithm, and another portion, pertaining to the variance of the experimental data against which the optimization is carried. It can therefore be hypothesized that, in the case of data group I, the error portion related to variance in experimental
data is large, compared to the error portion related to the algorithm’s variance, while in the case of
data group II, the opposite is true, and the variance of the algorithm dominates the results.

In order to obtain a measure of the overall ability of the PSO to estimate the parameters of each of the
proposed models, the following analysis was performed:

i) For each model and each data group, the mean value of the objective function (OF) over
the runs which satisfied the criterion \( \text{SSPD} \) in Table 6-1 was computed, ensuring that at
least three values were used to obtain this mean. If, for a given model and data group,
less than three runs satisfied the criterion, the three best runs were used, independently of
criterion satisfaction.

ii) This resulted in each model being characterized by two mean OF values, \( \overline{OF}_I \) and \( \overline{OF}_II \),
referring to data groups I and II. The sum of these two means, \( \overline{OF}_{\text{total}} \), was then
computed, and the models ranked according to its value (with lower values thus indicating
better models).

The results are shown in Table 6-5 below, along with each model’s number of parameters.

\[\begin{array}{|c|c|c|c|c|}
\hline
\text{Model} & \text{No. of Parameters} & \overline{OF}_I & \overline{OF}_II & \overline{OF}_{\text{total}} & \text{Rank} \\
\hline
1A & 26 & 1162.1 & 821.66 & 1983.76 & 8 \\
1B & 28 & 1069.1 & 837.73 & 1906.83 & 6 \\
1C & 28 & 1105.3 & 758.55 & 1863.85 & 4 \\
1D & 27 & 948.84 & 679.25 & 1628.09 & 1 \\
2A & 28 & 1130.9 & 1049.7 & 2180.6 & 9 \\
2B & 31 & 1073.5 & 809.59 & 1883.09 & 5 \\
2C & 31 & 1158.7 & 699.73 & 1858.43 & 3 \\
3A & 31 & 1107.9 & 835.72 & 1943.62 & 7 \\
3B & 28 & 1020.3 & 772.07 & 1792.37 & 2 \\
\hline
\end{array}\]

According to the ranking in Table 6-5, the PSO achieved its best overall performance for model 1D,
while the worst overall performance was observed for models 3A, 1A and 2A.

The relative success of the PSO in model 1D, in comparison to models with both a higher and lower
number of parameters, may be explained by the fact that this model is more constrained. In this
context, the term constrained means that, for a species which is consumed to simultaneously form two
products, its rate of consumption is defined by a single pair of kinetic parameters, \( \mu_{\text{max}} \) and \( K_S \), while
the rate of formation of each of the products is defined by multiplying the rate of substrate
consumption by a yield parameter, \( Y \), specific to each product. A less constrained model would, in the
same situation, define two distinct rates of consumption of substrate, characterized by two distinct
pairs of kinetic parameters, and define the products’ rate of formation by multiplying each of them by
the corresponding yields.

In all of the models tested, the consumption of xylose to form xylitol and arabitol was designed to be
constrained in this way, but, in model 1D, there are three more reactions which are also constrained,
namely the rates of consumption of substrates glycerol, xylitol and arabitol, each leading to the formation of both ethanol and biomass. This thus seems to be a desirable feature, from the point of view of this optimization algorithm, for this kind of model. Since it is suggested, in some of the literature [12], that the rate of consumption of substrates which form multiple products may be modeled with a single \( \mu_{\text{max}} \) parameter, further modeling efforts should prove more successful if incorporating this kind of constraint.

6.3.2 Analysis of the Fit

This section examines, in more detail, how well each model was able to predict the concentrations observed in the experimental data. It comprises two sections: in section 6.3.2.1, the best objective function values of the models are compared to the experimental SSPD, and in section 6.3.2.2, the \( R^2 \) goodness-of-fit coefficients for the predictions of each model are discussed.

6.3.2.1 Comparison of Objective Function Values with Experimental SSPD

In Figure 6-2 below, the best OF values found for each model, on each data group, are superimposed on the histograms corresponding to the data group’s SSPD. For the sake of readability, the vertical lines corresponding to the OF values are shown with different heights; these heights thus have no meaning of their own, since the OF values are indicated in the horizontal coordinate of the plots. Additionally, there is a yellow vertical line in each plot, indicating the SSPD of the corresponding data group, and a pair of red vertical lines, marking the 2.5th-percentile and the 97.5th-percentile of the data in the histogram; these lines are the bounds of an interval within which 95% of the samples of the SSPD are expected to be contained.

In addition to the differences between the two data groups, already discussed in section 6.3.1, Figure 6-2 shows that, for both data groups, all of the models’ best OF values (with the exception of model 2A, in data group II) fell within the 95% confidence interval for the experimental SSPD. This indicates that all the models, with the exception of model 2A, are statistically equivalent, that is, from a statistical viewpoint, they are equally competent at describing the experimental data, with its associated variability.
6.3.2.2 Goodness-of-Fit of the Models’ Predicted Concentrations

For the sake of brevity, the concentrations’ plots obtained from each model’s best run are not shown here, but may be found in Appendix C. Table 6-6 and Table 6-7 below, show the $R^2$ goodness-of-fit coefficients obtained, for each species, in the best run of each model, for data groups I and II, respectively. The last row on these tables shows an average goodness-of-fit coefficient for each species, i.e. the average of the $R^2$ coefficients of all the models for that species, while the last column shows an average goodness-of-fit coefficient for each model.

The $R^2$ coefficients, or coefficients of determination, correspond to the square of the correlation coefficients between the experimental concentrations and the concentrations predicted by each model. They are a measure of the goodness-of-fit of a model because their values represent the fraction of variation in the concentrations that is accounted for by the model (e.g. a value of $R^2=0.8$ for a given species means that 80% of the variation in the concentration of that species can be explained by the model’s equations, while the remaining 20% remains unexplained).

As can be seen from both the plots on Appendix C and Table 6-6 and Table 6-7 above, virtually all species’ predicted concentrations agree very well ($R^2$ coefficients close to 1) with the experimental data on both groups, with the exception of ethanol and, especially, glycerol.

Table 6-6. $R^2$ goodness-of-fit coefficients for the concentrations of each species, obtained for data group I, with the best run of each model.

<table>
<thead>
<tr>
<th>Model</th>
<th>Species</th>
<th>X</th>
<th>Glc</th>
<th>GlyOH</th>
<th>EtOH</th>
<th>Xyl</th>
<th>XOH</th>
<th>Ara</th>
<th>ArOH</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>1A</td>
<td>X</td>
<td>0.88</td>
<td>0.98</td>
<td>0.27</td>
<td>0.31</td>
<td>0.99</td>
<td>0.85</td>
<td>0.97</td>
<td>0.85</td>
<td>0.76</td>
</tr>
<tr>
<td>1B</td>
<td>X</td>
<td>0.89</td>
<td>0.99</td>
<td>0.23</td>
<td>0.87</td>
<td>1.00</td>
<td>0.86</td>
<td>0.98</td>
<td>0.90</td>
<td>0.84</td>
</tr>
<tr>
<td>1C</td>
<td>X</td>
<td>0.91</td>
<td>0.99</td>
<td>0.41</td>
<td>0.34</td>
<td>0.99</td>
<td>0.85</td>
<td>0.98</td>
<td>0.88</td>
<td>0.79</td>
</tr>
<tr>
<td>1D</td>
<td>X</td>
<td>0.91</td>
<td>0.98</td>
<td>0.23</td>
<td>0.88</td>
<td>0.99</td>
<td>0.84</td>
<td>0.98</td>
<td>0.88</td>
<td>0.84</td>
</tr>
<tr>
<td>2A</td>
<td>X</td>
<td>0.88</td>
<td>0.98</td>
<td>0.23</td>
<td>0.26</td>
<td>0.99</td>
<td>0.85</td>
<td>0.97</td>
<td>0.82</td>
<td>0.75</td>
</tr>
<tr>
<td>2B</td>
<td>X</td>
<td>0.89</td>
<td>0.99</td>
<td>0.30</td>
<td>0.92</td>
<td>0.99</td>
<td>0.84</td>
<td>0.97</td>
<td>0.85</td>
<td>0.84</td>
</tr>
<tr>
<td>2C</td>
<td>X</td>
<td>0.89</td>
<td>0.98</td>
<td>0.20</td>
<td>0.26</td>
<td>0.99</td>
<td>0.86</td>
<td>0.98</td>
<td>0.59</td>
<td>0.72</td>
</tr>
<tr>
<td>3A</td>
<td>X</td>
<td>0.88</td>
<td>0.99</td>
<td>0.35</td>
<td>0.91</td>
<td>0.99</td>
<td>0.84</td>
<td>0.98</td>
<td>0.89</td>
<td>0.85</td>
</tr>
<tr>
<td>3B</td>
<td>X</td>
<td>0.90</td>
<td>0.98</td>
<td>0.57</td>
<td>0.91</td>
<td>0.99</td>
<td>0.83</td>
<td>0.97</td>
<td>0.87</td>
<td>0.88</td>
</tr>
<tr>
<td>Average</td>
<td></td>
<td>0.89</td>
<td>0.98</td>
<td>0.31</td>
<td>0.63</td>
<td>0.99</td>
<td>0.85</td>
<td>0.98</td>
<td>0.84</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 6-7. $R^2$ goodness-of-fit coefficients for the concentrations of each species, obtained for data group II, with the best run of each model.

