Computers and Chemical Engineering 34 (2010) 1719–1730

Contents lists available at ScienceDirect

Computers and Chemical Engineering

journal homepage: www.elsevier.com/locate/compchemeng

Outer approximation-based algorithm for biotechnology studies in systems biology

Carlos Pozo[a], Gonzalo Guillén-Gosálbez[a,*], Albert Sorribas[b], Laureano Jiménez[a]

[a] Departament d'Enginyeria Quimica, Universitat Rovira i Virgili, Avinguda Paisos Catalans 26, 43007 Tarragona, Spain
[b] Departament de Ciències Mèdiques Bàsiques, Institut de Recerca Biomèdica de Lleida (IRBLLEIDA), Universitat de Lleida, Montserrat Roig 2, 25008 Lleida, Spain

ARTICLE INFO

Article history:
Received 31 July 2009
Received in revised form 15 February 2010
Accepted 1 March 2010
Available online 7 March 2010

Keywords:
Global optimization
Generalized Mass Action (GMA)
Metabolic engineering

ABSTRACT

Optimization methods play a central role in systems biology studies as they can help in identifying key processes that can be experimentally changed so that specific biological goals can be attained. Standard optimization methods used in this field rely on simplified linear models that may fail in capturing the underlying complexity of the target metabolic network. Within this general context, we present a novel approach to globally optimize metabolic networks. The approach presented relies on (1) adopting a general modeling framework for metabolic networks: the Generalized Mass Action (GMA) representation; (2) posing the optimization task as a non-convex nonlinear programming (NLP) problem; and (3) devising an efficient solution method for globally optimizing the resulting NLP that embeds a GMA model of the metabolic network. The capabilities of our method are illustrated through two case studies: the anaerobic fermentation pathway in *Saccharomyces cerevisiae* and the citric acid production using *Aspergillus niger*. Numerical results show that the method presented provides near optimal solutions in low CPU times even in cases where the commercial global optimization package BARON fails to close the optimality gap.

1. Introduction

The study of complex biological systems requires the integration of experimental and computational research by adopting a systems biology approach. Systems biology addresses the study of the interactions between the individual components of a biological system through the integration of data and mathematical models. Here, computational biology plays a major role by developing mathematical tools that aim to provide a powerful foundation from which to address critical scientific questions. In particular, the optimization of metabolic networks has emerged as a very important goal in biotechnology (Bailey, Birnbaum, Galazzo, Khosla, & Shanks, 1990; Banga, 2008; Cameron & Chaplen, 1997; Cameron & Tong, 1993; Mendes & Kell, 1996; Torres & Voit, 2002).

In recent years, the use of genetic manipulation techniques has led to significant improvements in the production of certain biochemical products. However, in most cases mutation and selection of new processes have been made in a trial-and-error basis, which has led to local optimal solutions. Hence, one expects that actual biological processes could be further improved if quantitative design principles for the modification of the genes were provided by a more rational approach like optimization (Banga, 2008; Chang & Sahinidis, 2005; Hatzimanikatis, Floudas, & Bailey, 1996; Polisetty, Gatzke, & Voit, 2008; Vera, de Atauri, Cascante, & Torres, 2003; Voit, 1992). This optimization is known as *metabolic engineering* (Bailey et al., 1990; Bailey, 1991, 1999) and consists of, given a model, finding the appropriate changes in the enzyme activities that optimize (maximize) a certain objective function (typically, the synthesis rate of the desired product). The enzyme activities obtained in the optimization solution can be implemented in the real system by tuning the expressions of the corresponding genes.

The use of mathematical optimization to improve biotechnological processes is nowadays gaining wider acceptance given their potential to produce significant economical savings. These may be achieved by reducing the number of experiments required to find those microorganisms that lead to higher yields. Furthermore, as manipulation of many enzymes at once may be prohibitive, a theoretical analysis on the more promising alternative combinations of limited changes is of great practical interest. Additionally, the solutions of the optimization procedure can provide valuable insights on the behavior of the biological systems, making these techniques useful in other applications such as evolution studies (Guillén-Gosálbez & Sorribas, 2009).

One of the key steps in this approach is the selection of the appropriate mathematical model among the different representations available. Here, we can distinguish between three main groups of models. The first group corresponds to stoichiometric models. These models constitute simple linear representations of

[*] Corresponding author.
E-mail address: Gonzalo.Guillen@urv.cat (G. Guillén-Gosálbez).

0098-1354/$ – see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compchemeng.2010.03.001

the stoichiometry of the network (i.e. network structure). However, their simplicity becomes at the same time their main limitation as they fail to capture the non-linear behavior of some key processes of the networks such as regulation (Gavalas, 1968; Heinrich & Schuster, 1996). On the other extreme of accuracy, we would find *ad hoc* models. These models rely on the formulation of detailed kinetics equations, such as Michaelis–Menten, that allow accounting for modulating effects. Unfortunately, optimizing these systems is not a straightforward task as it usually leads to complex mathematical formulations (Polisetty et al., 2008). A third group of models includes representations that result from the combination of linear stoichiometric descriptions and non-linear approximate representations to express the velocities of the metabolic reactions (Alves, Vilaprinyo, Hernàndez-Bermejo, & Sorribas, 2009; Sorribas, Hernndez-Bermejo, Vilaprinyo, & Alves, 2007). Among them, models using the so called power-law formalism show a good compromise between accuracy and simplicity (Marin-Sanguino, Voit, Gonzalez-Alzon, & Torres, 2007). This group includes the S-System and the General Mass Action (GMA) models, which seem a promising alternative in the area (Voit, 1992, 2003). The main advantage of these models is that they can capture the non-linearities required to describe the regulatory processes of the networks while showing linear properties in the logarithmic space. Additionally, these models constitute a very general framework since any kind of metabolic network can be represented through their formulations (Alves et al., 2009).

GMA models only differ from S-System models in the way in which the branching points are handled (Curto, Sorribas, & Cascante, 1995). In S-System models, all the input flows in the branching point are collected and modeled together as if they were a single flow. The same procedure is followed for the outputs so that, finally, the concentration of the metabolite being balanced is the result of just two contributions. On the other hand, in GMA models each process is approximated separately so that there are as many contributions as actual flows in the real system (Voit, 2000 and references therein). If the metabolic network only contains nodes that result from the contribution of an input flow and an output flow, the S-System and GMA representation coincide.

Models based on the power-law formalism were first used in metabolic optimization problems by Voit (1992). The choice of an S-Systems representation allowed him to obtain a linear representation by a simple logarithmic transformation performed on some variables of the model (Alvarez-Vasquez, Canovas, Iborra, & Torres, 2002; Marin-Sanguino & Torres, 2003; Marin-Sanguino et al., 2007; Vecchietti, Sangbum, & Grossmann, 2003). However, this is not possible in GMA models, since some equations cannot be directly reformulated using the logarithmic transformation. The optimization task then gives rise to a non-convex NLP that may show multiple local optima in which standard commercial packages can get trapped during the search.

In the context of performing a systems biology study, global optimality is particularly important, as one aims to draw general conclusions from the specific properties of the solution found. Hence, local solutions should be avoided, since they might hamper the entire biological analysis by providing insights that are not meaningful at all. A literature review in the area of global optimization of metabolic networks (Banga, 2008) reveals that this is indeed a ripe field for research. In a recent and pioneering work Polisetty et al. (2008) addressed the global optimization of GMA models (see also Marin-Sanguino et al., 2007; Marin-Sanguino & Torres, 2003). The main drawback of the strategy presented by Polisetty et al. (2008) is that it provides solutions with large optimality gaps (i.e., large differences between the best solution that could be found and the one calculated during the execution of the algorithm). More recently Guillén-Gosálbez and Sorribas (2009) presented a novel algorithm that makes use of global optimization techniques for performing feasibility analysis in evolution studies. The tool developed by these authors allowed characterizing the feasible space of optimization problems with embedded GMA models (Sorribas et al., 2010).

The aim of this work is to provide a systematic modeling framework and solution strategy for metabolic optimization problems arising in systems biology studies. The approach presented relies on posing the optimization task as a NLP with an embedded GMA model of the metabolic network under study. An outer-approximation algorithm is presented to solve this type of models to global optimality. We provide a theoretical analysis on some details of the algorithm and illustrate its capabilities through two examples, comparing our results with those produced by BARON, nowadays regarded as the “state of the art” global optimization package.

2. Problem statement

Given a metabolic network described by a GMA model, the optimization aims to determine the appropriate changes in enzyme activities and in the internal metabolite concentrations so that the synthesis rate of the desired product is maximized in steady state. Given data for the problem are: (1) the stoichiometry of the reactions involved in the production/consumption of each internal metabolite in the metabolic network; and (2) the value of the parameters of the power-law formalism representing the kinetics of each of these particular reactions at the basal state.

