

Research

Open Access

Optimization of biotechnological systems through geometric programming

Alberto Marin-Sanguino*¹, Eberhard O Voit², Carlos Gonzalez-Alcon³ and Nestor V Torres¹

Address: ¹Grupo de Tecnología Bioquímica. Departamento de Bioquímica y Biología Molecular, Facultad de Biología, Universidad de La Laguna, 38206 La Laguna, Tenerife, Islas Canarias, Spain, ²The Wallace H. Coulter Department of Biomedical Engineering at Georgia Institute of Technology and Emory University, 313 Ferst Drive, Atlanta, GA, 30332, USA and ³Grupo de Tecnología Bioquímica. Departamento de Estadística Investigación Operativa y Computación, Facultad de Física y Matemáticas, Universidad de La Laguna, 38206 La Laguna, Tenerife, Islas Canarias, Spain

Email: Alberto Marin-Sanguino* - amarin@ull.es; Eberhard O Voit - voit@bme.gatech.edu; Carlos Gonzalez-Alcon - cgalcon@ull.es; Nestor V Torres - ntorres@ull.es

* Corresponding author

Published: 26 September 2007

Received: 27 May 2007

Theoretical Biology and Medical Modelling 2007, **4**:38 doi:10.1186/1742-4682-4-38

Accepted: 26 September 2007

This article is available from: <http://www.tbiomed.com/content/4/1/38>

© 2007 Marin-Sanguino et al; licensee BioMed Central Ltd.

This is an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/2.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Abstract

Background: In the past, tasks of model based yield optimization in metabolic engineering were either approached with stoichiometric models or with structured nonlinear models such as S-systems or linear-logarithmic representations. These models stand out among most others, because they allow the optimization task to be converted into a linear program, for which efficient solution methods are widely available. For pathway models not in one of these formats, an Indirect Optimization Method (IOM) was developed where the original model is sequentially represented as an S-system model, optimized in this format with linear programming methods, reinterpreted in the initial model form, and further optimized as necessary.

Results: A new method is proposed for this task. We show here that the model format of a Generalized Mass Action (GMA) system may be optimized very efficiently with techniques of geometric programming. We briefly review the basics of GMA systems and of geometric programming, demonstrate how the latter may be applied to the former, and illustrate the combined method with a didactic problem and two examples based on models of real systems. The first is a relatively small yet representative model of the anaerobic fermentation pathway in *S. cerevisiae*, while the second describes the dynamics of the tryptophan operon in *E. coli*. Both models have previously been used for benchmarking purposes, thus facilitating comparisons with the proposed new method. In these comparisons, the geometric programming method was found to be equal or better than the earlier methods in terms of successful identification of optima and efficiency.

Conclusion: GMA systems are of importance, because they contain stoichiometric, mass action and S-systems as special cases, along with many other models. Furthermore, it was previously shown that algebraic equivalence transformations of variables are sufficient to convert virtually any types of dynamical models into the GMA form. Thus, efficient methods for optimizing GMA systems have multifold appeal.

Background

Model based optimization of biotechnological processes is a key step towards the establishment of rational strategies for yield improvement, be it through genetic engineering, refined setting of operating conditions or both. As such, it is a key element in the rapidly emerging field of metabolic engineering [1,2]. Optimization tasks involving living organisms are notoriously difficult, because they almost always involve large numbers of variables, representing biological components that dominate cell operation, and must account for multitudinous and complex nonlinear interactions among them [3]. The steady increase in the ready availability of computing power has somewhat alleviated the challenge, but it has also, together with other technological breakthroughs, been raising the level of expectation. Specifically, modelers are more and more expected to account for complex biological details and to include variables of diverse types and origins (metabolites, RNA, proteins...). This trend is to be welcomed, because it promises improved model predictions, yet it easily compensates for the computer technological advances and often overwhelms available hardware and software methods. As a remedy, effort has been expanded to develop computationally efficient algorithms that scale well with the growing number of variables in typical optimization tasks.

The most straightforward attempts toward improved efficiency have been based, in one form or another, on the reduction of the originally nonlinear task to linearity, because linear optimization tasks are rather easily solved, even if they involve thousands of variables. One variant of this approach is the optimization of stoichiometric flux distribution models [4]. The two great advantages of this method are that the models are linear and that minimal information is needed to implement them, namely flux rates, and potentially numerical values characterizing metabolic or physico-chemical constraints. The significant disadvantage is that no regulation can be considered in these models.

An alternative is the use of S-system models within the modeling framework of Biochemical Systems Theory [5-7]. These models are highly nonlinear, thus allowing suitable representations of regulatory features, but have linear steady-state equations, so that optimization under steady-state conditions again becomes a matter of linear programming [8]. The disadvantages here are that much more (kinetic) information is needed to set up numerical models and that S-systems are based on approximations that are not always accepted as valid. Linear-logarithmic models [9] similarly have the advantage of linearity at steady state and the disadvantage of being a local approximation.

An extension of these linear approaches is the Indirect Optimization Method [10]. In this method, any type of kinetic model is locally represented as an S-system. This S-system is optimized with linear methods, and the resulting optimized parameter settings are translated back into the original model. If necessary, this linearized optimization may be executed in sequential steps.

An alternative to using S-system models is the General Mass Action (GMA) representation within BST. GMA systems are very interesting for several reasons. First, they contain both stoichiometric and S-system models as direct special cases, which would allow the optimization of combinations of the two. Second, mass action systems are special cases of GMA models, so that, in some sense, Michaelis-Menten functions and other kinetic rate laws are special cases, if they are expressed in their elemental, non-approximated form. Third, it was shown that virtually any system of differential equations may be represented exactly as a GMA system, upon equivalence transformations of some of the functions in the original system. Thus, GMA systems, as a mathematical representation, are capable of capturing any differentiable nonlinearity that one might encounter in biological systems. We show here that GMA systems, while highly nonlinear, are structured enough to permit the application of efficient optimization methods based on geometric programming.

Formulation of the optimization task

Pertinent optimization problems in metabolic engineering can be stated as the targeted manipulation of a system in the following way:

$$\max \text{ or } \min f_0(X) \quad (1)$$

subject to:

$$\text{operation in steady state} \quad (2)$$

$$\text{metabolic and physico-chemical constraints} \quad (3)$$

$$\text{cell viability} \quad (4)$$

In this generic representation, (1) usually targets a flux or a yield. The optimization must occur under several constraints. The first set (2) ensures that the system will operate under steady-state conditions. Other constraints (3) are imposed to retain the system within a physically and chemically feasible state and so that the total protein or metabolite levels do not impede cell growth. Yet other constraints (4) guarantee that no metabolites are depleted below minimal required levels or accumulate to toxic concentrations. These sets of constraints are designed to allow sustained operation of the system.

Biochemical Systems Theory (BST)

Biological processes are usually modeled as systems of differential equations in which the variation in metabolites X is represented as:

$$\frac{dX}{dt} = N \cdot v \tag{5}$$

The elements $n_{i,j}$ of the stoichiometric matrix N are constant. The vector v contains reaction rates, which are in general functions of the variables and parameters of the system. This structure is usually associated with metabolic systems, but it is similarly valid for models describing gene expression, bioreactors, and a wide variety of other processes in biotechnology. In typical stoichiometric analyses, the reaction rates are considered constant. Furthermore, the analysis is restricted to steady-state operation, with the consequence that (5) is set equal to 0 and thereby becomes a set of linear algebraic equations, which are amenable to a huge repertoire of analyses.