<table>
<thead>
<tr>
<th>Model</th>
<th>Species</th>
<th>X</th>
<th>Glc</th>
<th>GlyOH</th>
<th>EtOH</th>
<th>Xyl</th>
<th>XOH</th>
<th>Ara</th>
<th>ArOH</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>1A</td>
<td>X</td>
<td>0.98</td>
<td>0.98</td>
<td>0.25</td>
<td>0.58</td>
<td>0.99</td>
<td>0.94</td>
<td>0.99</td>
<td>0.89</td>
<td>0.84</td>
</tr>
<tr>
<td>1B</td>
<td>X</td>
<td>0.98</td>
<td>0.99</td>
<td>0.23</td>
<td>0.75</td>
<td>0.99</td>
<td>0.94</td>
<td>0.98</td>
<td>0.92</td>
<td>0.85</td>
</tr>
<tr>
<td>1C</td>
<td>X</td>
<td>0.99</td>
<td>0.99</td>
<td>0.28</td>
<td>0.58</td>
<td>1.00</td>
<td>0.97</td>
<td>0.98</td>
<td>0.92</td>
<td>0.84</td>
</tr>
<tr>
<td>1D</td>
<td>X</td>
<td>0.98</td>
<td>0.99</td>
<td>0.22</td>
<td>0.93</td>
<td>0.99</td>
<td>0.95</td>
<td>0.99</td>
<td>0.89</td>
<td>0.87</td>
</tr>
<tr>
<td>2A</td>
<td>X</td>
<td>0.97</td>
<td>0.99</td>
<td>0.20</td>
<td>0.00</td>
<td>0.99</td>
<td>0.94</td>
<td>0.99</td>
<td>0.88</td>
<td>0.75</td>
</tr>
<tr>
<td>2B</td>
<td>X</td>
<td>0.98</td>
<td>0.99</td>
<td>0.22</td>
<td>0.95</td>
<td>0.99</td>
<td>0.93</td>
<td>0.99</td>
<td>0.81</td>
<td>0.86</td>
</tr>
<tr>
<td>2C</td>
<td>X</td>
<td>0.99</td>
<td>0.98</td>
<td>0.34</td>
<td>0.57</td>
<td>1.00</td>
<td>0.96</td>
<td>0.99</td>
<td>0.93</td>
<td>0.85</td>
</tr>
<tr>
<td>3A</td>
<td>X</td>
<td>0.98</td>
<td>0.99</td>
<td>0.02</td>
<td>0.91</td>
<td>0.99</td>
<td>0.93</td>
<td>0.99</td>
<td>0.92</td>
<td>0.84</td>
</tr>
<tr>
<td>3B</td>
<td>X</td>
<td>0.98</td>
<td>0.99</td>
<td>0.31</td>
<td>0.58</td>
<td>1.00</td>
<td>0.96</td>
<td>0.99</td>
<td>0.88</td>
<td>0.84</td>
</tr>
<tr>
<td>Average</td>
<td></td>
<td>0.98</td>
<td>0.99</td>
<td>0.23</td>
<td>0.65</td>
<td>0.99</td>
<td>0.95</td>
<td>0.99</td>
<td>0.89</td>
<td>-</td>
</tr>
</tbody>
</table>
Comparing each table’s last column against each other, one concludes that the fit achieved for data group II was, in average, for the majority of models (with the exception of model 3A) better than or equal to the fit achieved for data group I. The worst average results were achieved by model 2A on data group II, and by models 2C, 2A, 1A and 1C, on data group I (average $R^2<0.8$).

Comparing each table’s last row against each other, one finds that the concentrations of every species were better fitted on data group II than on data group I, with the exception of glycerol. The poorer $R^2$ coefficients obtained for glycerol (and, to a lesser extent, also for ethanol) have to do with the fact that the covariance of its experimental and predicted concentrations is very small, since, in these cases, an approximately constant (zero) concentration is predicted. Nevertheless, the percentages of error pertaining to glycerol and ethanol (as shown in the plots in Appendix C) is usually low (around 2%), since their predicted concentrations do differ from the experimental ones by a small amount (in terms of absolute differences). Additionally, for models 1A, 1C, 2A, and 2C, the only route available for biomass production, in the initial 24 hours of the experiment, is through ethanol and/or glycerol consumption. Because for data group I there is a steep biomass growth rate during this period, the rates of ethanol and/or glycerol consumption must be high enough to account for it. This may result in an earlier full exhaustion of ethanol and/or glycerol, and thus be part of the explanation for the poorer fit obtained for these species.

Some curious results did however arise for data group II, in which a higher than observed accumulation of ethanol is predicted for the second half of the experiment - model 2C (on the left in Figure 6-3) and, to a lesser extent, also model 1B - , or a more or less constant ethanol concentration is predicted throughout it (model 2A, on the right in Figure 6-3).

For models 2A and 2C, these results find justification in the topology of the networks belonging to group 2. These models artificially divide ethanol into two pools: one originating from glucose metabolism, EtOH$_H$, and another from pentose metabolism, EtOH$_P$. These two pools are characterized by two different rates of consumption, as opposed to the single consumption rate defined by the rest of the models for EtOH. As stated above, the higher biomass growth rate observed in the beginning of the experiment, may (while forcing this single rate of ethanol consumption to be high enough to account for it) lead to the full exhaustion of ethanol approximately halfway through the experiment. On the other hand, when two independent consumption rates are defined, the rate of consumption of EtOH$_H$ may be set high enough to account for high initial biomass growth rate, while the one for EtOH$_P$ may simultaneously be set low enough to result in net ethanol accumulation in the second half of the experiment.
experiment, when the polyols from which it results are consumed (while also accounting for the lower biomass growth rate in this period).

From the models on group 2, only model 2B was able to achieve significantly better results, in terms of ethanol modeling, than the models outside of this group ($R^2$ coefficients of 0.92 and 0.95, on data groups I and II, respectively). On data group I, models 2A and 2C were the worst ethanol modelers of all, and, on data group II, model 2A was also the worst, while model 2C gave results comparable to those of the other models. The conclusion is that only model 2B fulfilled the intended aim for this group of models, being more accurate than the remaining models at predicting the ethanol time courses.

Similarly, from the models of group 1, 1B was also better at predicting ethanol time courses than both models 1A and 1C. This is attributed to the fact that, in both models 1B and 2B, there is a direct reaction from glucose to biomass (reaction 3 in Table B-1 of Appendix B), which is not present in models 1A, 1C, 2A and 2C. Models 1C and 2C replace this reaction with a reaction from glucose to ethanol, and models 1A and 2A have no replacement reaction. It is thus safe to say that, for the experimental conditions to which this work refers, it is best to include a non-fermentative pathway (i.e., a pathway that does not entail ethanol formation) for biomass growth in a physiological model for D. hansenii.

It is also interesting to notice that biomass concentration is in general better fitted for the data in group II than in group I, yielding $R^2$ coefficients ranging from 0.88 to 0.91 in the former, and from 0.97 to 0.99 in the latter. Figure 6-4 below shows an example of the predicted time course for biomass concentration, by model 1B, on both data groups.

The poorer biomass modeling in data group I is related to the fact that the experimental time courses for biomass are significantly different between the two data groups (see Appendix A). The time courses belonging to group I show three different zones of biomass growth rate, corresponding to the consumption of the three different groups of substrates: 1) a briefly fast initial growth, attributed to glucose consumption, 2) a steadily slower and longer growth, owing mainly to xylose consumption, and 3) a final short growth burst, elicited by arabinose, xylitol and arabitol consumption.

Because the experimental points are more finely sampled in the beginning of the experiment (there are 4 measurements in the first 24 hours, while the sampling intervals are 24-hours long from then on), the models were able to accurately predict the first zone of biomass growth. However, they were not able to model as accurately the third zone, and predicted an approximately homogeneous growth
rate from the 24\textsuperscript{th} hour until the end of the experiment (see Figure 6-4). This problem did not arise for data group II because, in the time courses belonging to this group, the pattern described above is not present, and the rate of biomass growth is, indeed, approximately constant throughout the experiment.

6.3.3 Analysis of the Parameters of the Best Model

As discussed in Chapter 2, the selection of the best model takes into account to criteria:

1. The sum of a model's best OF values for the two data groups.
2. The number of parameters of the model.

The values of these quantities are shown, for each of the proposed models, in Table 6-8.

Table 6-8. Best OF value found for each model in the two data groups and their sum. The number of parameters of the models is also shown and the best model is highlighted.

<table>
<thead>
<tr>
<th>Model</th>
<th>No. of Parameters</th>
<th>Best OF Value in Data Group I</th>
<th>Best OF Value in Data Group II</th>
<th>Sum of Best OF Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>1A</td>
<td>26</td>
<td>1106.1</td>
<td>763.02</td>
<td>1869.12</td>
</tr>
<tr>
<td>1B</td>
<td>28</td>
<td>840.85</td>
<td>752.43</td>
<td>1593.28</td>
</tr>
<tr>
<td>1C</td>
<td>28</td>
<td>1006.7</td>
<td>521.67</td>
<td>1528.37</td>
</tr>
<tr>
<td>1D</td>
<td>27</td>
<td>911.49</td>
<td>645.56</td>
<td>1557.05</td>
</tr>
<tr>
<td>2A</td>
<td>28</td>
<td>1113.6</td>
<td>952.01</td>
<td>2065.61</td>
</tr>
<tr>
<td>2B</td>
<td>31</td>
<td>966.05</td>
<td>738.79</td>
<td>1704.84</td>
</tr>
<tr>
<td>2C</td>
<td>31</td>
<td>1119.9</td>
<td>537.13</td>
<td>1657.03</td>
</tr>
<tr>
<td>3A</td>
<td>31</td>
<td>997.34</td>
<td>765.69</td>
<td>1763.03</td>
</tr>
<tr>
<td>3B</td>
<td>28</td>
<td>922.28</td>
<td>643.80</td>
<td>1566.08</td>
</tr>
</tbody>
</table>

According to the values in Table 6-8, four models stand out from the rest as candidates for best model, due to their lower sum of best OF values (criterion 1): they are models 1C, 1D, 3B and 1B (in this order). Within these models, the difference between the lowest (model 1C) and highest (model 1B) sum of OF values is 64.91, while the difference between the sum of OF values of the 4\textsuperscript{th} best model (model 1B) and the 5\textsuperscript{th} best model (model 2C) is 63.75. Since the sum of best OF values is very similar among these four models, and model 1D has one less parameter than the other three models (criterion 2), the best overall model is model 1D. The values of the parameters of this model, for data groups I and II (model 1D\textsuperscript{I} and 1D\textsuperscript{II}), are shown in Figure 6-5. In order to aid in the comparison of these values, a semi-logarithmic plot (log\textsubscript{10}), showing the ratio of the values of the parameters in group II to their values in group I, is also displayed in Figure 6-6.

The analysis that follows is divided into two sections. Section 6.3.3.1 examines the differences in the parameters of each of the reactions of model 1D, as well as the consequences of these differences for the kinetics of each species, when the model refers to data group I and data group II. In order to get a more integrated view of the physiological consequences of these differences, section 6.3.3.2 briefly analyzes the predicted contributions of each of the reactions leading to biomass growth, in the two data groups.
Figure 6-5. Parameters obtained for model 1D in its best run, for data group I (a) and data group II (b).
6.3.3.1 Comparison of the Parameters in each Reaction

The following paragraphs discuss, in order, the main differences between each of the reactions in models 1D\textsuperscript{I} and 1D\textsuperscript{II}.