3. Mathematical formulation

3.1. GMA representation

The GMA representation of a metabolic network containing n internal metabolites whose concentration X_i can vary with the time t due to the action of p flows can be expressed as follows:

$$\frac{dX_i}{dt} = \sum_{r=1}^{p} \mu_{ir} v_r \quad i = 1, \ldots, n \tag{1}$$

where μ_{ir} is the stoichiometric coefficient of the metabolite i in the process r and indicates the number of molecules of metabolite i involved in such a process. Hence, it is always an integer value that is positive when process r contributes to the production of metabolite i, negative when process r consumes metabolite i and 0 otherwise (i.e., if process r does not participate in the production/consumption of metabolite i). The velocity v_r can be described using different representations, but, as stated previously, the so-called power-law formalism (Savageau, 1969a,b; Voit, 2000) is an appropriate one:

$$v_r = \gamma_r \prod_{j=1}^{n+m} X_j^{f_{rj}} \quad r = 1, \ldots, p \tag{2}$$

In this representation, γ_r is an apparent rate constant for flow r. f_{rj} is the kinetic order of metabolite j in process r and quantifies its effect on the considered rate. Note that contributions of the m (independent) external metabolites are also accounted for in this representation.

By introducing Eq. (2) into Eq. (1) and assuming steady state conditions for the network, one obtains a GMA model as follows:

$$\frac{dX_i}{dt} = \sum_{r=1}^{p} \left(\mu_{ir} \gamma_r \prod_{j=1}^{n+m} X_j^{f_{rj}} \right) = 0 \quad i = 1, \ldots, n \tag{3}$$

3.2. NLP formulation

In order to compute the changes in the enzyme activities, we shall rewrite the apparent rate constant γ_r in Eq. (3) as a product of the basal state enzyme activity γ_r (constant parameter) and its fold-change K_r (continuous variable):

$$\sum_{r=1}^{p}\left(\mu_{ir}K_r\gamma_r\prod_{j=1}^{n+m}X_j^{f_{rj}}\right)=0 \quad i=1,\ldots,n \tag{4}$$

The final goal of the optimization task is to find the appropriate changes to be performed in the enzyme activities in order to optimize a given biological criteria (typically a flow) described through algebraic equations. This requires the determination of the optimal values of K_r, v_r and X_j that maximize/minimize the given objective function while fulfilling the GMA model equations in steady state. In general, it will be possible to express the desired criterion in mathematical terms using a specific mathematical function $U(K_r, v_r, X_j)$, so that the optimization task can be posed as a non-linear programming problem (NLP) of the following form:

$$\begin{aligned}(\textbf{ONLP})\quad \min\quad & U(K_r, v_r, X_j)\\ s.t.\quad & \sum_{r=1}^{p}\mu_{ir}v_r=0 \qquad i=1,\ldots,n\\ & v_r=K_r\gamma_r\prod_{j=1}^{n+m}X_j^{f_{rj}} \quad r=1,\ldots,p\\ & K_r, v_r, X_j\in\mathbb{R}_+\end{aligned}$$

Note that maximization problems can be easily reformulated into minimization ones by changing the sign of the objective function. The nonlinear equality constraints that define the velocity terms in **ONLP** give rise to a non-convex search space. Hence, to solve **ONLP** to global optimality, it is necessary to resort to global optimization techniques (see Grossmann & Bigler, 2004; Floudas & Gounaris, 2009) that can provide solutions to the problem with a desired optimality tolerance. These methods can handle a wide variety of non-convex formulations arising in many types of applications. Unfortunately, in practice, their numerical performance may vary drastically depending on the specific problem being solved, leading in some cases to prohibitive CPU times (Grossmann & Bigler, 2004). A possible way to overcome this limitation consists of devising customized algorithms that exploit the mathematical properties of the specific problem under study. This is indeed the underlying idea of our approach.

4. Solution strategy

The method we propose to globally optimize **ONLP** is an outer-approximation algorithm based on the works of Bergamini, Aguirre, and Grossmann (2005) and Polisetty et al. (2008). Our method relies on decomposing the original problem **ONLP** into two problems at different hierarchical levels: an upper level master problem **CMILP** and a lower level slave problem **RNLP**. The master level entails the solution of a mixed-integer linear programming (MILP) problem that is a relaxation of **ONLP**. This implies that **CMILP** will predict valid lower bounds on the solution of **ONLP** (the solution of the relaxation will be, at least, as good as that of the original problem). In the lower level, the original problem is locally optimized in a reduced search space (**RNLP**) providing a valid upper bound on its global optimum. These two problems are solved iteratively until the optimality gap is reduced below a given tolerance. A detailed description of the algorithm is given in the following sections.

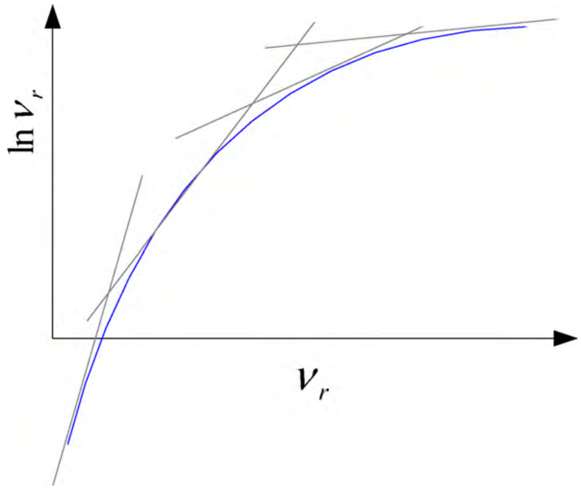

Fig. 1. Natural logarithm overestimation by a 1st degree Taylor series.

4.1. Upper level master problem

To construct a valid relaxation of **ONLP** (i.e., **CMILP**), we first reformulate the equations arising from the power-law formalism via a logarithmic transformation:

$$\ln v_r=\ln K_r+\ln\gamma_r+\sum_{j=1}^{n+m}f_{rj}\ln X_j \quad r=1,\ldots,p \tag{5}$$

We then introduce two new auxiliary variables, k_r and x_j, which are defined as follows:

$$k_r=\ln K_r$$
$$x_j=\ln X_j$$

By replacing the original variables in Eq. (5) by the reformulated ones, the following equality can be obtained.

$$\ln v_r=k_r+\ln\gamma_r+\sum_{j=1}^{n+m}f_{rj}x_j \quad r=1,\ldots,p \tag{6}$$

Eq. (6) can then be expressed via the following inequalities:

$$\ln v_r\geq k_r+\ln\gamma_r+\sum_{j=1}^{n+m}f_{rj}x_j \quad r=1,\ldots,p \tag{7}$$

$$\ln v_r\leq k_r+\ln\gamma_r+\sum_{j=1}^{n+m}f_{rj}x_j \quad r=1,\ldots,p \tag{8}$$

The logarithmic terms appearing in the left-hand side of these equations can be replaced by valid upper and lower estimators (note that γ_r is a known model parameter). Specifically, in Eq. (7), the logarithmic function can be approximated by L supporting hyper-planes (see Fig. 1), which are first order Taylor expansions of the natural logarithm at different points l of the domain of v_r:

$$\ln v_r^l+\frac{1}{v_r^l}(v_r-v_r^l)\geq k_r+\ln\gamma_r+\sum_{j=1}^{n+m}f_{rj}x_j \quad r=1,\ldots,p$$
$$l=1,\ldots,L \tag{9}$$

Since the logarithmic function is concave, these hyper-planes constitute valid overestimators that do not chop off any feasible solution of **ONLP**.

Furthermore, the left-hand side of Eq. (8) can be underestimated by a piecewise linear approximation. For that, we consider a partition of the original domain $[\underline{v_r}, \overline{v_r}]$ defined by a set of grid

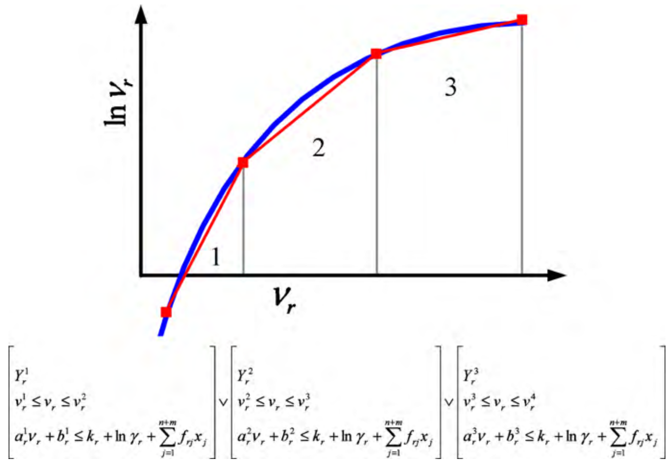

Fig. 2. Example of natural logarithm underestimation by piecewise linear functions.

points $v_r^1, v_r^2, \ldots, v_r^{H+1}$, being $v_r^1 = \underline{v_r}, v_r^{H+1} = \overline{v_r}$ and $v_r^{h+1} \ge v_r^h$ for $h = 1, \ldots, H$. The piecewise linear approximation can then be modeled via a disjunction with H terms as follows:

$$\bigvee_{h=1,\ldots,H} \begin{bmatrix} Y_r^h \\ v_r^h \le v_r \le v_r^{h+1} \\ a_r^h v_r + b_r^h \le k_r + \ln \gamma_r + \sum_{j=1}^{n+m} f_{rj} x_j \end{bmatrix} \quad r = 1, \ldots, p$$

$$Y_r^h \in \{True, False\} \quad r = 1, \ldots p \quad h = 1, \ldots, H$$

where a_r^h and b_r^h are the coefficients of the straight line equation in the h^{th} interval and Y_r^h indicates whether the h^{th} term in the disjunction of the r^{th} velocity is active or not. Fig. 2 shows and illustrative example of a piecewise function with three terms.