In analyses accounting for regulation, the reaction rates become functions that depend on system variables and outside influences. Even at steady state, these may be very complex, thereby rendering direct analysis of the system a formidable task [11]. As a remedy, BST suggests to represent these rate functions with power laws:

$$v_i = \gamma_i \prod_{j=1}^{n+m} X_j^{f_{i,j}} \tag{6}$$

In analogy with chemical kinetics, γ_i is called the rate constant and $f_{i,j}$ are kinetic orders, which may be any real numbers. Positive kinetic orders indicate augmentation, whereas negative values are indicative of inhibition. Kinetic orders of 0 result in automatic removal of the corresponding variable from the term. In the notation of BST, the first n variables are often considered the *dependent variables*, which change dynamically under the action of the system, while the remaining variables X_i for $i = n + 1 \dots m + n$ are considered *independent variables* and typically remain constant throughout any given simulation study. Thus, metabolites, enzymes, membrane potentials or other system components can easily be made dependent or independent by the modeler without requiring alterations in the structure of the equations. BST is very compact and explicitly distinguishes variables from parameters.

Because we will later introduce concepts of geometric programming, it is noted that the power-law term in Eq. 6 is also called a *monomial*. If this monomial is an approximation of reaction rate V , its parameters can be directly related to V , by virtue of the fact that the monomial is in fact a Taylor linearization in logarithmic space [12]. Thus, choosing an operating point with index 0, one obtains:

$$\ln v_i = \ln v_{i0} + \left| \frac{\partial \ln v_i}{\partial \ln X_1} \right|_0 (\ln X_1 - |\ln X_1|_0) + \dots + \left| \frac{\partial \ln v_i}{\partial \ln X_{m+n}} \right|_0 (\ln X_{m+n} - |\ln X_{m+n}|_0) \tag{7}$$

Thus, it follows directly from 7 that the parameters of a power-law (monomial) term can be computed as

$$\gamma_i = \frac{|v_i|_0}{\left| \prod_{j=1}^{n+m} X_j^{f_{i,j}} \right|_0} \tag{8}$$

$$f_{i,j} = \left| \frac{\partial \ln v_i}{\partial \ln X_j} \right|_0 = \left| \frac{\partial v_i}{\partial X_j} \frac{X_j}{v_i} \right|_0 \tag{9}$$

System equations in BST may be designed in slightly different ways. For the GMA form, each reaction is represented by its own monomial, and the result is therefore

$$\frac{dX_i}{dt} = \sum_{j=1}^p n_{i,j} \gamma_j \prod_{k=1}^{n+m} X_k^{f_{j,k}} \quad i = 1 \dots n \tag{10}$$

Note that this is actually a spelled-out version of Eq. 5, where the reaction rates are monomials as in Eq. 6. As an alternative to the GMA format, one may, for each dependent variable, collect all incoming reactions in one term V_i^+ and do the same with all outgoing fluxes, which are collectively called V_i^- . These *aggregated* terms are now represented as monomials, and the result is

$$\frac{dX_i}{dt} = V_i^+ - V_i^- = \alpha_i \prod_{j=1}^{n+m} X_j^{g_{i,j}} - \beta_i \prod_{j=1}^{n+m} X_j^{h_{i,j}} \tag{11}$$

Thus, there are at most one positive and one negative term in each S-system equation.

The conversion of a GMA into an S-system will become important later. It is achieved by collecting the aggregated fluxes into vectors

$$\begin{aligned} V^+ &= N^+ v \\ V^- &= N^- v \end{aligned} \tag{12}$$

where N^+ and N^- are matrices containing respectively the positive and negative coefficients of N such that $N = N^+ - N^-$. With these definitions, we can derive the matrices of kinetic orders of S-systems from those of the corresponding GMA representation. Namely,

$$G = (V^+)^{-1}N^+VF \tag{13}$$

$$H = (V^-)^{-1}N^-VF$$

where V , V^+ and V^- are square matrices of zeros having the corresponding vectors as their main diagonals. G and H contain the kinetic orders of the S-system while F contains those of the GMA [13]. GMA systems may be constructed in three manners [11]. First, given a pathway diagram, each reaction rate is represented by a monomial, and equations are assembled from all reaction rates involved. Second, it is possible (though not often actually done) to dissect enzyme catalyzed reactions into their underlying mass action kinetics, without evoking the typical quasi-steady-state assumption. The result is directly the special case of a GMA system where most kinetic orders are zero, one, or in some cases 2. Third, it has been shown that virtually any nonlinearity can be represented equivalently as a GMA system [14]. As an example for this *recasting* technique, consider a simple equation where production and degradation are formulated as traditional Michaelis-Menten rate laws:

$$\frac{dX_1}{dt} = \frac{V_{max,1}X_0}{K_{M,1} + X_0} - \frac{V_{max,2}X_1}{K_{M,2} + X_1} \tag{14}$$

where X_0 is a dependent or independent variable describing the substrate for the generation of X_1 . To effect the transformation into a GMA equation, define auxiliary variables as $X_2 = K_{M,2} + X_1$ and $X_3 = K_{M,1} + X_0$. The equation then becomes

$$\frac{dX_1}{dt} = V_{max,1}X_0X_3^{-1} - V_{max,2}X_1X_2^{-1} \tag{15}$$

For simplicity of discussion, suppose that X_0 is a constant, independent variable. Thus, X_3 is also constant and does not need its own equation. By contrast, X_2 is a new dependent variable and from its definition we can calculate its initial value and see that its derivative must be equal to that of X_1 . Therefore the equations:

$$\frac{dX_1}{dt} = V_{max,1}X_0X_3^{-1} - V_{max,2}X_1X_2^{-1}$$

$$\frac{dX_2}{dt} = V_{max,1}X_0X_3^{-1} - V_{max,2}X_1X_2^{-1} \tag{16}$$

$$X_1(t_0) = X_1^0$$

$$X_2(t_0) = K_{M,2} + X_1^0$$

form a system that is an exact equivalent of the original system but in GMA format.

Recasting can be useful with equations that are difficult to handle otherwise or for purposes of streamlining a model

structure and its analysis. One must note though that often the number of variables increases significantly. In the case shown, the number of equations rises from one to two if X_0 is independent or to three if it is a dependent variable.

Current optimization methods based on BST

The overall task is to reset some of the independent variables so that some objective is optimized. The independent variables in question are typically enzyme activities, which are experimentally manipulated through genetic means, such as the application of customized promoters or plasmids. The objective is usually the maximization of a metabolite concentration or a flux. Three approaches have been proposed in the literature.

Pure S-systems

Among a number of convenient properties, the steady states of an S-system can be computed analytically by solving a system of algebraic linear equation [6]. Equating Eq. 11 to zero and rearranging one obtains:

$$\frac{\alpha_i \prod_{j=1}^n X_j^{g_{i,j}}}{\beta_i \prod_{j=1}^n X_j^{h_{i,j}}} = 1 \tag{17}$$

which is a monomial of the form

$$\frac{\alpha_i}{\beta_i} \prod_{j=1}^n X_j^{g_{i,j} - h_{i,j}} = 1. \tag{18}$$

Monomial equations become linear by taking logarithms on both sides thus reducing the steady-state computation to a linear task:

$$A \cdot y = b \tag{19}$$

where

$$A_{i,j} = g_{i,j} - h_{i,j}$$

$$y_i = \ln X_i$$

$$b_i = \ln \frac{\beta_i}{\alpha_i}$$

Monomial objective functions become linear by taking logarithms and so holds for many constraints on metabolites or fluxes. Therefore, constrained optimization of pathways modeled as S-systems becomes a straightforward linear program [8].