Reaction 1 is the reaction that is more similar between data groups, with its values for the maximum specific rate and yield coefficient very slightly decreasing in group II, and the value of \( K_{S,1} \) showing a 1.7-fold increase. For group II, the value of \( K_{S,1} \) is approximately the same as the initial (maximum) glucose concentration, and thus, at the initial instant, the reaction proceeds at a specific rate of \( \mu_{\text{max},1}/2 \). From then on, the specific rate of the reaction decreases, since glucose concentration is decreasing. For group I, on the other hand, the value of \( K_{S,1} \) is about half of the initial glucose concentration, and the reaction therefore attains a specific rate of \( \mu_{\text{max},1}/2 \) when glucose concentration decreases to this value, starting out faster at the initial instant. These observations are in accordance with the predicted time course for the concentration of glucose in both groups (see Appendix C), since for the first three experimental points, the decrease in glucose concentration is steeper in group I than in group II. This thus leads to the conclusion that, in terms of glucose consumption, the effect of the inhibitors present in data group II was achieved through the lowering of the affinity of the yeast for glucose. This conclusion is in accordance with the results of the work for which these experimental data were obtained [60].

For reaction 2, there was a marked increase of \( \mu_{\text{max},2} \) and decrease of \( K_{S,2} \). This resulted in an overall faster rate of glycerol consumption, and explains the lack of accumulation of glycerol predicted by this
model for data group II. As for the yield coefficients, they both showed a slight decrease, although the yield for the conversion of glycerol to ethanol ($Y_{2.1}$) remained higher than the yield of the conversion of glycerol to biomass ($Y_{2.2}$) on both data groups. This preference for the conversion of glycerol to ethanol is however more pronounced on group II than on group I ($Y_{2.1}$ is 1.9-fold higher than $Y_{2.2}$ on group II and 1.4-fold higher on group I).

In reaction 3, $\mu_{\text{max},3}$ increased by a factor of about 1.5 and $K_{S,3}$ decreased by a factor of about 0.005 from data group I to data group II. If the reaction were not inhibited, this would lead to an increase of its specific rate $\mu_3$. However, the value of the inhibition constant of this reaction, $K_{I,1}$, is, in group II, about 1/4 of its value in group I. Because the concentration of the inhibitor (glucose) is also higher for group II in the initial period of the experiment, the ratio $K_{I,1}/\text{Glc}$, whose inverse is indicative of the degree of inhibition, is smaller for group II and it may be concluded that the reaction is more inhibited in group II. In this case, two different situations must be considered:

i) until glucose is fully depleted (4th time instant), the specific rate of this reaction is therefore lower in group II than in group I (due to the greater inhibitory effect of glucose in group II); as the concentration of the inhibitor decreases, the reaction proceeds faster, and,

ii) from the moment when the concentration of glucose drops to zero until the end of the experiment, the specific rate of the reaction is higher on group II than on group I, since there is no inhibition and both $\mu_{\text{max},3}$ and $1/K_{S,3}$ are higher in group II.

As for each of the yield coefficients, they slightly increase from data group I to data group II. The ratio $Y_{3.1}/Y_{3.2}$ also increases, being approximately 6 in group I and 10 in group II, that is, the preference for the conversion of xylose into xylitol, as opposed to arabinol, becomes more pronounced in group II. This is in accordance with the higher maximum xylitol concentrations achieved in group II (see Appendix C).

In regard to reaction 4, the only parameter that is not significantly different in the two data groups is $\mu_{\text{max},4}$. $K_{S,4}$ is more than $10^{-4}$ times smaller in group II (see Figure 6-6), resulting in a faster specific rate of xylitol consumption in group II, as may be observed in the predicted time course for xylitol in Appendix C (xylitol decrease is steeper for group II than for group I). The yield coefficients also show great discrepancies between data groups. In group I, the yield for the conversion of xylitol to biomass is negligible ($Y_{4.2} = 0.0001$), while in group II, it is the yield of the conversion of xylitol to ethanol that is negligible ($Y_{4.1} = 0.0001$). For data group I, the preferred pathway for the production of biomass from xylitol is via ethanol production (with the yield of biomass production from ethanol also being higher in group I than in group II, and the corresponding reaction faster). For data group II, almost no xylitol is converted to ethanol, and instead biomass is directly produced. This effect is at least partially responsible for the higher final concentration of biomass obtained for group II.

In the case of reaction 5, both $\mu_{\text{max},5}$ and $K_{S,5}$ increased by approximately two orders of magnitude in data group II, while $Y_5$ increased 1.7-fold. The inhibition constant, $K_{I,2}$, decreased by more than $10^{-1}$ (see Figure 6-6). For group II, the value of $K_{S,5}$ is approximately that of the initial concentration of arabinose, and thus, if the reaction were not inhibited, it would start at an initial specific rate $\mu_5(t=0) = \frac{\mu_{\text{max},5}}{2}$, there from decreasing with decreasing substrate concentration. However, since it is inhibited, the initial $\mu$ of the reaction is lower, and gradually increases until zero concentration of the inhibitor
(xylose) is attained. On the contrary, for group I, the value of $K_{S,5}$ is much lower than the initial concentration of arabinose, and thus, if the reaction were not inhibited, it would start at an $\mu$ much higher than $\mu_{\text{max,5}}/2 = 0.75 \text{ g.L}^{-1}\cdot\text{h}^{-1}$, there from decreasing and reaching a value of $\mu_{\text{max,5}}/2 = 0.0114 \text{ g.L}^{-1}\cdot\text{h}^{-1}$ (much lower than that of data group II) at a concentration of arabinose equal to $K_{S,5} = 0.0471 \text{ g.L}^{-1}$. However, since the reaction is inhibited, its initial specific rate is lower, and gradually increases until zero concentration of xylose is attained (that is, until the inhibition factor reaches 1). The dominant effect of the higher $\mu_{\text{max,5}}$ obtained for group II on the specific rate $\mu_{S,5}$ is more easily acknowledged if one plots, on the same axes and for both data groups, the values taken over time by:

i) the rate of reaction 5, $|dAra/dt|$, given by

$$\frac{dAra}{dt} = \mu_{S}(t) \cdot X(t) = \frac{\mu_{\text{max,5}} A(t)}{A(t) + K_{S}} \frac{K_{I,2}}{K_{I,2} + Xyl(t)} X(t)$$  \hspace{1cm} (6.1)

ii) the concentration of arabinose, $Ara(t)$, and

iii) the inhibition factor, given by

$$\frac{K_{I,2}}{K_{I,2} + Xyl(t)}.$$

Figure 6-7 shows such a plot, where, for readability reasons, the comparison was made only between one replicate in each of the data groups (replicate B2E1 from group I and replicate B2E10 from group II). This means that, for each data group, the sets of initial concentrations given by the chosen replicates, were used to obtain, via numerical integration, the values of the concentrations, $Ara(t)$ and $Xyl(t)$, needed to compute the aforementioned values. Also, because the time courses of biomass concentration, $X(t)$, are similar on both data groups, they are not shown on Figure 6-7, as their value is not important to assess the differences between the groups, in terms of rate of arabinose consumption.

![Figure 6-7. Time courses of arabinose concentration, $Ara(t)$, of (scaled) arabinose rate of consumption, $dAra/dt$, and of the corresponding (scaled) inhibition factor, for replicate B2E1 in data group I (labeled GI) and B2E10 in data group II (labeled GII). For clarity reasons, the ordinate axis is not labeled and the units of the plotted values are given in the legend. The scaling factors are the same for both data groups.](image-url)
As may be seen in Figure 6-7, the rate of arabinose consumption (solid and dashed red lines marked with circles) increases more with increasing inhibition factor in group II than in group I (solid and dashed black lines marked with diamonds). The peak value of the arabinose rate of consumption is 1.5 times greater in group II than in group I, and occurs earlier in the former than in the latter, leading to an earlier and steeper decrease of arabinose concentration (dashed green line marked with squares).

In regard to reaction 6, describing arabitol consumption, parameters $\mu_{\text{max},6}$ and $K_{S,6}$ increased in group II by about one order of magnitude (see Figure 6-6), while the yield coefficients, $Y_{6.1}$ and $Y_{6.2}$, decreased by about two and three orders of magnitude, respectively. The pronounced increased observed for $\mu_{\text{max},6}$ and $K_{S,6}$ is of course related to the increase observed for the corresponding parameters of reaction 5, in which arabitol is produced. A consequence of the increase of $\mu_{\text{max},6}$ and $K_{S,6}$ is that the concentration of arabitol only *slightly* decreases at the end of the experiment in group I, while its decrease is steeper in group II.

For reaction 7, describing the conversion of ethanol to biomass, $\mu_{\text{max},7}$ remained similar in both data groups, while $K_{S,7}$ increased by approximately two orders of magnitude in group II. The yield coefficient decreased, in group II, to half of the value it took in group I. In this case too, it is more informative to observe a plot showing, for both data groups, the temporal profiles of: i) the specific rate of ethanol consumption, and ii) the concentration of ethanol. Such a plot is depicted in Figure 6-8. In group I (solid lines of Figure 6-8), the specific rate of ethanol consumption very quickly rises to its maximum value $\mu_{\text{max},7} = 0.0152 \, \text{h}^{-1}$ and remains at that value for as long as there is ethanol to consume. On the contrary, on group II, the initial rise of $\mu_{7}$ is less steep and its peak value of less than $\mu_{\text{max},7}/2 = 0.008 \, \text{h}^{-1}$ is attained at an ethanol concentration close to $K_{S,7} = 7.0726 \, \text{g.L}^{-1}$. The decrease in the ethanol specific consumption rate is also much less steeper in group II than in group I, since ethanol concentration also decreases more smoothly in the latter group.