The disjunction can be reformulated using either the big-M or convex hull reformulations (see Vecchietti et al., 2003). The latter technique allows translating the disjunction into a set of equalities and inequalities using auxiliary (disaggregated) variables as follows:

$$\sum_{h=1}^{H} z_r^h = v_r \quad r = 1, \ldots, p \tag{10}$$

$$v_r^h y_r^h \le z_r^h \le v_r^{h+1} y_r^h \quad r = 1, \ldots, p \quad h = 1, \ldots, H \tag{11}$$

$$\sum_{h=1}^{H} y_r^h = 1 \quad r = 1, \ldots, p \tag{12}$$

$$\sum_{h=1}^{H} \left(a_r^h z_r^h + b_r^h y_r^h \right) \le k_r + \ln \gamma_r + \sum_{j=1}^{n+m} f_{rj} x_j \qquad r = 1, \ldots, p \tag{13}$$

where z_r^h is the new disaggregated variable and y_r^h is a new binary variable that takes a value of 1 if the h^{th} interval of the r^{th} velocity is active and 0 otherwise. Thus, the overall master problem can be finally expressed as follows:

$$\begin{array}{lll} (\textbf{CMILP}) & \min & U(k_r, x_j, v_r, z_r^h, y_r^h) \\ & s.t. & \text{constraints } 1, 9, 10 \text{ to } 13 \\ & & k_r, x_j \in \mathbb{R} \\ & & v_r, z_r^h \in \mathbb{R}_+ \\ & & y_r^h \in \{0, 1\} \end{array}$$

Model **CMILP** takes the form of a mixed-integer linear programming (MILP) problem. These problems can be solved efficiently via standard branch & bound (B&B) techniques.

4.2. Lower level slave problem

The slave problem in the lower level of the algorithm, **RNLP**, is obtained by tightening the search space of **ONLP**. This is accomplished by adding lower and upper bounds on the velocity terms v_r. The associated mathematical formulation is as follows:

$$\begin{array}{lll} (\textbf{RNLP}) & \min & U(K_r, v_r, X_j) \\ & s.t. & \sum_{r=1}^{p} \mu_{ir} v_r = 0 \quad i = 1, \ldots, n \\ & & v_r = K_r \gamma_r \prod_{j=1}^{n+m} X_j^{f_{rj}} \quad r = 1, \ldots, p \\ & & \underline{v_r} \le v_r \le \overline{v_r} \quad r = 1, \ldots, p \\ & & K_r, v_r, X_j \in \mathbb{R}_+ \end{array}$$

Hence, the search space of **RNLP** is tighter than that of **ONLP**. For this reason, **RNLP** provides an upper bound on the solution of **ONLP**. Note that in model **RNLP**, bounds on v_r (third group of constraints) can be obtained from the active intervals of the disjunctions of **CMILP**. For instance, let v_r^* be the solution of the master problem. We know that v_r^* must fall within the active interval of the term of the disjunction defined by $[v_r^h, v_r^{h+1}]$. Hence, we can set $\underline{v_r} = v_r^h$ and $\overline{v_r} = v_r^{h+1}$.

4.3. Algorithm steps

The detailed algorithmic steps of the proposed strategy are as follows:

(1) Set iteration count $it = 0$, $UB = \infty$, $LB = -\infty$ and tolerance error $= tol$.
(2) Set $it = it + 1$. Solve master problem **CMILP**.
   (a) If **CMILP** is infeasible, stop. **ONLP** is infeasible.
   (b) Otherwise, update the current LB as $LB = \max_{it}(LB_{it})$, where LB_{it} is the value of the objective function of **CMILP** in the it^{th} iteration. Set bounds on v_r for the slave problem according to the solution of the master problem ($\underline{v_r} = v_r^h$ and $\overline{v_r} = v_r^{h+1}$).
(3) Solve the slave problem **RNLP**.
   (a) If **RNLP** is infeasible update the grid (see remark 5) and go to step 2 of the algorithm.
   (b) Otherwise, update the current UB as $UB = \min_{it}(UB_{it})$, where UB_{it} is the value of the objective function of **RNLP** in the it^{th} iteration.
(4) Calculate the optimality gap OG as $OG = (|UB - LB|)/UB$.
   (a) If $OG \le tol$, then stop. The current UB can be regarded as the global optimal solution of **ONLP** within the predefined tolerance.
   (b) Otherwise, update the grid and go to step 2 of the algorithm.

4.4. Remarks

- The reformulation of Eq. (6) into two inequalities is only required for those velocities that are involved in balances at branching points, that is, where Eq. (3) includes more than two terms. In equations with only two terms, the logarithmic transformation is enough to obtain a linear constraint (note that the stoichiometric coefficients μ_{ir} are known). Hence, in mathematical terms, we have:

$$\mu_{ir} v_r = -\mu_{ir'} v_{r'} \quad i \in XT \quad r, r' \in VT_i \tag{14}$$

$$\ln \mu_{ir} + \ln v_r = \ln(-\mu_{ir'}) + \ln v_{r'} \quad i \in XT \quad r, r' \in VT_i \tag{15}$$

$$\ln \mu_{ir} + k_r + \ln \gamma_r + \sum_{j=1}^{n+m} f_{rj} x_j$$
$$= \ln(-\mu_{ir'}) + k_{r'} + \ln \gamma_{r'} + \sum_{j=1}^{n+m} f_{r'j} x_j \quad i \in XT \quad r, r' \in VT_i \tag{16}$$

where XT is the set of equations involving only two terms and VT_i is the set of velocities that appear in those equations in XT. Note that in S-System models, all the balances include only two terms. This allows reformulating the model into a linear equivalent form, which greatly helps computations (Voit, 1992). Another major advantage of the logarithmic transformation is that it gives rise to concave univariate terms (i.e., logarithmic functions) for which tight under and over estimators can be defined.

- Supporting hyper-planes can be located following different patterns. It can be shown that the one that minimizes the rectilinear distance (i.e., L_1 norm) between the hyper-planes and the actual logarithmic function is that in which this distance is the same at every interjection of two adjacent hyper-planes (see proof in Appendix A). This allocation can be obtained by solving an optimization problem.
- Similarly, the grid points of the piecewise approximation can be selected according to different criteria. One possible strategy consists of splitting the range $[\underline{v_r}, \overline{v_r}]$ into H intervals with the same width. It can be shown that in order to minimize the rectilinear distance (i.e., L_1 norm) between the piecewise approximation and the logarithmic function, one needs to define intervals of equal width in the logarithmic space (see proof in Appendix A). Hence, we would have:

$$\ln v_r^{h+1} - \ln v_r^h = \ln v_r^{h+2} - \ln v_r^{h+1} = \ldots = \ln v_r^{H+1} - \ln v_r^H$$
$$r = 1, \ldots, p \quad h = 1, \ldots, H \tag{17}$$

- Increasing the number of terms of the piecewise function and supporting hyper-planes leads to tighter bounds and hence to less iterations. Unfortunately, this is accomplished at the expense of adding more variables to the original problem. This is specially critic in the case of the piecewise approximation, which requires the definition of binary variables that increase considerably the computational burden of the master problem and consequently the time required by each iteration. Hence, a compromise should be found between the number of iterations and the time spent in each of them.
- There are different ways to update the piecewise grid of **CMILP** (steps 3a and 4b in the algorithm). One possible strategy is the division of the active interval into 2 sub-intervals with the same width either in the Cartesian space, $(v_r^h + v_r^{h+1})/2$, (see Fig. 3) or in the logarithmic space, $(\ln v_r^h + \ln v_r^{h+1})/2$. Another possibility is to split the active interval by adding the point corresponding to the solution of **RNLP** in the last iteration.
- Additional supporting hyper-planes can be iteratively added to **CMILP** in order to improve the overestimation of the logarithmic function. Again, the points where the new supporting hyper-planes will be allocated can be selected following different criteria.

5. Case studies

As benchmark problems to test the capabilities of the approach presented, we propose to use the ethanol production in the fermentation of *Saccharomyces cerevisiae* (case study 1) and the citric acid production by *Aspergillus niger* (case study 2) (see Figs. 4 and 5[1]).