Any other relevant constraint or objective function that is not a power law can also be approximated using the

abovementioned methods. Then logarithms can be taken and Eqns 1–4 can be rewritten as:

$$\max \text{ or } \min \quad F(\mathbf{y})$$

Subject to:

$$A \cdot \mathbf{y} = \mathbf{b} \tag{20}$$

$$B \cdot \mathbf{y} = \mathbf{d} \tag{21}$$

$$C \cdot \mathbf{y} \leq \mathbf{e} \tag{22}$$

$$\mathbf{y}^L \leq \mathbf{y} \leq \mathbf{y}^U \tag{23}$$

Where F is the logarithm of the flux or variable to be optimized, and superscripts L and U refer to lower and upper bounds. Eq. 20 assures operation at steady state. Matrix B and vector \mathbf{d} account for additional equality constraints and C and \mathbf{e} are analogous constraints for additional inequalities, which could, for instance, limit the magnitude of a metabolite concentration or flux, and improve the chances of viability. Optimization problems of this type are called *linear programs* (LPs) and can be solved very efficiently for large numbers of variables and constraints [15].

The advantage of the pure S-system approach is its great speed combined with the fact that S-system models have proven to be excellent representations of many pathways. The disadvantage is that the optimization process, by design, moves the system away from the chosen operating point, so that questions arise as to how accurate the S-system representation is at the steady state suggested by the optimization.

Indirect Optimization Method

If the pathway is not modeled as an S-system, the reduction of the optimization task to linearity is jeopardized. A compromise solution that has turned out to be quite effective is the Indirect Optimization Method (IOM) [10]. The first step of IOM is approximation of the alleged model with an S-system. This S-system is optimized as shown above. The solution is then translated back into the original system in order to confirm that it constitutes a stable steady state and is really an improvement from the basal state of the original model. The S-system solution typically differs somewhat from a direct optimization result with the original model, but since it is obtained so fast, it is possible to execute IOM in several steps with relatively tight bounds, every time choosing a new operating point and not deviating too much from this point in the next iteration [16]. The speed of the process is slower than in the pure S-system case, but still reasonable. Variations on IOM are to search for subsets of independent variables to

be manipulated for optimal yield at lower cost and for multi-objective optimization tasks [17,18].

Global GMA optimization

A global optimization method for GMA systems [19] has been recently proposed based on branch-and-reduce methods combined with convexification. These methods are interesting because of the variety of roles that GMA models can play (see above). The disadvantage of the global method is that it quickly leads to very large systems that are non-convex, even though they allow relatively efficient solutions.

Geometric programming

Geometric programming (GP) [20] addresses a class of problems that include linear programming (LP) and other tasks within the broader category of convex optimization problems. Convex problems are among the few nonlinear tasks where, thanks to powerful interior point methods, the efficient determination of global optima is feasible even for large scale systems. For example, a geometric program of 1,000 variables and 10,000 constraints can be solved in less than a minute on a desktop computer [21]; the solution is even faster for sparse problems as they are found in metabolic engineering. Furthermore, easy to use solvers are starting to become available [22,23].

GP addresses optimization programs where the objective function and the constraints are sums of monomials, i.e., power-law terms as shown in Eq. 6. Because of their importance in GP, sums of monomials, all with positive sign, are called *posynomials*. If some of the monomials enter the sum with negative signs, the collection is called a *signomial*. The peculiarities of convexity and GP methods render the difference between posynomials and signomials crucial.

A GP problem has the generic form:

$$\min \quad P_0(\mathbf{x}) \tag{24}$$

Subject to:

$$P_i(\mathbf{x}) \leq 1 \quad i = 1 \dots n \tag{25}$$

$$M_i(\mathbf{x}) = 1 \quad i = 1 \dots p \tag{26}$$

where $P_i(\mathbf{x})$ and $M_i(\mathbf{x})$ must fulfill strict conditions. Every function $M_i(\mathbf{x})$ must be a monomial, while the objective function $P_0(\mathbf{x})$ and the functions $P_i(\mathbf{x})$ involved in inequalities must be posynomials. Signomials are not permitted, and optimization problems involving them require additional effort.

The equivalence between monomials and power laws immediately suggests the potential use of GP for optimization problems formulated within BST. In the next sections, several methods will be proposed to develop such potential.

Results and discussion

It is easy to see that steady-state equations of S-systems are readily arranged as monomials as shown in Eq 18 and that optimization tasks for S-systems directly adhere to the format of a GP, except that GP mandates minimization. However, this is easily remedied for maximization tasks by minimizing the inverse of the objective, which again is a monomial. By contrast, steady-state GMA equations as shown in Eq. 10 do not automatically fall within the GP structure, because GMA systems usually include negative terms, thus making them signomials. Furthermore, inversion of an objective that contains more than one monomial is not equivalent to a monomial.

When the objective or some restriction falls outside the GMA formalism, it can be recast into proper form as has been discussed above and will be shown in one of the case studies.

Two strategies

The proposed solutions for adapting GP solvers to treat GMA systems rely on *condensation* [24], but they do it in different ways. Condensation is a standard procedure in GP which is exactly equivalent to aggregation in BST. Namely, the sum of monomials is approximated by a single monomial. In the terminology of GP, the condensation $\hat{C}()$ is generically denoted as

$$\hat{C}(P(\mathbf{x})) = \hat{C}(M_1(\mathbf{x}) + \dots + M_n(\mathbf{x})) = M_0(\mathbf{x}) \quad (28)$$

and, in the terminology of Eqs. 10 and 11, defined as:

$$\hat{C}\left(\sum_{j=1}^k n_{i,j} \gamma_j \prod_{k=1}^{n+m} X_k^{f_{j,k}}\right) = \alpha_i \prod_{j=1}^n X_j^{g_{i,j}} \quad (29)$$

where α_i and $g_{i,j}$ are chosen such that equality holds at a chosen operating point; thus, the result is equivalent to the Taylor linearization that is fundamental in BST as was shown in eqn. 7 [5,7,12]. As in the Taylor series, the condensed form is equal to the original equation at the operating point. For any other point, as it can be shown that the left and right hand side of eqn. 29 are equivalent to those of the Arithmetic-Geometric inequality:

$$\sum_{i=1}^n a_i \geq \prod_{i=1}^n \left(\frac{a_i}{w_i}\right)^{w_i} \quad (30)$$

and therefore, the condensed form is an understimation of the original.

Objective functions can only be minimized in GP, this is seldom a problem given that the functions to maximize are often monomials that can be inverted: a variable, a reaction rate or a flux ratio. Posynomial objectives are usually entitled for minimization, like the sum of certain variables. Nonetheless, it is also relevant in metabolic engineering to consider the maximization of posynomials, such as the sum of variables or fluxes. In such cases, condensation or recasting can be used. For an extensive introduction on GP modelling see [25].

A local approach: Controlled Error Method

The steady-state equation of a GMA system may be written as the single difference of two posynomials:

$$P(\mathbf{x}) - Q(\mathbf{x}) = 0 \quad (31)$$

If both posynomials are condensed, every equation will be reduced to the standard form for monomial equations:

$$\frac{\hat{C}(P(\mathbf{x}))}{\hat{C}(Q(\mathbf{x}))} = 1 \quad (32)$$

Because the division of a monomial by another is itself a monomial.