![Time Courses of Ethanol Concentration and Ethanol Specific Rate of Consumption for Data Groups I and II](image)

*Figure 6-8. Time courses of ethanol concentration, $E_{\text{OH}}(t)$, and of (scaled) ethanol specific rate of consumption, $\mu$, for replicates B2E1 in data group I (labeled GI) and B2E10 in data group II (labeled GII).*
6.3.3.2 Comparison of the Contributions of the Reactions leading to Biomass Growth Rate

An aspect deserving further analysis is the effect of each of the terms contributing to the observed biomass growth rate over time. This effect may be studied by drawing a plot simultaneously featuring the specific rate of biomass growth, \( \frac{dX}{dt} \), given by

\[
\frac{dX}{dt} = \frac{1}{X} \mu(t) = y_{2,2} \mu_2(t) + y_{3,2} \mu_3(t) + y_{6,2} \mu_6(t) + y_{7} \mu_7(t)
\]  

(6.2)

and each of the terms in the equation (6.2), corresponding the specific rates of the reactions which originate biomass. This plot is shown in Figure 6-9, for the same replicates used in Figure 6-7 and Figure 6-8.

In group I, three different regions of behavior for the specific rate of biomass growth are observed:

i) an initial region, characterized by a very fast initial increase of \( \mu \), until a peak value of approximately 0.13 h\(^{-1}\) is reached at the 6\(^{th}\) hour, followed by an equally fast decrease;

ii) an intermediate region, characterized by a less pronounced decrease of \( \mu \) from the 12\(^{th}\) to the 18\(^{th}\) hour, followed by a period (until the 54\(^{th}\) hour) in which \( \mu \) remains approximately constant and equal to 0.013 h\(^{-1}\), and finally ending with an additional decrease of \( \mu \) to 0.004 h\(^{-1}\) in the 60\(^{th}\) hour;

iii) a final region in which, from the 60\(^{th}\) hour until the end of the experiment, \( \mu \) remains approximately constant and equal to 0.004 h\(^{-1}\).

These three regions are governed by the contributions of different reactions, as may be observed in Figure 6-9. In the first region, the main contributing reaction is reaction 2, in which glycerol is converted to biomass, with the profile of \( Y_{2,2,12} \) closely following the profile of \( \mu \). The conversion of ethanol to biomass is the main reaction responsible for region ii), with the initial increase of \( Y_{7,7,17} \) causing \( \mu \) to decrease less rapidly, its subsequent constant value resulting in \( \mu \) also not changing, and its final decrease leading to \( \mu \) also decreasing with similar slope. In the final region, the main contributing reaction is reaction 6 (the conversion of arabitol to biomass), and, to a lesser extent, also
reaction 7. $Y_{6.2l46}$ slightly increases during this period, while $Y_{7l17}$ slightly decreases, leading to an overall approximately constant value for $\mu$ in this region. It is important to notice that reaction 4 practically doesn't contribute to biomass growth, that is, no biomass is directly produced from xylitol (the maximum value of $Y_{4.2l44}$ is $8.99 \times 10^{-7} \text{ h}^{-1}$). As discussed in section 6.3.3.1, for data group I, practically all the available xylitol is converted to ethanol, which is subsequently converted to biomass, without any significant extracellular accumulation.

As for group II, the initial region of the temporal profile of $\mu$ is analogous to that found in group I, with the main reaction responsible for such a region being the conversion of glycerol to biomass (reaction 2). The differences are that: i) for data group II, reaction 6, referring to the conversion of xylitol to biomass, also plays a minor role in this initial region, and ii) the peak value of $\mu$ is, for data group II, about half of that observed in group I. The two remaining regions defined for group I are, in group II, merged into a single region, in which $\mu$ very slowly decreases until reaching a final value of 0.01 h$^{-1}$. In this region, the temporal profile of $\mu$ may be described as the sum of an approximately constant base rate, due to reaction 4 and given by $Y_{4.2l44}$, and a slowly decreasing rate, due to reaction 7 and given by $Y_{7l17}$. Thus, for this data group, it is arabitol that practically does not contribute to biomass growth, with the maximum value of $Y_{6.2l46}$ being $9.32 \times 10^{-7} \text{ h}^{-1}$).

In summary, and in terms of the species that contribute to biomass growth, the main differences between the models for the two data groups are:

i) In the absence of the inhibitors (group I), xylitol consumption is slower and it is mostly used to produce ethanol (since the yield coefficient for biomass production is very low). This causes arabitol to be used instead for biomass growth, and its conversion to ethanol to occur to a lesser extent.

ii) In the presence of the inhibitors (group II), more xylitol is consumed in the final period of the experiment, and its consumption is mainly directed to biomass growth (the yield coefficient for ethanol production is very low). *D. hansenii* also assimilates arabitol, but it is not capable of efficiently growing or producing ethanol from it (both yield coefficients are very low).

An analogous analysis could be done for the ethanol production rate, confirming what was just stated. These predictions are an illustrative example of the hypothesis that may be put forward by physiological models such as the ones developed in this work, regarding the direction and magnitude of the physiological fluxes (rates of consumption of substrates and formation of products) induced by different experimental settings. The physiological accuracy of such predictions shall of course depend on the simplifying assumptions made when constructing the model, and should be subject to experimental scrutiny (for example, through enzymatic assays, whose results in terms of enzyme activity may be qualitatively compared to the predicted direction of the fluxes).
7 Conclusions

The fitting of fermentation data to kinetic models of microbial growth is a complex problem for which the usual deterministic optimization algorithms, based on the derivatives of the objective function, yield poor results. This work tested two alternative parameter estimation strategies, based on population-based stochastic optimization algorithms, namely particle swarm optimization (PSO) and a genetic algorithm (GA). It was concluded that the PSO performed much better than the GA.

The tuning of the operating parameters of the PSO revealed that the best settings were those for which the algorithm presents a convergent behavior, with oscillation or zigzagging. Within those cases, the best performance was achieved when local search ability was preferred over global search ability, as determined by the values of the acceleration constants and of the inertia weight. The zone of convergence without oscillation or zigzagging yielded the worst performance, leading to the premature convergence of the algorithm. For the GA, the tuning revealed that it worked better with high crossover fractions and that single point crossover was more adequate than Wright's heuristic crossover. Additionally, it was found that too high a mutation range in the last iterations of the algorithm had very negative effects on its performance.

In terms of the physiological modeling, nine model topologies were tested for the two data groups available (groups I and II, referring respectively to growth in the absence and in the presence of metabolic inhibitors). The parameter estimation results, on which several samples of solutions found by the PSO were considered, showed that the best objective function values obtained by each model were statistically equivalent to the variability of the experimental data. The only exception was model 2A, which gave very poor results when optimized against data group II. The model for which the PSO consistently yielded the best objective function values was model 1D, and it was concluded that its more constrained type of kinetic modeling was the reason for the success of the algorithm in finding a good optimum.

The concentrations predicted by each model all closely agreed with the experimental ones ($R^2 \approx 1$), except for glycerol and, in some instances, for ethanol. This was attributed to specific constraints of the models and to how they relate to the initial growth rate of biomass. The fit achieved for data group II was, in general, better than the fit achieved for data group I, mainly due to the lowest experimental variability observed in the former.

Model 1D was found to be the best model, considering both the objective function values obtained for the two data groups and its reduced number of parameters. It was possible to provide an interpretation for the variation of its parameters from one data group to the other, and formulate a testable hypothesis for the phenomena that took place, in terms of the different physiological pathways leading to biomass growth. This hypothesis was that:

i) In the absence of metabolic inhibitors (data group I), xylitol consumption is slower and this polyol is mostly used to produce ethanol, causing arabitol to be used instead for biomass growth, and its conversion to ethanol to occur to a lesser extent.
ii) In the presence of inhibitors (data group II), more xylitol is consumed in the final period of the experiment, and its consumption is mainly directed towards biomass growth, instead of ethanol production; *D. hansenii* also assimilates arabitol, but it is not capable of efficiently growing or producing ethanol from it.

It was also concluded that, when dealing with models of this complexity, a more diverse set of initial concentrations (of biomass and the three primary substrates) should be experimentally tested, if one wishes the values of the parameters to be more reliable. On the other hand, this diversity should not be too large, since one is interested in capturing the physiological behavior of the system in a certain zone of operation, and not for all possible combinations of initial concentrations.

The importance of obtaining high level physiological models of microorganism behavior, as opposed to microscopic metabolic models (for which the fitting exercise requires measurements that are most often very expensive to obtain), is, thus, reiterated. Furthermore, having demonstrated better performance than the GA, the PSO algorithm was shown to be a viable and valuable optimization technique for estimating the parameters of these complex models.
References


Appendix A  Experimental Data

The following figures depict the experimental data used in this work: Figure A-1 refers to data belonging to group I, corresponding to standard experimental conditions and comprising three experiments, labeled B1E1, B2E1 and B2E2, and Figure A-2 to data belonging to group II, comprising four experiments, labeled B1E12, B2E9, B2E10 and B2E11, which were carried out in the presence of all the studied inhibitory substances (acetic acid, formic acid and hydroquinone). Each plot in the figures below refers to each of the eight species modeled in this work – X, Glc, GlyOH, EtOH, Xyl, XOH, Ara and ArOH –, plotting each of its n time courses on the n replicates belonging to each group; only the markers (and not the lines connecting them) correspond to experimental measurements. For the sake of consistency, the scale of both axes is the same on all plots but, because the concentrations of glycerol and ethanol are very low compared to those of the other species, two additional plots are presented for them where a more appropriate scale is used.
Figure A-1. Experimental time courses for each of the species in group I.
Figure A-2. Experimental time courses for each of the species in group II.
Appendix B  
Models and Parameters