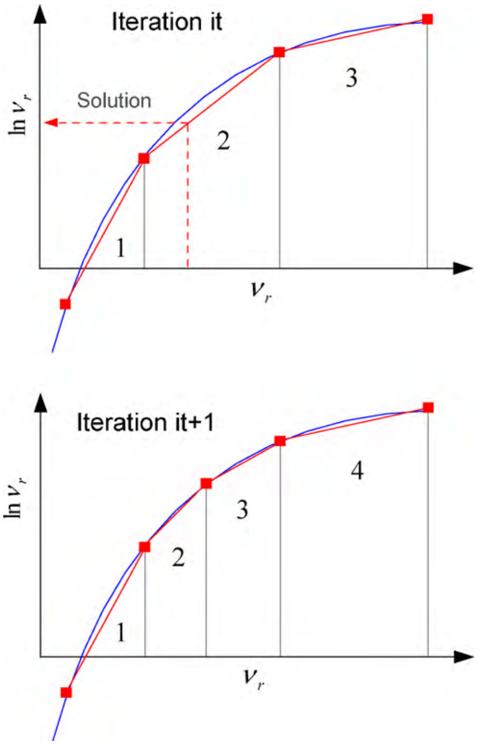

Fig. 3. Piecewise grid update example. As the solution of the first iteration is found in the second interval, it is split into two sub-intervals for the next iteration.

These two problems are convenient since their optimal solutions have already been published in the literature (Polisetty et al., 2008), the GMA models for the two systems can also be found in the same reference).

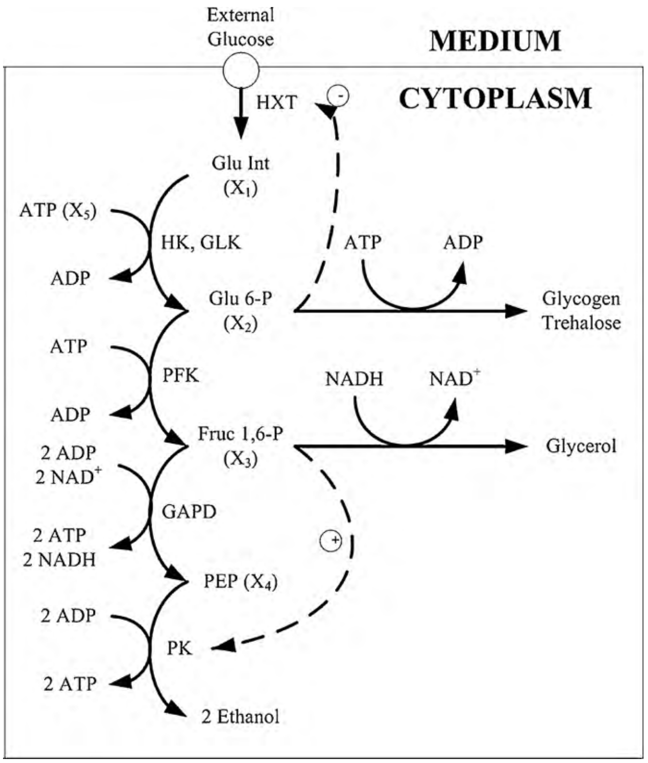

Fig. 4. Metabolic pathway of the fermentation of *Saccharomyces cerevisiae*.

[1] Figures adapted from the original work by Polisetty et al. (2008).

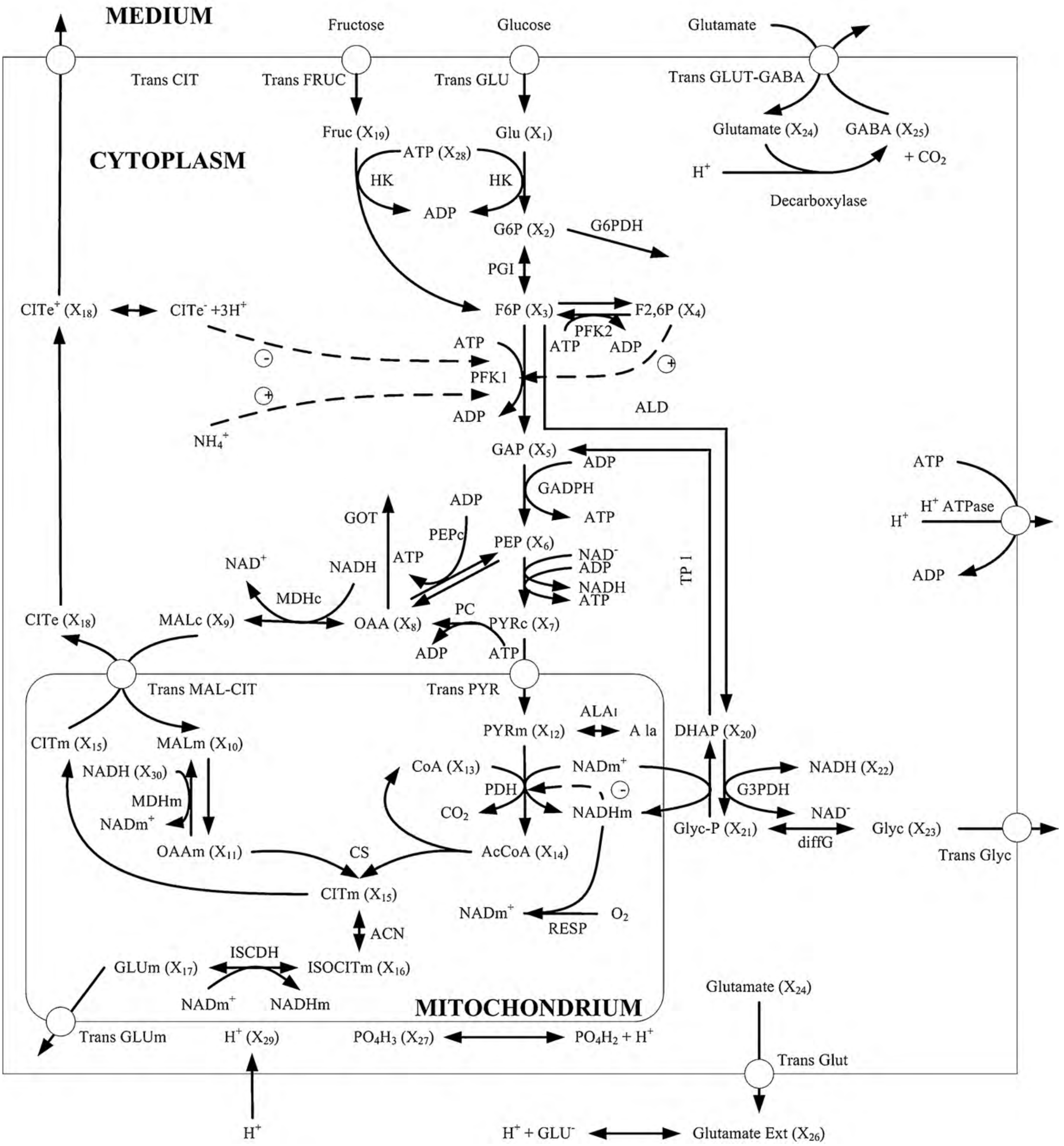

Fig. 5. Metabolic network for the citric acid production in *Aspergillus niger*.

The algorithm proposed in Section 4.3 was implemented in GAMS, using CPLEX (version 11.2.0) to solve the master MILPs and CONOPT (version 3.14s) to locally optimize the slave NLPs on an Intel 1.2 GHz machine. Data about the size of the models can be found in Table 1.

Note that henceforth the optimization problems we deal with are maximizations. Thus, the convexified master problem **CMILP** will determine upper bounds to the solution of **ONLP** whereas the lower bounds will be identified by the nonlinear slave problem **RNLP**.

Table 1
Numerical data of the size of the models.

	Ethanol production (*Saccharomyces cerevisiae*)	Citric acid production (*Aspergillus niger*)
CIMLP equations	518	4235
Continuous variables	50	439
Integer variables	53	471
RNLP equations	40	211
Variables	14	91

Table 2
Results of the global optimization of the ethanol production in *Saccharomyces cerevisiae* (GMA models from Polisetty et al., 2008). Gap: optimality gap.

	Polisetty et al.[a]	BARON	Proposed algorithm
Synthesis rate of ethanol (mM min^{-1})	157.59	157.59	157.59
UB	Not available	–	157.88
LB	157.59	–	157.59
Gap (%)	Not available	0.20	0.18
Iterations	–	–	3
Time (CPU s)	Not available	0.17	0.37

[a] Data termed as "Not available" is not shown in original work by Polisetty et al. (2008).

Table 3
Enzyme activities and metabolite concentrations (mM) in the global optimum for the ethanol synthesis rate in *Saccharomyces cerevisiae*.

i, *r*	1	2	3	4	5	6	7	8
K_r	5.00	0.89	5.00	0.20	1.25	0.20	5.00	5.00
X_i (mM)	0.35	1.06	91.44	0.01	1.25	–	–	–

5.1. Ethanol production in S. cerevisiae

This case study was solved using a tolerance (*tol*) of 0.20% and considering that all the enzymes of the network are subject to modification. For comparison purposes, we solved the same problem with the standard global optimization package BARON. In order to provide the solver with a feasible starting point, the basal state solution was used. Note that this point can be easily computed before the optimization takes place by fixing all the K_r to 1 in the original model and solving the resulting system of nonlinear equations.