Since the steady state equations of the GMA have been condensed to those of an s-system, this method could be regarded as a direct generalization of classical IOM methods. One of the advantages of this approach is the possibility of keeping posynomial inequalities and objectives as they are and therefore reduce the amount of condensation (approximation) needed, but there is another interesting possibility. When a posynomial is approximated by condensation, the A-G inequality, Eq. 30, guarantees that the monomial is an understimation of the constraint. Furthermore, the posynomial structure is not altered when divided by a monomial so the quotient between a posynomial and its condensed form is always greater than or equal to 1 and provides the exact error as a posynomial function. Therefore the problem can be constrained to allow a maximum error per condensed constraint:

$$\frac{\sum_j \delta_j \prod_k X_k^{b_{j,k}}}{C\left(\sum_j \delta_j \prod_k X_k^{b_{j,k}}\right)} \leq 1 + \varepsilon \quad (33)$$

So the original problem is solved as a series of GPs in which the GMA equations are successively condensed using the previous solution as the reference point. To assure validity an extra set of constraints is added to

ensure that every iteration will only explore the neighborhood of the feasible region in which error due to condensation remains below an arbitrary tolerance set by the user.

A global approach: Penalty Treatment

A similar yet distinct strategy that minimizes the use of condensation is an extension of the penalty treatment method [26], a classic algorithm for signomial programming. In this method, a signomial constraint such as

$$P(\mathbf{x}) - Q(\mathbf{x}) = 0 \tag{34}$$

where P and Q are posynomials, is replaced by two posynomial equalities through the creation of an ancillary variable t:

$$\begin{aligned} P(\mathbf{x}) &= t \\ Q(\mathbf{x}) &= t \end{aligned} \tag{35}$$

These are not valid GP constraints, so the following relaxed version is used:

$$\begin{aligned} P(\mathbf{x}) &\leq t \\ Q(\mathbf{x}) &\leq t \end{aligned} \tag{36}$$

Upon dividing by t, the feasible area of the original problem is contained in the feasible area of the new relaxed version and approximation by condensation is not needed. In order to force these inequalities to be tight in the final solution, the objective function is augmented with penalty terms that grow with the slackness of the constraints, namely the inverses of the condensation of the relaxed constraints. The result of this procedure is a legal GP:

$$\begin{aligned} \min \quad & P_0(\mathbf{x}) + \sum \left(w_i^+ \frac{t}{c[P_i(\mathbf{x})]} + w_i^- \frac{t}{c[Q_i(\mathbf{x})]} \right) \\ \text{subject to:} \quad & \frac{P_i(\mathbf{x})}{t} \leq 1 \\ & \frac{Q_i(\mathbf{x})}{t} \leq 1 \quad i = 0 \dots n \end{aligned} \tag{37}$$

Where the condensed terms are calculated at the basal steady state. If the obtained solution falls within the feasible area of the original problem, it is taken as a solution, if it does not (any of the relaxed inequalities is below 1, the solution is used as the next reference point: condensations are calculated again, the weights of the violated constraints are increased and the new problem is solved. This procedure is repeated until a satisfactory solution is obtained. The original method used 1 as the initial value of the weights and increased them all in every iteration, some modifications are useful for our purposes:

- The initial weights are selected such that the overall penalty terms are just a fraction of the total objective in the initial point. In the case studies explored in this paper, such fraction was 10%.
- The weights are only increased if their corresponding constraint was violated in the last iteration. In such cases, the weight would be multiplied times a fixed value. For the case studies considered here, the choice in the value of such multiplier didn't have a significant impact in the performance of the method.

These variations on the original method serve to prevent the penalty terms from dominating the objective function and pushing the relaxed problem towards the boundaries of the feasible region from the very beginning.

Case studies

In order to illustrate the combination of GP with BST, some optimization tasks were explored. The first example demonstrates the procedure with a very simple two variable GMA system. The second example is a model of the anaerobic fermentation pathway in *Saccharomyces cerevisiae*. The third example revisits an earlier case study concerned with the tryptophan operon in *E. coli*. These systems were optimized using the Matlab based solver ggpplab [23] running on an ordinary laptop (1.6 GHz Pentium centrino, 512 Mb RAM). Matlab scripts were written in order to perform all the transformations required by the two methods described. For comparison, the models were also optimized using IOM [10] as well as Matlab's optimization toolbox. The function used in this toolbox, fmincon(), is based on an iterative algorithm called *Sequential Quadratic Programming*, which uses the BGFS formula to update the estimated Hessian matrix during every iteration [27,28].

A seemingly simple problem

A very distinctive difference between the alternative methods for GMA optimization can be illustrated by a problem modified from [24], which presents the simplest possible fragmented feasible region (see Fig. 1).

$$\begin{aligned} \min \quad & X_1 \\ \text{subject to:} \quad & \frac{1}{4} X_1 + \frac{1}{2} X_2 - \frac{1}{16} X_1^2 - \frac{1}{16} X_2^2 - 1 = 0 \\ & \frac{1}{14} X_1^2 + \frac{1}{14} X_2^2 + 1 - \frac{3}{7} X_1 - \frac{3}{7} X_2 = 0 \\ & 1 \leq X_1 \leq 5.5 \\ & 1 \leq X_2 \leq 5.5 \end{aligned} \tag{38}$$

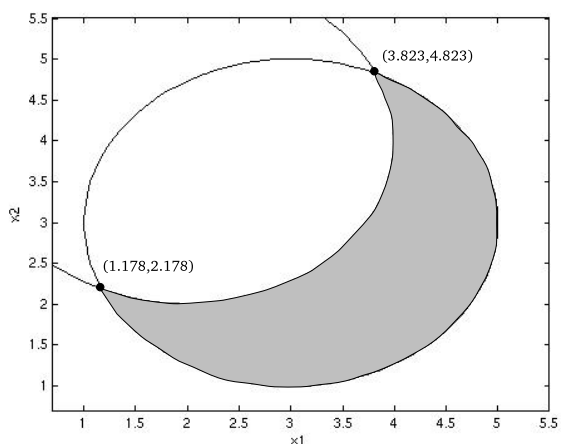


Figure 1
Feasible area of the first example. The lines show the nullclines of each of the two equations of the system. They intersect at two (unconnected) points, which constitute the only feasible solutions. The feasible area of the relaxed problem in the penalty treatment is marked in grey.

The feasible region of this problem consists of two points (1.178,2.178) and (3.823,4.823), of which clearly the first solution is superior, because X_1 is to be minimized. As these points are not connected, local methods are not able to find one solution using the other as a starting point. The problem was solved using IOM, controlled error and penalty treatment methods. The initial point was set to be (3.823,4.823), which is disconnected from the true optimal solution. While both IOM and the Controlled-Error method reported the initial point as the solution, the penalty treatment algorithm found the global optimum at (1.178,2.178).

In this case, most methods failed to find the optimal solution because the approximated s-system had the operating point as the only feasible solution while the relaxed problem for the penalty treatment algorithm had a feasible area (shaded in Fig. 1) that included and connected both feasible solutions.