\[
\begin{align*}
\frac{dX}{dt} &= \left[ Y_1 \cdot \mu_{\max,1} \cdot \frac{\text{EtOH}}{\text{EtOH} + K_{5,7}} - \frac{K_{1,4}}{K_{1,4} + \text{Glc}} \right] \cdot X \\
\frac{d\text{Glc}}{dt} &= -\mu_{\max,1} \cdot \frac{\text{Glc}}{\text{Glc} + K_{5,1}} \cdot X \\
\frac{d\text{GlyOH}}{dt} &= \left[ Y_1 \cdot \mu_{\max,1} \cdot \frac{\text{Glc}}{\text{Glc} + K_{5,1}} - \mu_{\max,2} \cdot \frac{\text{GlyOH}}{\text{GlyOH} + K_{5,2}} \right] \cdot X \\
\frac{d\text{EtOH}}{dt} &= \left[ Y_2 \cdot \mu_{\max,2} \cdot \frac{\text{GlyOH}}{\text{GlyOH} + K_{5,2}} + Y_5 \cdot \mu_{\max,5} \cdot \frac{\text{XOH}}{\text{XOH} + K_{5,4}} \cdot \frac{K_{1,2}}{K_{1,2} + X_{\text{yl}}} + \right. \\
&\quad \left. + Y_6 \cdot \mu_{\max,6} \cdot \frac{\text{ArOH}}{\text{ArOH} + K_{5,6}} - \mu_{\max,3} \cdot \frac{\text{EtOH}}{\text{EtOH} + K_{5,7}} \cdot \frac{K_{1,4}}{K_{1,4} + \text{Glc}} \right] \cdot X \\
\frac{d\text{Xyl}}{dt} &= -\mu_{\max,3} \cdot \frac{\text{Xyl}}{\text{Xyl} + K_{5,3}} \cdot \frac{K_{1,1}}{K_{1,1} + \text{Glc}} \cdot X \\
\frac{d\text{XOH}}{dt} &= \left[ Y_1 \cdot \mu_{\max,1} \cdot \frac{\text{Xyl}}{\text{Xyl} + K_{5,3}} \cdot \frac{K_{1,1}}{K_{1,1} + \text{Glc}} - \mu_{\max,4} \cdot \frac{\text{XOH}}{\text{XOH} + K_{5,4}} \cdot \frac{K_{1,2}}{K_{1,2} + X_{\text{yl}}} \right] \cdot X \\
\frac{d\text{Ara}}{dt} &= -\mu_{\max,5} \cdot \frac{\text{Ara}}{\text{Ara} + K_{5,5}} \cdot \frac{K_{1,3}}{K_{1,3} + \text{Xyl}} \cdot X \\
\frac{d\text{ArOH}}{dt} &= \left[ -\mu_{\max,6} \cdot \frac{\text{ArOH}}{\text{ArOH} + K_{5,7}} \cdot Y_3 \cdot \mu_{\max,8} \cdot \frac{\text{EtOH}}{\text{EtOH} + K_{5,8}} + \right. \\
&\quad \left. + Y_5 \cdot \mu_{\max,5} \cdot \frac{\text{XOH}}{\text{XOH} + K_{5,4}} \cdot \frac{K_{1,2}}{K_{1,2} + X_{\text{yl}}} + Y_6 \cdot \mu_{\max,6} \cdot \frac{\text{ArOH}}{\text{ArOH} + K_{5,6}} \cdot \frac{K_{1,3}}{K_{1,3} + \text{Xyl}} \right. \\
&\quad \left. + Y_8 \cdot \mu_{\max,3} \cdot \frac{\text{Xyl}}{\text{Xyl} + K_{5,3}} \cdot \frac{K_{1,1}}{K_{1,1} + \text{Glc}} \right] \cdot X
\end{align*}
\]

Figure B-1. System of ODEs of model 1A.

\[
\begin{align*}
\frac{dX}{dt} &= \left[ Y_3 \cdot \mu_{\max,3} \cdot \frac{\text{Glc}}{\text{Glc} + K_{5,3}} + Y_5 \cdot \mu_{\max,8} \cdot \frac{\text{EtOH}}{\text{EtOH} + K_{5,8}} \right] \cdot X \\
\frac{d\text{Glc}}{dt} &= -\mu_{\max,1} \cdot \frac{\text{Glc}}{\text{Glc} + K_{5,1}} \cdot \frac{\text{Glc}}{\text{Glc} + K_{5,3}} \cdot X \\
\frac{d\text{GlyOH}}{dt} &= \left[ Y_1 \cdot \mu_{\max,1} \cdot \frac{\text{Glc}}{\text{Glc} + K_{5,1}} - \mu_{\max,2} \cdot \frac{\text{GlyOH}}{\text{GlyOH} + K_{5,2}} \right] \cdot X \\
\frac{d\text{EtOH}}{dt} &= \left[ Y_2 \cdot \mu_{\max,2} \cdot \frac{\text{GlyOH}}{\text{GlyOH} + K_{5,2}} + Y_5 \cdot \mu_{\max,5} \cdot \frac{\text{XOH}}{\text{XOH} + K_{5,4}} \cdot \frac{K_{1,2}}{K_{1,2} + X_{\text{yl}}} + \right. \\
&\quad \left. + Y_6 \cdot \mu_{\max,6} \cdot \frac{\text{ArOH}}{\text{ArOH} + K_{5,6}} - \mu_{\max,3} \cdot \frac{\text{EtOH}}{\text{EtOH} + K_{5,7}} \cdot \frac{K_{1,4}}{K_{1,4} + \text{Glc}} \right] \cdot X \\
\frac{d\text{Xyl}}{dt} &= -\mu_{\max,4} \cdot \frac{\text{Xyl}}{\text{Xyl} + K_{5,4}} \cdot \frac{K_{1,1}}{K_{1,1} + \text{Glc}} \cdot X \\
\frac{d\text{XOH}}{dt} &= \left[ Y_1 \cdot \mu_{\max,1} \cdot \frac{\text{Xyl}}{\text{Xyl} + K_{5,3}} \cdot \frac{K_{1,1}}{K_{1,1} + \text{Glc}} - \mu_{\max,5} \cdot \frac{\text{XOH}}{\text{XOH} + K_{5,5}} \cdot \frac{K_{1,2}}{K_{1,2} + X_{\text{yl}}} \right] \cdot X \\
\frac{d\text{Ara}}{dt} &= -\mu_{\max,5} \cdot \frac{\text{Ara}}{\text{Ara} + K_{5,5}} \cdot \frac{K_{1,3}}{K_{1,3} + \text{Xyl}} \cdot X \\
\frac{d\text{ArOH}}{dt} &= \left[ -\mu_{\max,6} \cdot \frac{\text{ArOH}}{\text{ArOH} + K_{5,7}} \cdot Y_3 \cdot \mu_{\max,8} \cdot \frac{\text{EtOH}}{\text{EtOH} + K_{5,8}} + \right. \\
&\quad \left. + Y_5 \cdot \mu_{\max,5} \cdot \frac{\text{XOH}}{\text{XOH} + K_{5,4}} \cdot \frac{K_{1,2}}{K_{1,2} + X_{\text{yl}}} + Y_6 \cdot \mu_{\max,6} \cdot \frac{\text{ArOH}}{\text{ArOH} + K_{5,6}} \cdot \frac{K_{1,3}}{K_{1,3} + \text{Xyl}} + Y_8 \cdot \mu_{\max,3} \cdot \frac{\text{Xyl}}{\text{Xyl} + K_{5,3}} \cdot \frac{K_{1,1}}{K_{1,1} + \text{Glc}} \right] \cdot X
\end{align*}
\]

Figure B-2. System of ODEs of model 1B.
\[
\frac{dX}{dt} = Y_s \cdot \mu_{\text{max},8} \cdot \frac{\text{EtOH}}{\text{EtOH} + K_{s,8}} \cdot X
\]

\[
\frac{d\text{Glc}}{dt} = \left[ -\mu_{\text{max},1} \cdot \frac{\text{Glc}}{\text{Glc} + K_{s,1}} - \mu_{\text{max},3} \cdot \frac{\text{Glc}}{\text{Glc} + K_{s,3}} \right] \cdot X
\]

\[
\frac{d\text{GlyOH}}{dt} = \left[ Y_s \cdot \mu_{\text{max},1} \cdot \frac{\text{Glc}}{\text{Glc} + K_{s,1}} - \mu_{\text{max},2} \cdot \frac{\text{GlyOH}}{\text{GlyOH} + K_{s,2}} \right] \cdot X
\]

\[
\frac{d\text{EtOH}}{dt} = \left[ Y_s \cdot \mu_{\text{max},2} \cdot \frac{\text{GlyOH}}{\text{GlyOH} + K_{s,2}} + Y_s \cdot \mu_{\text{max},3} \cdot \frac{\text{Glc}}{\text{Glc} + K_{s,3}} + Y_s \cdot \mu_{\text{max},7} \cdot \frac{\text{ArOH}}{\text{ArOH} + K_{s,7}} + \right.
\]
\[+ Y_s \cdot \mu_{\text{max},5} \cdot \frac{\text{XOH}}{\text{XOH} + K_{s,5}} \cdot K_{t,2} + X_y \] \[ - \mu_{\text{max},8} \cdot \frac{\text{EtOH}}{\text{EtOH} + K_{s,8}} \right] \cdot X
\]

\[
\frac{dX_y}{dt} = -\mu_{\text{max},4} \cdot \frac{\text{Xyl}}{X_y + K_{s,4} + \text{Glc}} \cdot X
\]

\[
\frac{d\text{XOH}}{dt} = \left[ Y_s \cdot \mu_{\text{max},4} \cdot \frac{\text{Xyl}}{X_y + K_{s,4}} \cdot K_{t,1} + \text{Glc} - \mu_{\text{max},5} \cdot \frac{\text{XOH}}{\text{XOH} + K_{s,5}} \cdot K_{t,2} + X_y \right] \cdot X
\]

\[
\frac{d\text{Ara}}{dt} = -\mu_{\text{max},6} \cdot \frac{\text{ArOH}}{\text{ArOH} + K_{s,6}} \cdot K_{t,1} + \text{Xyl} \cdot X
\]

\[
\frac{d\text{ArOH}}{dt} = \left[ -\mu_{\text{max},7} \cdot \frac{\text{ArOH}}{\text{ArOH} + K_{s,7}} + Y_s \cdot \mu_{\text{max},6} \cdot \frac{\text{Xyl}}{X_y + K_{s,4}} \cdot K_{t,1} + \text{Xyl} + Y_s \cdot \mu_{\text{max},4} \cdot \frac{\text{Xyl}}{X_y + K_{s,4}} \cdot K_{t,1} + \text{Glc} \right] \cdot X
\]

Figure B-3. System of ODEs of model 1C.