As it can be observed in Table 2, the results produced by our algorithm and BARON are in consonance with those reported in the literature by Polisetty et al. (2008): 157.59 mM/min (see Table 3 for the enzyme activities and metabolite concentrations in the solution). This is indeed a problem of small size (see Table 1 for details) for which both algorithms are able to identify the global optimal solution in few seconds of CPU time.

In order to further illustrate the capabilities of our algorithm, we have reproduced (Table 4) some of the results reported in Polisetty et al. (2008) where only a set of reactions are allowed to be modified, whereas the remaining enzyme activities are constrained to their basal state (K_r = 1). These calculations provide valuable information as the implementation of solutions requiring a large number of genetic manipulations might be impractical due to their elevated cost and complexity. Again, we have chosen a tolerance of 0.20% for both, our algorithm and BARON.

As observed in Table 4, the three methods were capable of determining the global optimal solution in a similar CPU time for the 8 combinations of free reactions. BARON showed to be slightly faster than the other two algorithms. With regard to the quality of the solutions found, it is interesting to notice that the method proposed by Polisetty et al. (2008) provides very loose optimality gaps. Particularly, although the method finds the global optimum in all the cases, the reported optimality gaps are very large (i.e., more than 40%). This constitutes a major limitation of this strategy. Interestingly, we identified two cases (5 and 7) where the same values of the objective function were obtained through three different enzyme activities combinations. These results suggest that the problem possess a certain degree of degeneracy. This issue should be carefully studied before attempting to reproduce any of these solutions in the laboratory, as there might be some particular features not considered in the analysis that would make the imple-

Table 4
Results of the global optimization of the ethanol production in *Saccharomyces cerevisiae* when fixing all the enzyme activities, but two, to their basal state. LB: lower bound in mM min^{-1} (solution of ONLP). Gap: optimality gap.

		Polisetty et al.				BARON				Proposed method			
Case	Modified reactions *r*	[K_r]	LB	Gap (%)	Time[a] (CPU s)	[K_r]	LB	Gap (%)	Time (CPU s)	[K_r]	LB	Gap (%)	Time (CPU s)
1	[1, 3]	[5.00, 2.85]	103.66	21.66	0.81	[5.00, 2.85]	103.66	0.20	0.11	[5.00, 2.85]	103.66	0.09	0.94
2	[1, 4]	[5.00, 5.00]	73.18	48.96	0.26	[5.00, 5.00]	73.18	0.20	0.22	[5.00, 5.00]	73.18	0.16	2.03
3	[1, 7]	[5.00, 5.00]	73.41	47.46	0.20	[5.00, 5.00]	73.41	0.20	0.16	[5.00, 5.00]	73.41	0.11	2.17
4	[1, 6]	[5.00, 0.20]	73.41	47.15	0.24	[5.00, 0.20]	73.41	0.20	0.12	[5.00, 0.20]	73.41	0.11	2.72
5	[1, 5]	[5.00, 1.65]	72.68	48.47	0.22	[5.00, 0.63]	72.68	0.20	0.14	[5.00, 1.00]	72.68	0.11	2.63
6	[1, 8]	[5.00, 5.00]	87.77	22.13	0.19	[5.00, 5.00]	87.77	0.20	0.12	[5.00, 5.00]	87.77	0.14	2.59
7	[1, 2]	[5.00, 1.97]	72.68	47.48	0.24	[5.00, 5.00]	72.68	0.20	0.16	[5.00, 1.00]	72.68	0.11	2.49
8	[3, 7]	[5.00, 5.00]	44.67	76.18	0.09	[5.00, 5.00]	44.67	0.20	0.2	[5.00, 5.00]	44.67	0.08	1.82

[a] The author only reported the CPU time of the master MILP.

Table 5
Results of the global optimization of the citric acid production in *Aspergillus niger* (GMA models from Polisetty et al., 2008). Gap: optimality gap.

	Polisetty et al.	BARON	Proposed algorithm
Synthesis rate of citric acid (mM min^{-1})	384.23	Failed[a]	384.22
UB	384.23	–	390.66
LB	384.23	–	384.22
Gap (%)	0.00	–	1.68
Iterations	–	–	4
Time (CPU s)	5.68[b]	24,000	33.37

[a] Failed means that the optimality gap was higher than 100%.
[b] The author only reported the CPU time of the master MILP.

Table 6
Metabolite concentrations (mM) in the global optimum for the citric acid synthesis rate in *Aspergillus Niger*.

i	X_i	i	X_i	i	X_i	i	X_i	i	X_i
1	0.02	7	0.20	13	5.60	19	0.04	25	1.41
2	1.00	8	1.00×10^{-4}	14	0.01	20	0.85	26	0.98
3	0.12	9	21.21	15	31.26	21	0.71	27	0.30
4	0.02	10	130.00	16	0.52	22	0.35	28	0.26
5	0.71	11	0.01	17	1.70	23	0.01	29	6.00×10^{-8}
6	0.01	12	0.01	18	22.55	24	0.01	30	0.21

Table 7
Enzyme activities in the global optimum for the citric acid synthesis rate in *Aspergillus Niger*.

r	K_r	r	K_r	r	K_r	r	K_r	r	K_r
1	4.16	13	0.20	25	2.60	37	0.20	49	5.00
2	1.00	14	0.20	26	0.20	38	2.57	50	0.20
3	0.20	15	5.00	27	2.46	39	3.23	51	2.07
4	5.00	16	0.20	28	4.38	40	5.00	52	5.00
5	0.26	17	3.91	29	5.00	41	2.69	53	0.20
6	1.47	18	2.60	30	0.20	42	5.00	54	2.20
7	0.44	19	5.00	31	5.00	43	0.20	55	2.55
8	4.99	20	2.40	32	3.96	44	0.20	56	0.46
9	5.00	21	1.03	33	0.20	45	5.00	57	0.20
10	5.00	22	1.00	34	5.00	46	1.00	58	0.20
11	5.00	23	1.72	35	5.00	47	0.20	59	5.00
12	2.60	24	2.72	36	0.20	48	0.20	60	5.00

mentation of one of them advantageous when compared to the others.

5.2. *Citric acid production in A. niger*

The procedure explained for the first case study has been applied to solve the second case study with the only change of using a tolerance of 2.00%. The results obtained in the optimization are reported in Table 5 (the optimal values of the metabolite concentrations and the enzyme activities can be found in Tables 6 and 7, respectively).

This second case study considers a more complex network (4235 equations and 471 integer variables in the master problem vs 518 equations and 53 variables in the ethanol case). In this case, BARON failed at reducing the optimality gap below the specified tolerance (i.e., 2.00%) after 24,000 s of CPU time, whereas our algorithm was able to identify a solution in a relatively low computational time (i.e., less than 35 CPU seconds). In fact, after the aforementioned CPU time, BARON could only attain an optimality gap above 100%, which is very far away from the desired tolerance.

Additionally, we have applied our method to solve different cases where only a set of reactions were allowed to be modified. These cases have been selected from Polisetty et al. (2008). The results of these calculations are shown in Table 8.

Again, our method was able to provide the global optimal solution considering an optimality gap of 2.00% in few CPU seconds. Here, case E1 deserves particular attention since our solution slightly improves that obtained with Polisetty's approach. On the other hand, BARON could only reach an optimality gap above 100%.

Surprisingly, the method proposed by Polisetty et al. (2008) provides large optimality gaps for this case, where only a subset of the enzymes are subject to modification. On the other hand, this method is able to find the global optimum with a zero optimality gap for the case in which all the enzymes can be changed (Table 5).

Note that increasing the complexity of the model (i.e., increasing the number of reactions that can change) is not necessarily translated in bigger CPU times as one could expect. Although the CPU time required to solve a problem is generally ruled by the number

Table 8
Results of the global optimization of the citric acid production in *Aspergillus niger* when fixing some enzyme activities to their basal state. The number of reactions allowed to be modified depends on the case: Case B: one reaction; Case C: two reactions; Case D: three reactions; Case E: five reactions. LB: Lower bound in mM min^{-1} (solution of ONLP). Gap: optimality gap.

		Polisetty et al.				BARON	Proposed method			
Case	Modified reactions r	$[K_r]$	LB	Gap (%)	Time[a] (CPU s)	Results	$[K_r]$	LB	Gap (%)	Time (CPU s)
B	[40]	[5.00]	25.82	1234.12	9.11	Failed[b]	[5.00]	25.82	1.97	16.69
B	[59]	[1.00]	12.35	871.17	30.13	Failed	[1.00]	12.35	1.33	45.15
C	[40, 59]	[5.00, 1.00]	25.78	1254.54	13.27	Failed	[5.00, 1.00]	25.78	1.41	52.53
C	[1, 40]	[1.00, 5.00]	25.82	1241.75	26.4	Failed	[1.00, 5.00]	25.82	1.97	17.93
D	[1, 40, 60]	[1.27, 5.00, 1.12]	40.88	765.46	30.49	Failed	[1.27, 5.00, 1.12]	40.88	1.33	167.35
D	[1, 40, 59]	[1.16, 5.00, 5.00]	176.8	98.63	9.75	Failed	[1.16, 5.00, 5.00]	176.79	1.82	18.07
E	[1, 39, 40, 59, 60]	[1.24, 0.88, 5.00, 5.00, 1.01]	347.32	3.23	231.97	Failed	[1.40, 0.92, 5.00, 5.00, 1.07]	347.93	1.56	6.48
E	[1, 28, 40, 59, 60]	[1.46, 1.01, 5.00, 5.00, 1.11]	256.59	38.81	46.22	Failed	[1.46, 1.01, 5.00, 5.00, 1.11]	256.59	1.84	1093.09

[a] The author only reported the CPU time of the master MILP.
[b] Failed means that the optimality gap was higher than 100%.