Anaerobic fermentation in S. cerevisiae

This GMA model [29] (see also appendix) is derived from a previous version [30] formulated with traditional Michaelis Mentem kinetics to explain experimental data, and has been used to illustrate other optimization methods [10,17,19]. It has the following structure (see Fig. 2):

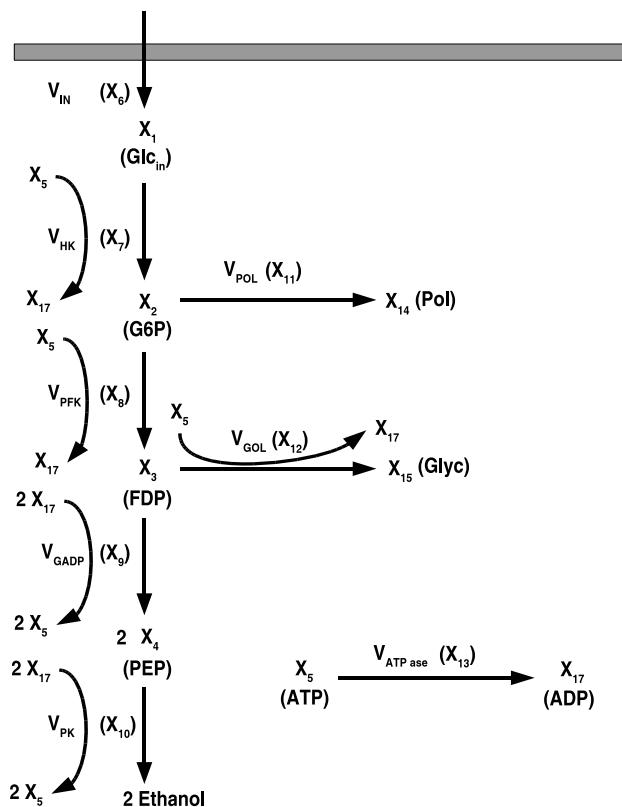


Figure 2
Anaerobic fermentation in *S. cerevisiae*.

$$\begin{aligned}
 \dot{X}_1 &= v_{in} - v_{HK} \\
 \dot{X}_2 &= v_{HK} - v_{PFK} = v_{POL} \\
 \dot{X}_3 &= v_{PFK} - v_{GAPD} - \frac{1}{2} v_{GOL} \\
 \dot{X}_4 &= 2 \cdot v_{GAPD} - v_{PK} \\
 \dot{X}_5 &= 2 \cdot v_{GAPD} + v_{PK} - v_{HK} - v_{PFK} - v_{POL} - v_{ATP}
 \end{aligned}
 \tag{39}$$

The model was already formulated [29] as a GMA system, so that all its fluxes are monomials:

$$\begin{aligned}
 v_{in} &= 0.8122 X_2^{-0.2344} X_6 \\
 v_{HK} &= 2.8632 X_1^{0.7464} X_5^{0.0243} X_7 \\
 v_{PFK} &= 0.5232 X_2^{0.7318} X_5^{-0.3941} X_8 \\
 v_{GAPD} &= 0.011 X_3^{0.6159} X_4^{0.1308} X_9 X_{14}^{-0.6088} \\
 v_{PK} &= 0.0945 X_3^{0.05} X_4^{0.533} X_5^{-0.0822} X_{10} \\
 v_{POL} &= 0.0009 X_2^{8.6107} X_{11} \\
 v_{GOL} &= 0.0945 X_3^{0.05} X_4^{0.533} X_5^{-0.0822} X_{12} \\
 v_{ATP} &= X_5 X_{13}
 \end{aligned}
 \tag{40}$$

The objective is (constrained) maximization of the ethanol production rate, v_{PK} . Together with the upper and lower bounds of the variables, two extra constraints will be studied. The first is an upper limit to the total amount of protein. This is especially important for pathways of the central carbon metabolism as they represent a significant fraction of the total amount of cell protein and increasing the expression of its enzymes by large amounts might compromise cell viability. As a first example, we assume that the activity to protein ratio is the same for every enzyme and set an arbitrary limit of four times the amount of enzymes in the basal state. As an alternative, we explore the effect of limiting the total substrate pool. This constraint will later be subject to tradeoff analysis in order to see its influence in the optimum steady state (see Fig 3). Being posynomial functions, the constraints will be supported by GP without any transformation. The Appendix contains a complete formulation of the optimization problem.

The results are summarized in Table 1. Both GP methods and the SQP found the same solution, although GP finished in 0.5 s while SQP was significantly slower, taking 1.5 s for the calculation. The IOM method was as fast as GP but its solution violated one constraint.

Tryptophan operon

The third example addresses the tryptophan operon in *E. coli*, as illustrated in Fig. 4. This is an appealing benchmark system, because it has already been optimized with other methods [16,31].

A model of the system was recently presented by [32] and includes transcription, translation, chemical reactions and tryptophan consumption for growth. It is thus more than a simple pathway model and demonstrates that GP and BST are applicable in more complex contexts. Finally, this model doesn't follow the structure of any standard formalism so it will be a good example on how recasting wid-

Table 1: Optimization results for the GMA glycolytic model in *S. cerevisiae*. Constraint violations are shown in boldface. GP column stands for both methods

variable	basal	IOM	GP & SQP (times basal)
X_1	0.03456	2.1946	2.0000
X_2	1.0110	1.5801	2.0000
X_3	9.1876	1.5294	2.0000
X_4	0.009532	1.1936	2.0000
X_5	1.1278	0.2803	0.5000
X_6	19.7	7.4873	7.3343
X_7	68.5	3.8583	3.7794
X_8	31.7	2.9176	2.8577
X_9	49.9	6.4799	4.7179
X_{10}	3440	5.7195	4.1642
X_{11}	14.31	0.0100	0.0100
X_{12}	203	0.0100	0.0100
X_{13}	25.1	27.0452	14.0396
X_{14}	0.042	1.0000	1.0000
Flux	30.2231	214.6250	198.8542

ens the applicability of the method to a higher degree of generality. The model takes the form

$$\begin{aligned}
 \dot{X}_1 &= v_1 - v_2 \\
 \dot{X}_2 &= v_3 - v_4 \\
 \dot{X}_3 &= v_5 - v_6 - v_7 - v_8
 \end{aligned}
 \tag{41}$$

Here X_1 , X_2 and X_3 are dimensionless quantities representing mRNA, enzyme levels and the tryptophan concentration, respectively. The rate equations are:

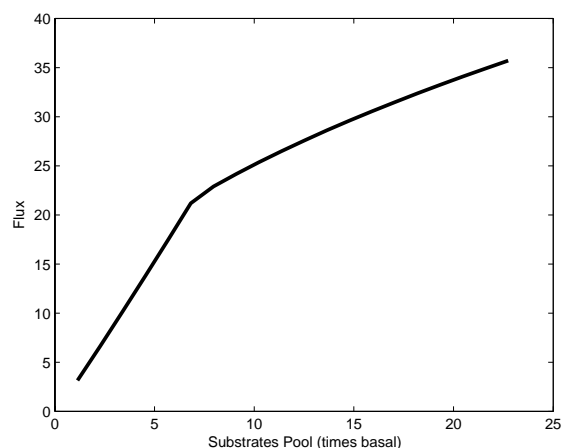


Figure 3 Tradeoff curve for the anaerobic fermentation pathway if the total substrate pools are kept fixed. No upper limit for total enzyme was used in this case.