\[
\frac{dX}{dt} = Y_s \cdot \mu_{\text{max},8} \cdot \frac{\text{EtOH}}{\text{EtOH} + K_{s,8}} \cdot X
\]

\[
\frac{d\text{Glc}}{dt} = \left[ -\mu_{\text{max},1} \cdot \frac{\text{Glc}}{\text{Glc} + K_{s,1}} - \mu_{\text{max},3} \cdot \frac{\text{Glc}}{\text{Glc} + K_{s,3}} \right] \cdot X
\]

\[
\frac{d\text{GlyOH}}{dt} = \left[ Y_s \cdot \mu_{\text{max},1} \cdot \frac{\text{Glc}}{\text{Glc} + K_{s,1}} - \mu_{\text{max},2} \cdot \frac{\text{GlyOH}}{\text{GlyOH} + K_{s,2}} \right] \cdot X
\]

\[
\frac{d\text{EtOH}}{dt} = \left[ -\mu_{\text{max},7} \cdot \frac{\text{EtOH}}{\text{EtOH} + K_{s,7}} + Y_s \cdot \mu_{\text{max},2} \cdot \frac{\text{GlyOH}}{\text{GlyOH} + K_{s,2}} + Y_s \cdot \mu_{\text{max},4} \cdot \frac{\text{XOH}}{\text{XOH} + K_{s,5}} + Y_s \cdot \mu_{\text{max},6} \cdot \frac{\text{ArOH}}{\text{ArOH} + K_{s,6}} \right] \cdot X
\]

\[
\frac{dX_y}{dt} = -\mu_{\text{max},3} \cdot \frac{\text{Xyl}}{X_y + K_{s,4} + \text{Glc}} \cdot X
\]

\[
\frac{d\text{XOH}}{dt} = \left[ Y_s \cdot \mu_{\text{max},3} \cdot \frac{\text{Xyl}}{X_y + K_{s,4}} \cdot K_{t,1} + \text{Glc} - \mu_{\text{max},4} \cdot \frac{\text{XOH}}{\text{XOH} + K_{s,5}} \right] \cdot X
\]

\[
\frac{d\text{Ara}}{dt} = -\mu_{\text{max},5} \cdot \frac{\text{ArOH}}{\text{ArOH} + K_{s,5}} \cdot K_{t,1} + \text{Xyl} \cdot X
\]

\[
\frac{d\text{ArOH}}{dt} = \left[ -\mu_{\text{max},6} \cdot \frac{\text{ArOH}}{\text{ArOH} + K_{s,6}} + Y_s \cdot \mu_{\text{max},3} \cdot \frac{\text{Xyl}}{X_y + K_{s,4}} \cdot K_{t,1} + \text{Glc} + Y_s \cdot \mu_{\text{max},5} \cdot \frac{\text{ArOH}}{\text{ArOH} + K_{s,6}} \right] \cdot X
\]

Figure B-4. System of ODEs of model 1D.
Figure B-5. System of ODEs of model 2A.

\[
\begin{align*}
\frac{dX}{dt} &= \left[ Y_1 \cdot \mu_{\text{max},3} \cdot \frac{\text{EtOH}_H}{\text{EtOH}_H + K_{S,7}} + Y_9 \cdot \mu_{\text{max},8} \cdot \frac{\text{EtOH}_P}{\text{EtOH}_P + K_{S,8}} \right] \cdot X \\
\frac{d\text{Glc}}{dt} &= -\mu_{\text{max},1} \cdot \frac{\text{Glc}}{\text{Glc} + K_{S,1}} \cdot X \\
\frac{d\text{GlyOH}}{dt} &= \left[ Y_1 \cdot \mu_{\text{max},3} \cdot \frac{\text{Glc}}{\text{Glc} + K_{S,1}} - Y_6 \cdot \mu_{\text{max},2} \cdot \frac{\text{GlyOH}}{\text{GlyOH} + K_{S,2}} \right] \cdot X \\
\frac{d\text{EtOH}_P}{dt} &= \left[ -\mu_{\text{max},3} \cdot \frac{\text{EtOH}_P}{\text{EtOH}_P + K_{S,8}} + Y_6 \cdot \mu_{\text{max},6} \cdot \frac{\text{ArOH}}{\text{ArOH} + K_{S,5}} + Y_9 \cdot \mu_{\text{max},8} \cdot \frac{\text{XOH}}{\text{XOH} + K_{S,6} + K_{I,2} + \text{Xyl}} \right] \cdot X \\
\frac{d\text{Xyl}}{dt} &= -\mu_{\text{max},3} \cdot \frac{\text{Xyl}}{\text{Xyl} + K_{S,3} + K_{I,1} + \text{Glc}} \cdot X \\
\frac{d\text{GlyOH}}{dt} &= \left[ Y_1 \cdot \mu_{\text{max},3} \cdot \frac{\text{Xyl}}{\text{Xyl} + K_{S,3}} \cdot \frac{K_{I,1}}{K_{I,1} + \text{Glc}} - \mu_{\text{max},4} \cdot \frac{\text{XOH}}{\text{XOH} + K_{S,4} + K_{I,2} + \text{Xyl}} \right] \cdot X \\
\frac{d\text{ArOH}}{dt} &= \left[ -\mu_{\text{max},6} \cdot \frac{\text{ArOH}}{\text{ArOH} + K_{S,5}} + Y_5 \cdot \mu_{\text{max},3} \cdot \frac{\text{Ar}}{\text{Ar} + K_{S,5} + K_{I,2} + \text{Xyl}} + Y_9 \cdot \mu_{\text{max},8} \cdot \frac{\text{Xyl}}{\text{Xyl} + K_{S,3} + K_{I,1} + \text{Glc}} \right] \cdot X
\end{align*}
\]

Figure B-6. System of ODEs of model 2B.
\[
\begin{align*}
\frac{dx}{dt} &= \left[ Y_6 \cdot \mu_{\text{max},3} \cdot \frac{\text{EtOH}_H}{\text{EtOH}_H + K_{5,8}} + Y_9 \cdot \mu_{\text{max},9} \cdot \frac{\text{EtOH}_P}{\text{EtOH}_P + K_{5,9}} \right] \cdot x \\
\frac{d\text{Glc}}{dt} &= \left[ -\mu_{\text{max},3} \cdot \frac{\text{Glc}}{\text{Glc} + K_{5,1}} - \mu_{\text{max},3} \cdot \frac{\text{Glc}}{\text{Glc} + K_{5,3}} \right] \cdot x \\
\frac{d\text{GlyOH}}{dt} &= \left[ Y_1 \cdot \mu_{\text{max},1} \cdot \frac{\text{Glc}}{\text{Glc} + K_{5,1}} + \mu_{\text{max},2} \cdot \frac{\text{Glc}}{\text{Glc} + K_{5,2}} - \mu_{\text{max},3} \cdot \frac{\text{GlyOH}}{\text{GlyOH} + K_{5,2}} \right] \cdot x \\
\frac{d\text{EtOH}_H}{dt} &= \left[ Y_2 \cdot \mu_{\text{max},2} \cdot \frac{\text{GlyOH}}{\text{GlyOH} + K_{5,2}} + Y_5 \cdot \mu_{\text{max},2} \cdot \frac{\text{Glc}}{\text{Glc} + K_{5,3}} - \mu_{\text{max},8} \cdot \frac{\text{EtOH}_H}{\text{EtOH}_H + K_{5,8}} \right] \cdot x \\
\frac{d\text{EtOH}_P}{dt} &= \left[ -\mu_{\text{max},8} \cdot \frac{\text{EtOH}_P}{\text{EtOH}_P + K_{5,9}} + Y_7 \cdot \mu_{\text{max},7} \cdot \frac{\text{ArOH}}{\text{ArOH} + K_{5,7}} + Y_9 \cdot \mu_{\text{max},9} \cdot \frac{\text{XOH}}{\text{XOH} + K_{5,5}} \cdot \frac{K_{1,2}}{K_{1,2} + x} \right] \cdot x \\
\frac{dxyl}{dt} &= \left[ -\mu_{\text{max},4} \cdot \frac{xyl}{xyl + K_{5,4}} \cdot \frac{K_{1,1}}{K_{1,1} + \text{Glc}} \right] \cdot x \\
\frac{dxoh}{dt} &= \left[ Y_6 \cdot \mu_{\text{max},6} \cdot \frac{xyl}{xyl + K_{5,6}} \cdot \frac{K_{1,1}}{K_{1,1} + \text{Glc}} - \mu_{\text{max},5} \cdot \frac{\text{XOH}}{\text{XOH} + K_{5,5}} \cdot \frac{K_{1,2}}{K_{1,2} + x} \right] \cdot x \\
\frac{dara}{dt} &= \left[ -\mu_{\text{max},6} \cdot \frac{ara}{ara + K_{5,6}} \cdot \frac{K_{1,1}}{K_{1,1} + \text{Xyl}} \right] \cdot x \\
\frac{daroh}{dt} &= \left[ -\mu_{\text{max},7} \cdot \frac{\text{ArOH}}{\text{ArOH} + K_{5,7}} + Y_6 \cdot \mu_{\text{max},6} \cdot \frac{\text{Xyl}}{\text{Xyl} + K_{5,4}} \cdot \frac{K_{1,1}}{K_{1,1} + \text{Glc}} + Y_9 \cdot \mu_{\text{max},9} \cdot \frac{\text{Xyl}}{\text{Xyl} + K_{5,4}} \cdot \frac{K_{1,1}}{K_{1,1} + \text{Glc}} \right] \cdot x \\
\end{align*}
\]

Figure B-7. System of ODEs of model 2C.