Table 9
Local optimal solutions obtained by solving RNLP from different starting points.

Case	1	2	3	4	5
RNLP solution (mM min^{-1})	354.87	379.75	384.22	372.86	384.21

of variables and constraints and also by the quality of the relaxation (i.e., the difference between the lower bound obtained in the slave problem and the upper bound predicted by the master problem), there are other facts that can affect it. For instance, the way in which the branch and bound is implemented to solve the MILPs (i.e., branching rules, order in which the nodes are explored, derivation of cutting planes, etc.) can have a major influence on the total CPU time.

Finally, it is interesting to notice that during the calculations we confirmed the existence of multiple local optimal solutions in the *A. niger* model. For that, we solved the original non-convex **ONLP** with a local optimizer (i.e., CONOPT) using five different starting points that were calculated by solving different master problems **CMILP**, each of them with a different initial number of piecewise terms (from 1 to 5). The results obtained, which are given in Table 9, show that different local optima can be obtained depending on the starting point used in the initialization of the algorithm. This observation justifies the use of global optimization tools to avoid falling in local optima during the search (see Table 5 for the global optimum obtained).

6. Conclusions

This paper has addressed the development of a systematic framework for the global optimization of metabolic networks that can be described by the Generalized Mass Action model. The strategy proposed is based on reformulating the original GMA model via a logarithmic transformation, which gives rise to a non-convex NLP. This model is globally optimized by an outer approximation algorithm that exploits its specific structure.

The capabilities of the proposed method have been illustrated by globally optimizing the fermentation pathway of *S. cerevisiae* and the metabolic network associated with the citric acid production in *A. niger*. For both cases, we have obtained the appropriate changes that need to be performed in the corresponding enzyme activities in order to maximize the production of ethanol and citric acid, respectively. Our algorithm has been able to reproduce the results previously reported by Polisetty et al. (2008), but achieving significant improvements in the optimality gaps of the final solutions. Besides, the method proposed has shown promising results even when applied to a moderately complex network (case study 2), absolutely surpassing the performance of BARON, which failed to solve that particular example within the predefined tolerance.

The generality of the optimization framework introduced in this paper makes it very interesting for biotechnological applications. At this point, the major drawbacks for getting practical results are: (1) the ability of obtaining appropriate models; and (2) the possibility of effectively manipulating the required enzymes. The main limitation for obtaining good mathematical models is the lack of experimental data that can be used for parameter estimation (Chou & Voit, 2009). Unfortunately, most of the Systems Biology effort has focused on gene sequences, protein structures, and so on, with relatively few results on actual data on intact systems. The kind of data required for this task would involve measuring metabolites and fluxes in vivo, a problem that is not totally solved yet. The optimization method presented here can yield valuable insights if and only if the underlying model is a good approximation to reality. Besides this problem, optimization results require experimental confirmation; that is manipulation of enzymes for obtaining the desired optimal increment on the objective function. However, although gene expression changes can be easily introduced in living cells, there is no guaranty that an appropriate change in enzyme activity is also obtained.

In conclusion, our results show that it is possible to appropriately analyze a highly non-linear mathematical model and obtain optimal solution for a given objective function. This should stimulate experimentalists for developing appropriate tools for measuring living cells and for manipulating them so that practical results can be obtained.

Acknowledgements

The authors wish to acknowledge support of this research work from the Spanish Ministry of Education and Science (projects BFU2005-00234/BMC, BFU2008-00196/BMC, DPI2008-04099, PHB2008-0090-PC, BFU2008-00196 and CTQ2009-14420-C02), the Spanish Ministry of External Affairs (projects HS2007-0006, A/020104/08 and A/023551/09), and the Generalitat de Catalunya (FI programs).

Appendix A.

Lemma 1 shows that the maximum error between the linear outer approximation and the logarithmic function lies in a vertex. Proposition 1 uses the results of Lemma 1 to show that the allocation of hyper-planes that minimizes the rectilinear distance (i.e., L_1 norm) between the hyper-planes and the actual logarithmic function is that in which this distance is the same at every interjection of two adjacent hyper-planes. Lemma 2 and Proposition 2 are similar to Lemma 1 and Proposition 1 but apply to the piece-wise approximation. Finally, Proposition 3 complements Proposition 2 and provides a direct way of defining a piece-wise approximation with minimum error.

Lemma 1. *Consider an outer approximation of the function* $\ln v_r$ *with* L *supporting hyper-planes (see Fig. 6). The maximum error*, $error_{max}$, *(defined as the linear distance*, L_1 *norm), between the hyper-planes and the logarithmic function is attained in a vertex.*

Proof. We first show that the point with the maximum error lies in a hyperplane, and then prove that it must correspond to one of its intersections with adjacent hyperplanes. Consider problem **PA**, which seeks the maximum difference between a set of hyperplanes and the logarithmic function:

$$
\begin{array}{lll}
(\mathbf{PA}) & \min & \ln v_r - y \\
 & s.t. & y - \left(\ln v_r^l + \dfrac{1}{v_r^l}\left(v_r - v_r^l\right)\right) \le 0 \quad l = 1, \ldots, L \\
 & & v_r - \overline{v_r} \le 0 \\
 & & \underline{v_r} - v_r \le 0 \\
 & & y \in \mathbb{R} \\
 & & v_r \in \mathbb{R}_+
\end{array}
$$

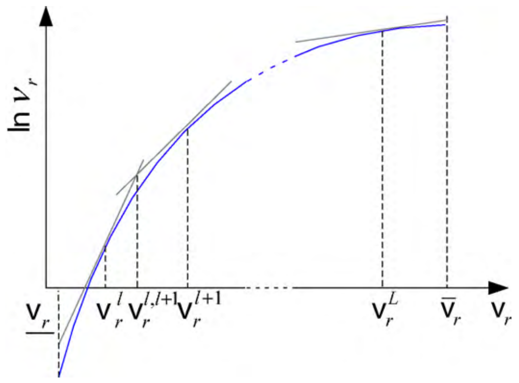

Fig. 6. Approximation of the $\ln v_r$ function by L supporting hyper-planes.

where $\underline{v_r} \leq v_r^l \leq \overline{v_r}$. The Karush-Kunh-Tucker (KKT) conditions of **PA** are:

$$-1+\sum_{l=1}^{L} u_1^l = 0 \tag{18}$$

$$\frac{1}{v_r}-\sum_{l=1}^{L}\frac{u_1^l}{v_r^l}+u_2-u_3=0 \tag{19}$$

$$u_1^l\left(y-\left(\ln v_r^l+\frac{1}{v_r^l}\left(v_r-v_r^l\right)\right)\right)=0 \tag{20}$$

$$u_2(v_r-\overline{v_r})=0 \tag{21}$$

$$u_3(\underline{v_r}-v_r)=0 \tag{22}$$

$$u_1^l \geq 0 \tag{23}$$

$$u_2 \geq 0 \tag{24}$$

$$u_3 \geq 0 \tag{25}$$

From Eq. (18), it follows that at least one supporting hyper-plane (*SH*) must be active in the optimal solution. Now, consider problem **PB** that seeks the maximum error along the active SH_l between its extremes v_r^{lo} and v_r^{up}, which are given by the intersection of the hyper-plane with either an adjacent SH_j or a limit of the interval $[\underline{v_r}, \overline{v_r}]$.

$$\begin{array}{lll}(\mathbf{PB}) & \min & \ln v_r-\left(\ln v_r^l+\dfrac{1}{v_r^l}\left(v_r-v_r^l\right)\right)\\ & s.t. & v_r-v_r^{up}\leq 0\\ & & v_r^{lo}-v_r\leq 0\\ & & v_r\in\mathbb{R}_+\end{array}$$

The KKT conditions of **PB** are:

$$\frac{1}{v_r}-\frac{1}{v_r^l}+u_1-u_2=0 \tag{26}$$

$$u_1(v_r-v_r^{up})=0 \tag{27}$$

$$u_2(v_r^{lo}-v_r)=0 \tag{28}$$

$$u_1 \geq 0 \tag{29}$$

$$u_2 \geq 0 \tag{30}$$

There are 3 possible solutions to this problem.

Case 1: $u_1=u_2=0$. From Eq. (26), we have:

$$v_r^*=v_r^l$$

And the resulting value of the objective function is:

$$OF=0$$

It is easy to see that this point is a maximum of **PB** in which the error is minimum. Note that this is the point in which the hyper-plane touches the logarithmic function.