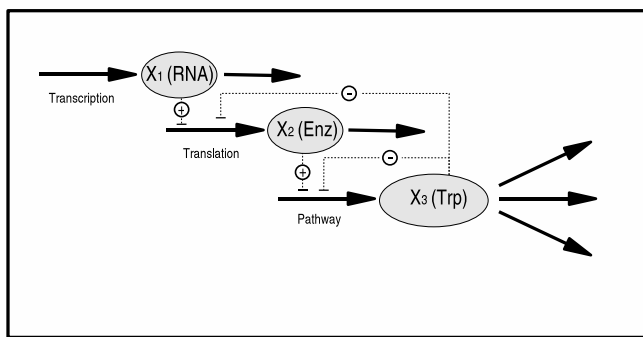


Figure 4
A model of the tryptophan operon. Adapted from [32].

$$\begin{aligned}
 v_1 &= \frac{X_3 + 1}{1 + (1 + X_5)X_3} \\
 v_2 &= (0.9 + X_4)X_1 \\
 v_3 &= X_1 \\
 v_4 &= (0.02 + X_4)X_2 \\
 v_5 &= \frac{X_2 X_6^2}{X_6^2 + X_3^2} \\
 v_6 &= X_3 X_4 \\
 v_7 &= \frac{0.0022 X_3 X_5}{1 + X_3} \\
 v_8 &= \frac{(1 - 7.5 X_4) X_4 X_3 X_7}{X_3 + 0.005}
 \end{aligned} \tag{42}$$

The GMA format is obtained by defining the following ancillary variables:

$$\begin{aligned}
 X_8 &= 1 + X_5 \\
 X_9 &= X_3 + 1 \\
 X_{10} &= 1 + X_8 X_3 \\
 X_{11} &= 0.9 + X_4 \\
 X_{12} &= 0.02 + X_4 \\
 X_{13} &= X_6^2 + X_3^2 \\
 X_{14} &= X_3 + 0.005 \\
 X_{15} &= 1 - 7.5 X_4
 \end{aligned} \tag{43}$$

which turns the rates into power laws:

$$\begin{aligned}
 v_1 &= X_9 X_{10}^{-1} \\
 v_2 &= X_{11} X_1 \\
 v_3 &= X_1 \\
 v_4 &= X_{12} X_2 \\
 v_5 &= X_2 X_6^2 X_{13}^{-1} \\
 v_6 &= X_3 X_4 \\
 v_7 &= 0.0022 X_3 X_5 X_9^{-1} \\
 v_8 &= X_{15} X_3 X_7 X_{14}^{-1}
 \end{aligned} \tag{44}$$

The objective function consists simply of v_8 , which may be regarded as an aggregate term for growth and tryptophan excretion.

A recurrent feature of previously found IOM solutions was the noticeable violation of a constraint retaining a minimum tryptophan concentration. This discrepancy is a feature for comparisons between methods beyond computational efficiency. The Appendix contains a complete formulation of the optimization problem.

In order to test the effectiveness of the controlled error approach, two variants were used in this model:

- Fixed tolerance. The standard method in which every iteration is limited to a maximum condensation error of 10% by constraints described in Eq. 33.
- Fixed step. No limit on the condensation error. The variation of the variables in every iteration is limited to 10% distance from the reference state.

When the constraints were absent (fixed step), the variation of the variables was restricted to a fraction of the total range in every iteration, in order to prevent them from moving too far from the operating point. Fig. 5 shows the evolution of the objective function and condensation errors through iterations, both for fixed step and fixed tolerance. Though both methods find the same solution, the fixed tolerance method is much faster and keeps the error within a limit specified *a priori*. The fixed step method remains within a lower margin of error in this case due to the good quality of the condensed approximation but this margin is not under direct control and will depend on the size of the subintervals and on the model in an unforeseeable way. When the error tolerance was lowered to match the values observed for the fixed step method, both performed very similarly with a slight advantage of the fixed tolerance.

Both the controlled error and penalty treatment methods yielded the same results while SQP returned a solution

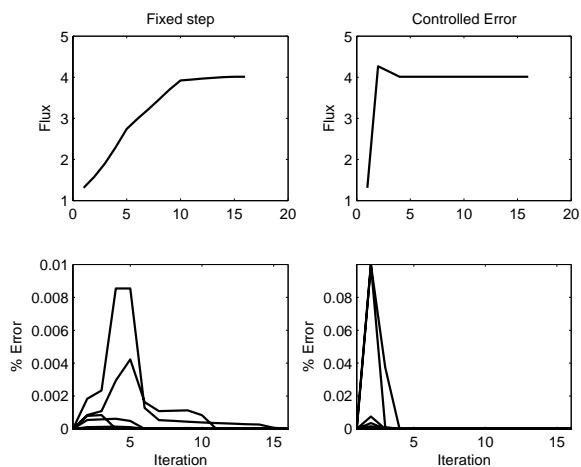


Figure 5
Effect of the error constraints in the optimization algorithm. Results of optimizing the model of the tryptophan operon using fixed step and fixed tolerance.

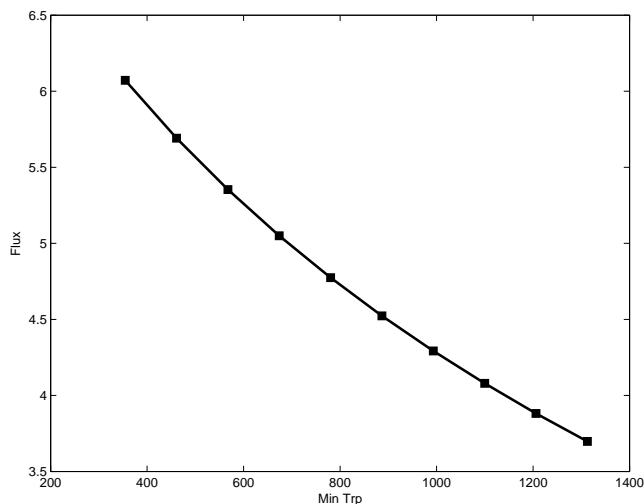


Figure 6
Tradeoff analysis for tryptophan model showing flux against lower bound for tryptophan.

that was feasible but yielded a lower flux. As can be seen in Table 2 no constraint violations occurred with GP. When the lower bound was extended to include the levels reached by other methods, all previous results were reproduced. The tradeoff curve resulting from solving the problem for different tryptophan lower bounds is depicted as Fig 6. SQP and error controlled method took about 1 s to find the solution while the penalty treatment took 0.3 s.

Conclusion

The main challenge of non-linear optimization is dealing with non-convexities. In some cases, like GP, there is an elegant transformation that convexifies the problem without adding undue complexity. But this is seldom the case and dealing with non-convexities usually implies developing *ad hoc* tricks such as subdividing the system in many subsystems, finding convex relaxations of the constraints,

adding extra variables or a combination of several of these strategies.

Geometric programming provides a simple and efficient tool for the optimization of biotechnological systems that takes advantage of the structural regularity and flexibility of GMA systems. In this work we have presented two different strategies to do so, of which the penalty treatment seems to be the most promising. The methods are quite general, as this treatment of GP and recasting can be applied to any rational function, which in fact include almost all rate functions used in representations of metabolic processes.