\[
\begin{align*}
\frac{dx}{dt} &= \left[ Y_3 \cdot \mu_{\text{max},3} \cdot \frac{\text{Glc}}{\text{Glc} + K_{5,3}} + Y_4 \cdot \mu_{\text{max},4} \cdot \frac{\text{GlyOH}}{\text{GlyOH} + K_{5,4}} + Y_5 \cdot \mu_{\text{max},3} \cdot \frac{\text{EtOH}}{\text{EtOH} + K_{5,5}} + Y_7 \cdot \mu_{\text{max},7} \cdot \frac{\text{XOH}}{\text{XOH} + K_{5,5}} \cdot \frac{K_{1,2}}{K_{1,2} + x} \right] \cdot x \\
\frac{d\text{Glc}}{dt} &= \left[ -\mu_{\text{max},1} \cdot \frac{\text{Glc}}{\text{Glc} + K_{5,1}} - \mu_{\text{max},2} \cdot \frac{\text{Glc}}{\text{Glc} + K_{5,2}} - \mu_{\text{max},3} \cdot \frac{\text{Glc}}{\text{Glc} + K_{5,3}} \right] \cdot x \\
\frac{d\text{GlyOH}}{dt} &= \left[ Y_1 \cdot \mu_{\text{max},1} \cdot \frac{\text{Glc}}{\text{Glc} + K_{5,1}} - \mu_{\text{max},4} \cdot \frac{\text{GlyOH}}{\text{GlyOH} + K_{5,4}} \right] \cdot x \\
\frac{d\text{EtOH}}{dt} &= \left[ Y_2 \cdot \mu_{\text{max},2} \cdot \frac{\text{GlyOH}}{\text{GlyOH} + K_{5,2}} - \mu_{\text{max},3} \cdot \frac{\text{EtOH}}{\text{EtOH} + K_{5,5}} \right] \cdot x \\
\frac{dxyl}{dt} &= \left[ -\mu_{\text{max},6} \cdot \frac{xyl}{xyl + K_{5,6}} \cdot \frac{K_{1,1}}{K_{1,1} + \text{Glc}} \right] \cdot x \\
\frac{dxoh}{dt} &= \left[ Y_6 \cdot \mu_{\text{max},6} \cdot \frac{xyl}{xyl + K_{5,6}} \cdot \frac{K_{1,1}}{K_{1,1} + \text{Glc}} - \mu_{\text{max},7} \cdot \frac{\text{XOH}}{\text{XOH} + K_{5,5}} \cdot \frac{K_{1,2}}{K_{1,2} + x} \right] \cdot x \\
\frac{dara}{dt} &= \left[ -\mu_{\text{max},6} \cdot \frac{ara}{ara + K_{5,6}} \cdot \frac{K_{1,1}}{K_{1,1} + \text{Xyl}} \right] \cdot x \\
\frac{daroh}{dt} &= \left[ -\mu_{\text{max},7} \cdot \frac{\text{ArOH}}{\text{ArOH} + K_{5,7}} + Y_6 \cdot \mu_{\text{max},6} \cdot \frac{\text{Xyl}}{\text{Xyl} + K_{5,4}} \cdot \frac{K_{1,1}}{K_{1,1} + \text{Glc}} + Y_9 \cdot \mu_{\text{max},9} \cdot \frac{\text{Xyl}}{\text{Xyl} + K_{5,4}} \cdot \frac{K_{1,1}}{K_{1,1} + \text{Glc}} \right] \cdot x \\
\end{align*}
\]

Figure B-8. System of ODEs of model 3A.
### Table B-1. Reactions’ numbers for each model (model 1A to 2A).

<table>
<thead>
<tr>
<th>Reaction Number</th>
<th>Reaction in each Model</th>
<th>Reaction</th>
</tr>
</thead>
<tbody>
<tr>
<td>1A</td>
<td>1B</td>
<td>1C</td>
</tr>
<tr>
<td>1</td>
<td>Glc → GlYOH</td>
<td>Glc → GlYOH</td>
</tr>
<tr>
<td>2</td>
<td>GlyOH → EtOH</td>
<td>GlyOH → EtOH</td>
</tr>
<tr>
<td>3</td>
<td>Xyl → XOH</td>
<td>Glc → X</td>
</tr>
<tr>
<td>4</td>
<td>XOH → EtOH</td>
<td>Xyl → XOH</td>
</tr>
<tr>
<td>5</td>
<td>Ara → ArOH</td>
<td>XOH → EtOH</td>
</tr>
<tr>
<td>6</td>
<td>ArOH → EtOH</td>
<td>Ara → ArOH</td>
</tr>
<tr>
<td>7</td>
<td>EtOH → X</td>
<td>ArOH → EtOH</td>
</tr>
<tr>
<td>8</td>
<td>Xyl → ArOH</td>
<td>EtOH → X</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>ArOH → EtOH</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Inhibition Number

<table>
<thead>
<tr>
<th>Reaction Number</th>
<th>Inhibition Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Glc on 3</td>
</tr>
<tr>
<td>2</td>
<td>Glc on 4</td>
</tr>
<tr>
<td>3</td>
<td>Glc on 4</td>
</tr>
<tr>
<td>4</td>
<td>Glc on 4</td>
</tr>
<tr>
<td>5</td>
<td>Glc on 4</td>
</tr>
<tr>
<td>6</td>
<td>Glc on 4</td>
</tr>
<tr>
<td>7</td>
<td>Glc on 4</td>
</tr>
<tr>
<td>8</td>
<td>Glc on 4</td>
</tr>
</tbody>
</table>

### Table B-1 (continued). Numbers for each model (model 2B to 3B).

<table>
<thead>
<tr>
<th>Reaction Number</th>
<th>Reaction in each Model</th>
<th>Reaction in each Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>2B</td>
<td>2C</td>
<td>3A</td>
</tr>
<tr>
<td>1</td>
<td>Glc → GlYOH</td>
<td>Glc → GlYOH</td>
</tr>
<tr>
<td>2</td>
<td>GlyOH → EtOHH</td>
<td>GlyOH → EtOHH</td>
</tr>
<tr>
<td>3</td>
<td>Glc → X</td>
<td>Glc → EtOHH</td>
</tr>
<tr>
<td>4</td>
<td>Xyl → XOH</td>
<td>Xyl → XOH</td>
</tr>
<tr>
<td>5</td>
<td>XOH → EtOHp</td>
<td>XOH → EtOHp</td>
</tr>
<tr>
<td>6</td>
<td>Ara → ArOH</td>
<td>Ara → ArOH</td>
</tr>
<tr>
<td>7</td>
<td>ArOH → EtOHp</td>
<td>ArOH → EtOHp</td>
</tr>
<tr>
<td>8</td>
<td>EtOHH → X</td>
<td>EtOHH → X</td>
</tr>
<tr>
<td>9</td>
<td>EtOHp → X</td>
<td>EtOHp → X</td>
</tr>
<tr>
<td>10</td>
<td>Xyl → ArOH</td>
<td>Xyl → ArOH</td>
</tr>
</tbody>
</table>

### Inhibition Number

<table>
<thead>
<tr>
<th>Reaction Number</th>
<th>Inhibition Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Glc on 4</td>
</tr>
<tr>
<td>2</td>
<td>Xyl on 5</td>
</tr>
<tr>
<td>3</td>
<td>Xyl on 6</td>
</tr>
</tbody>
</table>