Case 2: $u_1=0$, $u_2\neq 0$. From Eqs. (26) and (28), we get:

$$v_r^*=v_r^{lo};\ u_2=\frac{1}{v_r^{lo}}-\frac{1}{v_r^l}\geq 0 \quad OF=\ln\left(\frac{v_r^{lo}}{v_r^l}\right)-\frac{v_r^{lo}}{v_r^l}+1$$

Hence, this point is a minimum of **PB** and corresponds to a vertex.

Case 3: $u_2=0$, $u_1\neq 0$. From Eqs. (26) and (27), we get:

$$v_r^*=v_r^{up};\ u_1=\frac{1}{v_r^l}-\frac{1}{v_r^{up}}\geq 0 \quad OF=\ln\left(\frac{v_r^{up}}{v_r^l}\right)-\frac{v_r^{up}}{v_r^l}+1$$

This point (again a vertex) is another minimum of **PB**. Hence, the solution of **PB** must correspond to a vertex, and the proof is complete. □

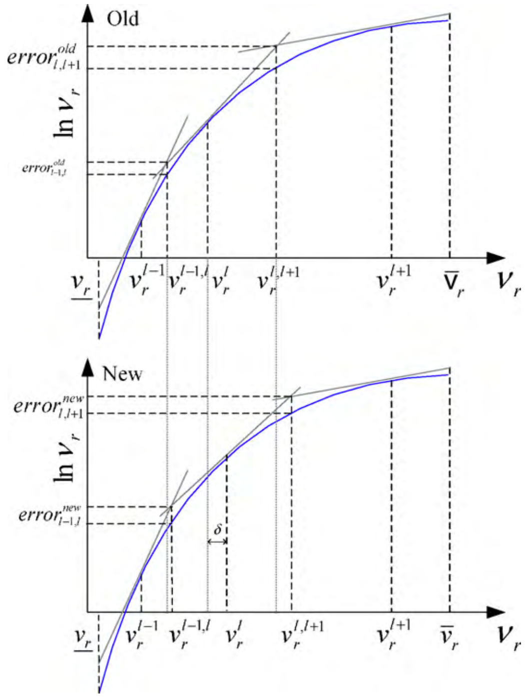

Fig. 7. Illustration of the decrease in $error_{max}$ by moving SH_l a distance δ towards the vertex $v_r^{l,l+1}$.

Proposition 1. *The allocation of L hyper-planes that minimizes* $error_{max}$ *is that in which the error is the same in all the L+1 vertexes.*

Proof. The proof is by contradiction. From Lemma 1, we know that the maximum error between the hyper-planes and the logarithmic function is attained in a vertex. Assume that in the optimal allocation there is at least one vertex $v_r^{l,l+1}$ with a different error. Now, identify a supporting hyperplane SH_l with different errors in its extreme vertexes ($error_{l-1,l}$ in $v_r^{l-1,l}$ and $error_{l,l+1}$ in $v_r^{l,l+1}$). Assume, without loosing generality, that $error_{l,l+1}\geq error_{l-1,l}$. Now, we consider two cases:

Case 1: the maximum $error_{max}=\max_{l\neq l'}\{error_{l,l'}\}$ corresponds to the right vertex $v_r^{l,l+1}$ of the selected hyperplane, that is, $error_{max}=error_{l,l+1}$. Now, define $error^{old}_{l,l+1}=error^{old}_{max}$ and move the hyperplane SH_l a small distance δ towards $v_r^{l,l+1}$, that is, make $v_r^{l,l+1new}=v_r^{l,l+1old}+\delta$, thus decreasing the slope of SH_l. This move decreases $error_{l,l+1}$ at the expense of increasing $error_{l-1,l}$. Since the logarithmic function is continuous, it is possible to find δ such that $error^{old}_{l-1,l}<error^{new}_{l-1,l}=error^{new}_{l,l+1}<error^{old}_{l,l+1}$ (Fig. 7), that is, a new solution with a smaller error in the right vertex of SH_l, and hence with a smaller $error_{max}$. This contradicts the fact that in the optimal solution there are vertexes with different errors.

Case 2: $error_{max}$ is placed in another hyper-plane $SH_{l'}$ ($l'\neq l, l+1$). In this case, the same procedure can be repeated recursively to the rest of the hyper-planes until no more hyper-planes containing different errors in their vertexes remain. It is straightforward to see that this would lead to a solution with lower $error_{max}$, which contradicts the assumption that the optimal allocation implies the existence of at least one hylperplane with different errors in its extremes vertexes. □

Lemma 2. *Consider an underestimation of the* $\ln v_r$ *function with a linear piecewise (PW) section* (Fig. 8). *The maximum error,* $error_{max}$, *defined as the* L_1 *norm* (i.e., $error(v_r)=\ln v_r-av_r-b$) *between the* $\ln v_r$ *and the PW linear function occurs at* $v_r^*=1/a$

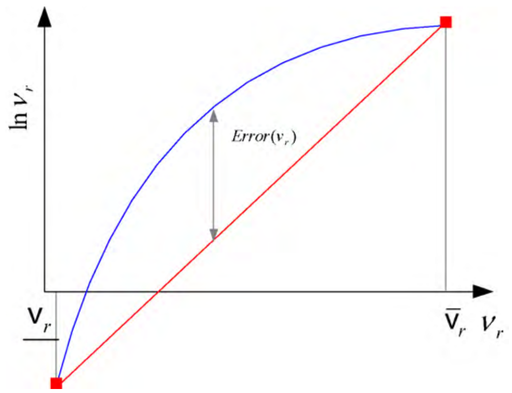

Fig. 8. Approximation of the ln v_r function by a one interval linear function.

Proof. The error is a concave function that depends on a single variable, hence in the optimal solution we get:

$$error' = \frac{1}{v_r} - a = 0 \leftrightarrow v_r^* = \frac{1}{a} \qquad error'' = -\frac{1}{v_r^2} \le 0$$

Therefore, $v_r^* = \frac{1}{a}$ is a maximum of the function *error*. □

Proposition 2. *Consider a piece-wise approximation of the function* ln v_r *with H intervals. Let* $error_h$ *be the maximum error in the* h^{th} *interval of the PW function and* $error_{max}$ *the maximum error among the different intervals* ($error_{max} = \max_h\{error_h\}$)*. The piece-wise approximation that minimizes* $error_{max}$ *is that in which* $error_h = error_{h'}$, $\forall h,h'(h \neq h')$.

Proof. The proof is by contradiction. From Lemma 2 we know that the maximum error in a piecewise section PW_h is attained at $1/a$. Assume that the optimal distribution of the domain is that where there is at least one section h with a different error, $error_h > error_{h+1}$. Consider the following cases:

Case 1: $error_h = error_{max}$. Move the grid point $\overline{v_r^h} = \underline{v_r^{h+1}}$ a distance δ towards $\underline{v_r^h}$, that is, make $\overline{v_r^{hnew}} = \overline{v_r^{hold}} - \delta$, thus increasing a_h and decreasing a_{h+1}. This decreases $error_h$ and increases $error_{h+1}$. Since the logarithmic function is continuous, we can define a δ such that $error_{h+l}^{old} < error_{h+l}^{new} = error_h^{new} < error_h^{old}$ (Fig. 9), that is, a new solution with a smaller $error_{max}$. This contradicts the original statement that in the optimal solution there are sections with different errors.

Case 2: $error_{max} = error_{h'} \neq error_h$ $(h' \neq h)$. It is straightforward to see that we can follow the same strategy described before until the error in every couple of adjacent sections is the same and smaller than $error_{max}^{old}$. □

Proposition 3. *The distribution of the H intervals where* $error_h = error_{h'}$, $\forall h,h'(h \neq h')$ *corresponds to that where all PW sections are of equal width Q in the logarithmic space.*

Proof. Consider two piecewise sections[2] PW_h and PW_{h+1} defined by grid point $\underline{v_r^{h+1}}$. When $error_h = error_{h+1}$ the following relationship holds:

$$\ln\left(\frac{1}{a_h}\right) - 1 - b_h = \ln\left(\frac{1}{a_{h+1}}\right) - 1 - b_{h+1}$$

$$\rightarrow \ln\left(\frac{a_{h+1}}{a_h}\right) - (b_h - b_{h+1}) = 0 \qquad (31)$$

As the piecewise functions take the same value at the common grid point $\overline{v_r^h} = \underline{v_r^{h+1}}$, we can rewrite Eq. (31) in terms of a_h, a_{h+1} and $\underline{v_r^{h+1}}$:

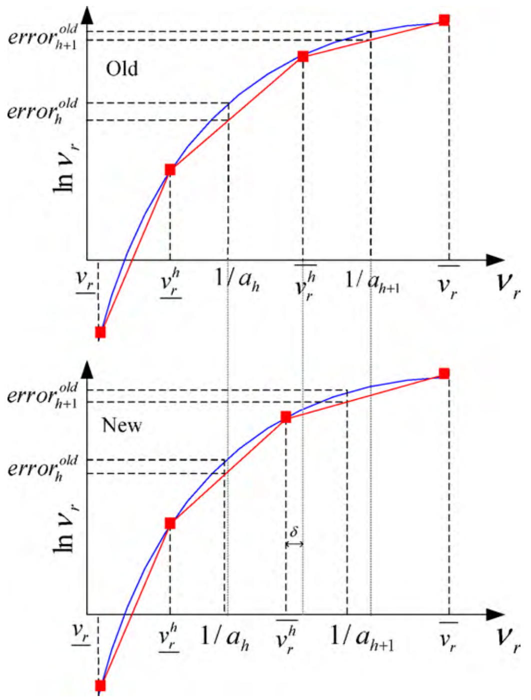

Fig. 9. Illustration of the decrease in $error_{max}$ by moving the grid point $\overline{v_r^h}$ a distance δ.