The use of geometric programming also provides a solution for the problem of constraint violations in the two strategies considered. The possibility of keeping an arbitrary

Table 2: Comparison of results obtained for the tryptophan model with different methods. All the results that violate the lower bound for X₃ were reproduced with GP by relaxing such bound. Constraint violations are shown in boldface.

	basal	IOM	iterative IOM	Modified IOM	GP	SQP
X ₁	0.18465	1.198 X ₁ ₀	1.198 X ₁ ₀	1.198 X ₁ ₀	1.199 X ₁ ₀	1.2 X ₁ ₀
X ₂	7.9868	1.071 X ₂ ₀	1.095 X ₂ ₀	1.055 X ₂ ₀	1.148 X ₂ ₀	1.180 X ₂ ₀
X ₃	1418	0.347 X ₃ ₀	0.465 X ₃ ₀	0.273 X ₃ ₀	0.8 X ₃ ₀	0.825 X ₃ ₀
X ₄	0.00312	0.0058	0.0053	0.062	0.00414	0.0035
X ₅	5	4	4	4	4	4
X ₆	2283	5000	5000	5000	5000	2384
X ₇	430	1000	1000	1000	1000	1000
V _{trp}	1.310	4.26 V _{trp} ₀	3.884 V _{trp} ₀	4.54 V _{trp} ₀	3.062 V _{trp} ₀	2.61 V _{trp} ₀

trarily small approximation error in every iteration prevents the buildup of discrepancies in the Controlled Error Method which results in a "safer" condensation while the Penalty treatment doesn't rely on condensation to define the feasible area. It has been shown elsewhere [21] that GP can deal with big systems, and the sparse nature of the problems in metabolic engineering improves the capabilities of the approach. It is therefore reasonable to expect both strategies considered here to scale well for big problems but it is yet to be seen which one of the two behaves better in such cases.

Geometric programming is a relatively recent and active area in operations research, which implies that further improvements and refinements for the optimization of GMA systems are to be expected. But even with existing methods, the optimization of this large class of systems, which is further expanded by the technique of recasting, has become feasible for execution of moderately sized tasks even on simple desktop computers.

A Optimization problems

Table 3: A.I Anaerobic fermentation by error controlled method

min	$(0.0945X_3^{0.05} X_4^{0.533} X_5^{-0.0822} X_{10})^{-1}$	
Subject to:		Steady state
	$\frac{0.8122 X_2^{-0.2344} X_6}{2.8632 X_1^{0.7464} X_5^{0.0243} X_7} = 1$	
	$\frac{2.8632 X_1^{0.7464} X_5^{0.0243} X_7}{C(0.5232 X_2^{0.7318} X_5^{-0.3941} X_8 + 0.0009 X_2^{8.6107} X_{11})} = 1$	
	$\frac{0.5232 X_2^{0.7318} X_5^{-0.3941} X_8}{C(0.011 X_3^{0.6159} X_5^{0.1308} X_9 X_{14}^{-0.6088} + \frac{1}{2} 0.0945 X_3^{0.05} X_4^{0.533} X_5^{-0.0822} X_{12})} = 1$	
	$\frac{2 \cdot 0.011 X_3^{0.6159} X_5^{0.1308} X_9}{0.0945 X_3^{0.05} X_4^{0.533} X_5^{-0.0822} X_{10}} = 1$	
	$\frac{C(2 \cdot 0.011 X_3^{0.6159} X_5^{0.1308} X_9 + 0.0945 X_3^{0.05} X_4^{0.533} X_5^{-0.0822} X_{10})}{C(2.8632 X_1^{0.7464} X_5^{0.0243} X_7 + 0.5232 X_2^{0.7318} X_5^{-0.3941} X_8 + 0.0009 X_2^{8.6107} X_{11} + X_5 X_{13})} = 1$	Error tolerances
	$\frac{0.5232 X_2^{0.7318} X_5^{-0.3941} X_8 + 0.0009 X_2^{8.6107} X_{11}}{C(0.5232 X_2^{0.7318} X_5^{-0.3941} X_8 + 0.0009 X_2^{8.6107} X_{11})} \leq 1 + \epsilon$	
	$\frac{0.011 X_3^{0.6159} X_5^{0.1308} X_9 X_{14}^{-0.6088} + \frac{1}{2} 0.0945 X_3^{0.05} X_4^{0.533} X_5^{-0.0822} X_{12}}{C(0.011 X_3^{0.6159} X_5^{0.1308} X_9 X_{14}^{-0.6088} + \frac{1}{2} 0.0945 X_3^{0.05} X_4^{0.533} X_5^{-0.0822} X_{12})} \leq 1 + \epsilon$	
	$\frac{2 \cdot 0.011 X_3^{0.6159} X_5^{0.1308} X_9 + 0.0945 X_3^{0.05} X_4^{0.533} X_5^{-0.0822} X_{10}}{\hat{C}(2 \cdot 0.011 X_3^{0.6159} X_5^{0.1308} X_9 + 0.0945 X_3^{0.05} X_4^{0.533} X_5^{-0.0822} X_{10})} \leq 1 + \epsilon$	
	$\frac{2.8632 X_1^{0.7464} X_5^{0.0243} X_7 + 0.5232 X_2^{0.7318} X_5^{-0.3941} X_8 + 0.0009 X_2^{8.6107} X_{11} + X_5 X_{13}}{\hat{C}(2.8632 X_1^{0.7464} X_5^{0.0243} X_7 + 0.5232 X_2^{0.7318} X_5^{-0.3941} X_8 + 0.0009 X_2^{8.6107} X_{11} + X_5 X_{13})} \leq 1 + \epsilon$	

Table 4: A.2 Anaerobic fermentation by penalty treatment

min

$$\begin{aligned}
 & (0.0945x_3^{0.05}x_4^{0.533}x_5^{-0.0822}x_{10})^{-1} \\
 & +w_1^+ \frac{t_1}{2.8632x_1^{0.7464}x_5^{0.0243}x_7} \\
 & +w_1^- \frac{t_1}{C(0.5232x_2^{0.7318}x_5^{-0.3941}x_8 + 0.0009x_2^{8.6107}x_{11})} \\
 & +w_2^+ \frac{t_2}{0.5232x_2^{0.7318}x_5^{-0.3941}x_8} \\
 & +w_2^- \frac{t_2}{C(0.011x_3^{0.6159}x_5^{0.1308}x_9x_{14}^{-0.6088} + w_1^+ \frac{1}{2}0.0945x_3^{0.05}x_4^{0.533}x_5^{-0.0822}x_{12})} \\
 & +w_3^+ \frac{t_3}{\hat{C}(2 \cdot 0.011x_3^{0.6159}x_5^{0.1308}x_9 + 0.0945x_3^{0.05}x_4^{0.533}x_5^{-0.0822}x_{10})} \\
 & +w_3^- \frac{t_3}{\hat{C}(2.8632x_1^{0.7464}x_5^{0.0243}x_7 + 0.5232x_2^{0.7318}x_5^{-0.3941}x_8 + 0.0009x_2^{8.6107}x_{11} + x_5x_{13})}
 \end{aligned}$$