---

80
Table B-2. Parameters’ numbers for each model.

<table>
<thead>
<tr>
<th>Parameter Number</th>
<th>1A</th>
<th>1B</th>
<th>1C</th>
<th>1D</th>
<th>2A</th>
<th>2B</th>
<th>2C</th>
<th>3A</th>
<th>3B</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$\mu_{\text{max},1}$</td>
<td>$\mu_{\text{max},1}$</td>
<td>$\mu_{\text{max},1}$</td>
<td>$\mu_{\text{max},1}$</td>
<td>$\mu_{\text{max},1}$</td>
<td>$\mu_{\text{max},1}$</td>
<td>$\mu_{\text{max},1}$</td>
<td>$\mu_{\text{max},1}$</td>
<td>$\mu_{\text{max},1}$</td>
</tr>
<tr>
<td>2</td>
<td>$\mu_{\text{max},2}$</td>
<td>$\mu_{\text{max},2}$</td>
<td>$\mu_{\text{max},2}$</td>
<td>$\mu_{\text{max},2}$</td>
<td>$\mu_{\text{max},2}$</td>
<td>$\mu_{\text{max},2}$</td>
<td>$\mu_{\text{max},2}$</td>
<td>$\mu_{\text{max},2}$</td>
<td>$\mu_{\text{max},2}$</td>
</tr>
<tr>
<td>3</td>
<td>$\mu_{\text{max},3}$</td>
<td>$\mu_{\text{max},3}$</td>
<td>$\mu_{\text{max},3}$</td>
<td>$\mu_{\text{max},3}$</td>
<td>$\mu_{\text{max},3}$</td>
<td>$\mu_{\text{max},3}$</td>
<td>$\mu_{\text{max},3}$</td>
<td>$\mu_{\text{max},3}$</td>
<td>$\mu_{\text{max},3}$</td>
</tr>
<tr>
<td>4</td>
<td>$\mu_{\text{max},4}$</td>
<td>$\mu_{\text{max},4}$</td>
<td>$\mu_{\text{max},4}$</td>
<td>$\mu_{\text{max},4}$</td>
<td>$\mu_{\text{max},4}$</td>
<td>$\mu_{\text{max},4}$</td>
<td>$\mu_{\text{max},4}$</td>
<td>$\mu_{\text{max},4}$</td>
<td>$\mu_{\text{max},4}$</td>
</tr>
<tr>
<td>5</td>
<td>$\mu_{\text{max},5}$</td>
<td>$\mu_{\text{max},5}$</td>
<td>$\mu_{\text{max},5}$</td>
<td>$\mu_{\text{max},5}$</td>
<td>$\mu_{\text{max},5}$</td>
<td>$\mu_{\text{max},5}$</td>
<td>$\mu_{\text{max},5}$</td>
<td>$\mu_{\text{max},5}$</td>
<td>$\mu_{\text{max},5}$</td>
</tr>
<tr>
<td>6</td>
<td>$\mu_{\text{max},6}$</td>
<td>$\mu_{\text{max},6}$</td>
<td>$\mu_{\text{max},6}$</td>
<td>$\mu_{\text{max},6}$</td>
<td>$\mu_{\text{max},6}$</td>
<td>$\mu_{\text{max},6}$</td>
<td>$\mu_{\text{max},6}$</td>
<td>$\mu_{\text{max},6}$</td>
<td>$\mu_{\text{max},6}$</td>
</tr>
<tr>
<td>7</td>
<td>$\mu_{\text{max},7}$</td>
<td>$\mu_{\text{max},7}$</td>
<td>$\mu_{\text{max},7}$</td>
<td>$\mu_{\text{max},7}$</td>
<td>$\mu_{\text{max},7}$</td>
<td>$\mu_{\text{max},7}$</td>
<td>$\mu_{\text{max},7}$</td>
<td>$\mu_{\text{max},7}$</td>
<td>$\mu_{\text{max},7}$</td>
</tr>
<tr>
<td>8</td>
<td>$Y_1$</td>
<td>$\mu_{\text{max},8}$</td>
<td>$\mu_{\text{max},8}$</td>
<td>$Y_1$</td>
<td>$\mu_{\text{max},8}$</td>
<td>$\mu_{\text{max},8}$</td>
<td>$\mu_{\text{max},8}$</td>
<td>$\mu_{\text{max},8}$</td>
<td>$\mu_{\text{max},8}$</td>
</tr>
<tr>
<td>9</td>
<td>$Y_2$</td>
<td>$Y_1$</td>
<td>$Y_2$</td>
<td>$Y_1$</td>
<td>$Y_1$</td>
<td>$Y_1$</td>
<td>$Y_1$</td>
<td>$Y_1$</td>
<td>$Y_1$</td>
</tr>
<tr>
<td>10</td>
<td>$Y_3$</td>
<td>$Y_2$</td>
<td>$Y_3$</td>
<td>$Y_2$</td>
<td>$Y_2$</td>
<td>$Y_2$</td>
<td>$Y_2$</td>
<td>$Y_2$</td>
<td>$Y_2$</td>
</tr>
<tr>
<td>11</td>
<td>$Y_4$</td>
<td>$Y_3$</td>
<td>$Y_4$</td>
<td>$Y_3$</td>
<td>$Y_3$</td>
<td>$Y_3$</td>
<td>$Y_3$</td>
<td>$Y_3$</td>
<td>$Y_3$</td>
</tr>
<tr>
<td>12</td>
<td>$Y_5$</td>
<td>$Y_4$</td>
<td>$Y_5$</td>
<td>$Y_4$</td>
<td>$Y_4$</td>
<td>$Y_4$</td>
<td>$Y_4$</td>
<td>$Y_4$</td>
<td>$Y_4$</td>
</tr>
<tr>
<td>13</td>
<td>$Y_6$</td>
<td>$Y_5$</td>
<td>$Y_6$</td>
<td>$Y_5$</td>
<td>$Y_5$</td>
<td>$Y_5$</td>
<td>$Y_5$</td>
<td>$Y_5$</td>
<td>$Y_5$</td>
</tr>
<tr>
<td>14</td>
<td>$Y_7$</td>
<td>$Y_6$</td>
<td>$Y_7$</td>
<td>$Y_6$</td>
<td>$Y_6$</td>
<td>$Y_6$</td>
<td>$Y_6$</td>
<td>$Y_6$</td>
<td>$Y_6$</td>
</tr>
<tr>
<td>15</td>
<td>$Y_8$</td>
<td>$Y_7$</td>
<td>$Y_8$</td>
<td>$Y_7$</td>
<td>$Y_7$</td>
<td>$Y_7$</td>
<td>$Y_7$</td>
<td>$Y_7$</td>
<td>$Y_7$</td>
</tr>
<tr>
<td>16</td>
<td>$K_{S,1}$</td>
<td>$Y_8$</td>
<td>$Y_8$</td>
<td>$Y_8$</td>
<td>$Y_8$</td>
<td>$Y_8$</td>
<td>$Y_8$</td>
<td>$Y_8$</td>
<td>$Y_8$</td>
</tr>
<tr>
<td>17</td>
<td>$K_{S,2}$</td>
<td>$Y_9$</td>
<td>$Y_9$</td>
<td>$Y_{10}$</td>
<td>$Y_9$</td>
<td>$Y_8$</td>
<td>$Y_8$</td>
<td>$Y_8$</td>
<td>$Y_8$</td>
</tr>
<tr>
<td>18</td>
<td>$K_{S,3}$</td>
<td>$K_{S,1}$</td>
<td>$K_{S,1}$</td>
<td>$K_{S,1}$</td>
<td>$K_{S,1}$</td>
<td>$K_{S,1}$</td>
<td>$K_{S,1}$</td>
<td>$K_{S,1}$</td>
<td>$K_{S,1}$</td>
</tr>
<tr>
<td>19</td>
<td>$K_{S,4}$</td>
<td>$K_{S,2}$</td>
<td>$K_{S,2}$</td>
<td>$K_{S,2}$</td>
<td>$Y_{10}$</td>
<td>$Y_{10}$</td>
<td>$Y_{10}$</td>
<td>$K_{S,2}$</td>
<td>$K_{S,2}$</td>
</tr>
<tr>
<td>20</td>
<td>$K_{S,5}$</td>
<td>$K_{S,3}$</td>
<td>$K_{S,3}$</td>
<td>$K_{S,3}$</td>
<td>$K_{S,3}$</td>
<td>$K_{S,3}$</td>
<td>$K_{S,3}$</td>
<td>$K_{S,3}$</td>
<td>$K_{S,3}$</td>
</tr>
<tr>
<td>21</td>
<td>$K_{S,6}$</td>
<td>$K_{S,4}$</td>
<td>$K_{S,4}$</td>
<td>$K_{S,4}$</td>
<td>$K_{S,4}$</td>
<td>$K_{S,4}$</td>
<td>$K_{S,4}$</td>
<td>$K_{S,4}$</td>
<td>$K_{S,4}$</td>
</tr>
<tr>
<td>22</td>
<td>$K_{S,7}$</td>
<td>$K_{S,5}$</td>
<td>$K_{S,5}$</td>
<td>$K_{S,5}$</td>
<td>$K_{S,5}$</td>
<td>$K_{S,5}$</td>
<td>$K_{S,5}$</td>
<td>$K_{S,5}$</td>
<td>$K_{S,5}$</td>
</tr>
<tr>
<td>23</td>
<td>$K_{I,1}$</td>
<td>$K_{S,6}$</td>
<td>$K_{S,6}$</td>
<td>$K_{S,6}$</td>
<td>$K_{S,6}$</td>
<td>$K_{S,6}$</td>
<td>$K_{S,6}$</td>
<td>$K_{S,6}$</td>
<td>$K_{S,6}$</td>
</tr>
<tr>
<td>24</td>
<td>$K_{I,2}$</td>
<td>$K_{S,7}$</td>
<td>$K_{S,7}$</td>
<td>$K_{S,7}$</td>
<td>$K_{S,7}$</td>
<td>$K_{S,7}$</td>
<td>$K_{S,7}$</td>
<td>$K_{S,7}$</td>
<td>$K_{S,7}$</td>
</tr>
<tr>
<td>25</td>
<td>$K_{I,3}$</td>
<td>$K_{S,8}$</td>
<td>$K_{S,8}$</td>
<td>$K_{S,8}$</td>
<td>$K_{S,8}$</td>
<td>$K_{S,8}$</td>
<td>$K_{S,8}$</td>
<td>$K_{S,8}$</td>
<td>$K_{S,8}$</td>
</tr>
<tr>
<td>26</td>
<td>$K_{I,4}$</td>
<td>$K_{I,1}$</td>
<td>$K_{I,1}$</td>
<td>$K_{I,1}$</td>
<td>$K_{I,1}$</td>
<td>$K_{I,1}$</td>
<td>$K_{I,1}$</td>
<td>$K_{I,1}$</td>
<td>$K_{I,1}$</td>
</tr>
<tr>
<td>27</td>
<td>$-$</td>
<td>$K_{I,2}$</td>
<td>$K_{I,2}$</td>
<td>$K_{I,2}$</td>
<td>$K_{I,2}$</td>
<td>$K_{S,8}$</td>
<td>$K_{S,8}$</td>
<td>$K_{S,8}$</td>
<td>$K_{I,2}$</td>
</tr>
<tr>
<td>28</td>
<td>$-$</td>
<td>$K_{I,3}$</td>
<td>$K_{I,3}$</td>
<td>$-$</td>
<td>$K_{I,3}$</td>
<td>$K_{S,9}$</td>
<td>$K_{S,9}$</td>
<td>$K_{S,9}$</td>
<td>$K_{I,3}$</td>
</tr>
<tr>
<td>29</td>
<td>$-$</td>
<td>$-$</td>
<td>$-$</td>
<td>$-$</td>
<td>$-$</td>
<td>$K_{I,1}$</td>
<td>$K_{I,1}$</td>
<td>$K_{I,1}$</td>
<td>$-$</td>
</tr>
<tr>
<td>30</td>
<td>$-$</td>
<td>$-$</td>
<td>$-$</td>
<td>$-$</td>
<td>$-$</td>
<td>$K_{I,2}$</td>
<td>$K_{I,2}$</td>
<td>$K_{I,2}$</td>
<td>$-$</td>
</tr>
<tr>
<td>31</td>
<td>$-$</td>
<td>$-$</td>
<td>$-$</td>
<td>$-$</td>
<td>$-$</td>
<td>$K_{I,3}$</td>
<td>$K_{I,3}$</td>
<td>$K_{I,3}$</td>
<td>$-$</td>
</tr>
</tbody>
</table>
Appendix C  Predicted Time Courses of Concentrations

The following figures show the time courses for each species' concentrations predicted by each model, with the parameters found in its best run, for each group of data. In each plot, each different line/marker corresponds to a prediction in which the integration procedure departed from the initial conditions given by each of the replicates available in a data group (3 available sets of initial conditions for group I and 4 for group II). The percentage of total error depicted in each plot refers to the percentage of the OF value that the species is responsible for.
Figure C-1. Predicted time courses for the concentrations of each species made by model 1A using the parameters found in its best run for data group I (a) and data group II (b).
Figure C-2. Predicted time courses for the concentrations of each species made by model 1B using the parameters found in its best run for data group I (a) and data group II (b).
Figure C-3. Predicted time courses for the concentrations of each species made by model 1C using the parameters found in its best run for data group I (a) and data group II (b).
Figure C-4. Predicted time courses for the concentrations of each species made by model 1D using the parameters found in its best run for data group I (a) and data group II (b).
Model 2A

Time Courses for Biomass in Group I

Time Courses for Glucose in Group I

Time Courses for Glycerol in Group I

Time Courses for Ethanol in Group I

Time Courses for Xylose in Group I

Time Courses for Xylool in Group I

Time Courses for Xylose in Group I

Time Courses for Xylool in Group I

(a)
Figure C-5. Predicted time courses for the concentrations of each species made by model 2A using the parameters found in its best runs for data group I (a) and data group II (b).
Figure C-6. Predicted time courses for the concentrations of each species made by model 2B using the parameters found in its best run for data group I (a) and data group II (b).
Model 2C

Time Courses for Biomass in Group I

Time Courses for Glucose in Group I

Time Courses for Oligonucleotide in Group I

Time Courses for Ethanol in Group I

Time Courses for Xylose in Group I

Time Courses for Xyitol in Group I

Time Courses for Arabinose in Group I

Time Courses for Arabinol in Group I

Percentage of Total Error %
Figure C-7. Predicted time courses for the concentrations of each species made by model 2C using the parameters found in its best run for data group I (a) and data group II (b).
Figure C-8. Predicted time courses for the concentrations of each species made by model 3A using the parameters found in its best run for data group I (a) and data group II (b).
Model 3B

(a)
Figure C-9. Predicted time courses for the concentrations of each species made by model 3B using the parameters found in its best run for data group I (a) and data group II (b).