$$\ln\left(\frac{a_{h+1}}{a_h}\right) - \underline{v_{rh+1}}(a_{h+1} - a_h) = 0 \qquad (32)$$

Now, we introduce two new variables Q and Q' defined as:

$$Q = \ln \overline{v_r^h} - \ln \underline{v_r^h} \qquad (33)$$

$$Q' = \ln \overline{v_r^{h+1}} - \ln \underline{v_r^{h+1}} \qquad (34)$$

Hence, we can express the width of each interval in the cartesian space (i.e., $\overline{v_r^h} - \underline{v_r^h}$ and $\overline{v_r^{h+1}} - \underline{v_r^{h+1}}$) in terms of Q and Q':

$$\underline{v_r^h} = \frac{\overline{v_r^h}}{\exp Q} \rightarrow \overline{v_r^h} - \underline{v_r^h} = \frac{\overline{v_r^h}}{\exp Q}(\exp Q - 1) \qquad (35)$$

$$\underline{v_r^{h+1}} = \frac{\overline{v_r^{h+1}}}{\exp Q'} \rightarrow \overline{v_r^{h+1}} - \underline{v_r^{h+1}} = \overline{v_r^{h+1}}(\exp Q' - 1) \qquad (36)$$

Similarly, we can redefine the slope of each of the linear piecewise functions, a_h and a_{h+1}, in terms of Q, Q' and $\underline{v_r^{h+1}}$:

$$a_h = \frac{\ln \overline{v_r^h} - \ln \underline{v_r^h}}{\overline{v_r^h} - \underline{v_r^h}} = \frac{Q(\exp Q)}{\underline{v_r^{h+1}}(\exp Q - 1)} \qquad (37)$$

$$a_{h+1} = \frac{\ln \overline{v_r^{h+1}} - \ln \underline{v_r^{h+1}}}{\overline{v_r^{h+1}} - \underline{v_r^{h+1}}} = \frac{Q'}{\underline{v_r^{h+1}}(\exp Q' - 1)} \qquad (38)$$

By introducing Eqs. (37) and (38) into Eq. (32), the following equality is obtained:

$$\ln\left(\frac{Q'(\exp Q - 1)}{(\exp Q' - 1)Q(\exp Q)}\right) - \left(\frac{Q'}{(\exp Q' - 1)} - \frac{Q(\exp Q)}{(\exp Q - 1)}\right) = 0 \qquad (39)$$

When $Q' = Q$ (i.e., when the intervals are of equal width in the logarithmic space) this equation is satisfied. □

[2] Note that more intervals could be considered and generality would not be lost.

References

Alvarez-Vasquez, F., Canovas, M., Iborra, J., & Torres, N. (2002). Modeling, optimization and experimental assessment of continuous l-(-)-carnitine production by *Escherichia coli* cultures. *Biotechnology and Bioengineering*, *80*(7), 794–805.

Alves, R., Vilaprinyo, E., Hernndez-Bermejo, B., & Sorribas, A. (2009). Mathematical formalisms based on approximated kinetic representations for modeling genetic and metabolic pathways. *Biotechnology & Genetic Engineering Reviews*, *25*, 1–40.

Bailey, J. (1991). Toward a science of metabolic engineering. *Science*, *252*, 1668–1675.

Bailey, J. (1999). Lessons from metabolic engineering for functional genomics and drug discovery. *Nature Biotechnology*, *17*, 616–618.

Bailey, J., Birnbaum, S., Galazzo, J., Khosla, C., & Shanks, J. (1990). Strategies and challenges in metabolic engineering. *Annals of the New York Academy of Sciences*, *589*, 1–15.

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

Bergamini, M., Aguirre, P., & Grossmann, I. (2005). Logic-based outer approximation for globally optimal synthesis of process networks. *Computers and Chemical Engineering*, *29*, 1914–1933.

Cameron, D., & Chaplen, F. (1997). Developments in metabolic engineering. *Current Opinion in Biotechnology*, *8*, 175–180.

Cameron, D., & Tong, J. (1993). Cellular and metabolic engineering: An overview. *Applied Biochemistry and Biotechnology*, *38*, 105–140.

Chang, Y., & Sahinidis, N. (2005). Optimization of metabolic pathways under stability considerations. *Computers and Chemical Engineering*, *29*, 467–479.

Chou, I., & Voit, E. (2009). Comparative characterization of the fermentation pathway of *Saccharomyces cerevisiae* using biochemical systems theory and metabolic control analysis: Model definition and nomenclature. *Mathematical Biosciences*, *219*, 57–83.

Curto, R., Sorribas, A., & Cascante, M. (1995). Comparative characterization of the fermentation pathway of *Saccharomyces cerevisiae* using biochemical systems theory and metabolic control analysis: Model definition and nomenclature. *Mathematical Biosciences*, *130*, 25–50.

Floudas, C., & Gounaris, C. (2009). A review of recent advances in global optimization. *Journal of Global Optimization*, *45*, 3–38.

Gavalas, G. (1968). *Nonlinear differential equations of chemical reacting systems*. Berlin: Springer-Verlag.

Grossmann, I., & Bigler, L. (2004). Part ii. future perspective on optimization. *Computers and Chemical Engineering*, *28*, 1193–1218.

Guillén-Gosálbez, G., & Sorribas, A. (2009). Identifying quantitative operation principles in metabolic pathways: A systematic method for searching feasible enzyme activity patterns leading to cellular adaptive responses. *BMC Bioinformatics*, *10*(386).

Hatzimanikatis, V., Floudas, C., & Bailey, J. (1996). Optimization of regulatory architectures in metabolic reaction networks. *Biotechnology and Bioengineering*, *52*, 485–500.

Heinrich, R., & Schuster, S. (1996). *The regulation of cellular systems*. New York: Chapman and Hall.

Marin-Sanguino, A., & Torres, N. (2003). Optimization of biochemical systems by linear programming and general mass action model representations. *Mathematical Biosciences*, *184*(2), 187–200.

Marin-Sanguino, A., Voit, E., Gonzalez-Alzon, C., & Torres, N. (2007). Optimization of biotechnological systems through geometric programming. *Theoretical Biology & Medical Modelling*, 4–38.

Mendes, P., & Kell, D. (1996). Making cells work – metabolic engineering for everyone. *Trends in Biotechnology*, *15*, 6–7.

Polisetty, P., Gatzke, E., & Voit, E. (2008). Yield optimization of regulated metabolic systems using deterministic branch-and-reduce methods. *Biotechnology and Bioengineering*, *99*(5), 1154–1169.

Savageau, M. (1969a). Biochemical systems analysis. i. Some mathematical properties of the rate law for the component enzymatic reactions. *Journal of Theoretical Biology*, *25*, 365–369.

Savageau, M. (1969b). Biochemical systems analysis. ii. The steady-state solutions for an n-pool system using a power-law approximation. *Journal of Theoretical Biology*, *25*, 370–379.

Sorribas, A., Hernndez-Bermejo, B., Vilaprinyo, E., & Alves, R. (2007). Cooperativity and saturation in biochemical networks: A saturable formalism using Taylor series approximations. *Biotechnology and Bioengineering*, *97*, 1259–1277.

Sorribas, A., Pozo, C., Vilaprinyo, E., Guillén-Gosálbez, G., Jiménez, L., & Alves, R. (2010). Global optimization techniques in Generalized Mass Action models. *J. Biotechnol.*, doi:10.1016/j.jbiotec.2010.01.026

Torres, N., & Voit, E. (2002). *Pathway analysis and optimization in metabolic engineering*. Cambridge: Cambridge University Press.

Vecchietti, A., Sangbum, L., & Grossmann, I. (2003). Modeling of discrete/continuous optimization problems: Characterization and formulation of disjunctions and their relaxations. *Computers and Chemical Engineering*, *27*, 433–448.

Vera, J., de Atauri, P., Cascante, M., & Torres, N. (2003). Multicriteria optimization of biochemical systems by linear programming: Application to production of ethanol by *Saccharomyces cerevisiae*. *Biotechnology and Bioengineering*, *83*(3), 335–343.

Voit, E. (1992). Optimization in integrated biochemical systems. *Biotechnology and Bioengineering*, *40*(5), 572–582.

Voit, E. (2000). *Computational analysis of biochemical systems. A practical guide for biochemists and molecular biologists*. Cambridge: Cambridge University Press.

Voit, E. (2003). Design principles and operating principles: the yin and yang of optimal functioning. *Mathematical Biosciences*, *182*, 81–92.