Subject to:
Steady state

$$\begin{aligned}
 & \frac{0.8122x_2^{-0.2344}x_6}{2.8632x_1^{0.7464}x_5^{0.0243}x_7} = 1 \\
 & \frac{2.8632x_1^{0.7464}x_5^{0.0243}x_7}{t_1} \leq 1 \\
 & \frac{0.5232x_2^{0.7318}x_5^{-0.3941}x_8 + 0.0009x_2^{8.6107}x_{11}}{t_1} \leq 1 \\
 & \frac{0.5232x_2^{0.7318}x_5^{-0.3941}x_8}{t_2} \leq 1 \\
 & \frac{0.011x_3^{0.6159}x_5^{0.1308}x_9x_{14}^{-0.6088} + \frac{1}{2}0.0945x_3^{0.05}x_4^{0.533}x_5^{-0.0822}x_{12}}{t_2} \leq 1 \\
 & \frac{2 \cdot 0.011x_3^{0.6159}x_5^{0.1308}x_9}{0.0945x_3^{0.05}x_4^{0.533}x_5^{-0.0822}x_{10}} = 1 \\
 & \frac{2 \cdot 0.011x_3^{0.6159}x_5^{0.1308}x_9 + 0.0945x_3^{0.05}x_4^{0.533}x_5^{-0.0822}x_{10}}{t_3} \leq 1 \\
 & \frac{2.8632x_1^{0.7464}x_5^{0.0243}x_7 + 0.5232x_2^{0.7318}x_5^{-0.3941}x_8 + 0.0009x_2^{8.6107}x_{11} + x_5x_{13}}{t_3} \leq 1
 \end{aligned}$$

Table 5: A.3 Tryptophan by error controlled method

min	$(X_{15}X_3X_7X_{14}^{-1})^{-1}$	
Subject to:		Steady state
	$\frac{X_9X_{10}^{-1}}{X_{11}X_1} = 1$	
	$\frac{X}{X_{12}X_2} = 1$	
	$\frac{X_2X_6^2X_{13}^{-1}}{\hat{C}(x_3x_4 + 0.0022X_3X_5X_9^{-1} + X_{15}X_3X_7X_{14}^{-1})} = 1$	Ancilliary variables
	$\hat{C}((1 + x_5)x_8^{-1}) = 1$	
	$\hat{C}((x_3 + 1)x_9^{-1}) = 1$	
	$\hat{C}((1 + x_8x_3)x_{10}^{-1}) = 1$	
	$\hat{C}((0.9 + x_4)x_{11}^{-1}) = 1$	
	$\hat{C}((0.02 + x_4)x_{12}^{-1}) = 1$	
	$\hat{C}((x_6^2 + x_3^2)x_{13}^{-1}) = 1$	
	$\hat{C}((x_3 + 0.005)x_{14}^{-1}) = 1$	
	$\hat{C}((1 - 7.5x_4)x_{15}^{-1}) = 1$	Error tolerances
	$\frac{x_3x_4 + 0.0022X_3X_5X_9^{-1} + X_{15}X_3X_7X_{14}^{-1}}{\hat{C}(x_3x_4 + 0.0022X_3X_5X_9^{-1} + X_{15}X_3X_7X_{14}^{-1})} \leq 1 + \varepsilon$	
	$\frac{1 + x_5}{\hat{C}(1 + x_5)} \leq 1 + \varepsilon$	
	$\frac{x_3 + 1}{\hat{C}(x_3 + 1)} \leq 1 + \varepsilon$	
	$\frac{1 + x_8x_3}{\hat{C}(1 + x_8x_3)} \leq 1 + \varepsilon$	
	$\frac{0.9 + x_4}{\hat{C}(0.9 + x_4)} \leq 1 + \varepsilon$	
	$\frac{0.02 + x_4}{\hat{C}(0.02 + x_4)} \leq 1 + \varepsilon$	
	$\frac{x_6^2 + x_3^2}{\hat{C}(x_6^2 + x_3^2)} \leq 1 + \varepsilon$	
	$\frac{x_3 + 0.005}{\hat{C}(x_3 + 0.005)} \leq 1 + \varepsilon$	
	$\frac{1 - 7.5x_4}{\hat{C}(1 - 7.5x_4)} \leq 1 + \varepsilon$	

Table 6: A.4 Tryptophan penalty approach

<p>min</p> $ \begin{aligned} & (X_{15}X_3X_7X_{14}^{-1})^{-1} \\ & +w_1^+ \frac{t_1}{X_2X_6^2X_{13}^{-1}} + w_1^- \frac{t_1}{C(x3x4 + 0.0022X_3X_5X_9^{-1} + X_{15}X_3X_7X_{14}^{-1})} \\ & +w_2 \frac{X_8}{C(1+x5)} + w_3 \frac{X_9}{C(X_3+1)} + w_4 \frac{X_{10}}{C(1+X_8x3)} \\ & +w_5 \frac{X_{11}}{C(0.9+X_4)} + w_6 \frac{X_{12}}{C(0.02+X_4)} + w_7 \frac{X_{13}}{C(X_6^2+X_3^2)} \\ & +w_8 \frac{X_4}{C(X_3+0.005)} + w_9 \frac{X_{15}}{C(1-7.5X_4)} \end{aligned} $ <p>Subject to:</p>	<p>Steady state</p> $ \begin{aligned} \frac{X_9X_{10}^{-1}}{X_{11}X_1} &= 1 \\ \frac{X_1}{X_{12}X_2} &= 1 \\ \frac{X_2X_6^2X_{13}^{-1}}{t_1} &\leq 1 \\ \frac{x3x4 + 0.0022X_3X_5X_9^{-1} + X_{15}X_3X_7X_{14}^{-1}}{t_1} &\leq 1 \end{aligned} $ <p>Ancilliary variables</p> $ \begin{aligned} (1+x5)X_8^{-1} &\leq 1 \\ (X_3+1)X_9^{-1} &\leq 1 \\ (1+X_8x3)X_{10}^{-1} &\leq 1 \\ (0.9+X_4)X_{11}^{-1} &\leq 1 \\ (0.02+X_4)X_{12}^{-1} &\leq 1 \\ (X_6^2+X_3^2)X_{13}^{-1} &\leq 1 \\ (X_3+0.005)X_{14}^{-1} &\leq 1 \\ (1-7.5X_4)X_{15}^{-1} &\leq 1 \end{aligned} $
---	---

Competing interests

The author(s) declare that they have no competing interests.

Acknowledgements

This work was supported by a research grant from the Spanish Ministry of Science and Education ref. BIO2005-08898-C02-02.

References

- Stephanopoulos G, Aristidou A, Nielsen J: *Metabolic Engineering: Principles and Methodologies* Academic Press; 1998.
- Torres N, Voit E: *Pathway Analysis and Optimization in Metabolic Engineering* Cambridge University Press; 2002.
- Mendes P, Kell D: **Non-linear optimization of biochemical pathways: applications to metabolic engineering and parameter estimation.** *Bioinformatics* 1998, **14(10)**:869-83.
- Varma A, Boesch BV, Palsson BO: **Metabolic flux balancing: Basic concepts, scientific and practical use.** *Bio-Technology* 1994:994-998.
- Savageau M: **Biochemical systems analysis. I. Some mathematical properties of the rate law for the component enzymatic reactions.** *J Theor Biol* 1969, **25(3)**:365-9.
- Savageau M: **Biochemical systems analysis. II. The steady-state solutions for an n-pool system using a power-law approximation.** *J Theor Biol* 1969, **25(3)**:370-9.
- Voit E: *Computational Analysis of Biochemical Systems. A Practical Guide for Biochemists and Molecular Biologists* Cambridge University Press; 2000.

8. Voit E: **Optimization in integrated biochemical systems.** *Biotechnol Bioeng* 1992:572-582.
9. Hatzimanikatis V, Bailey JE: **MCA has more to say.** *J Theor Biol* 1996:233-242.
10. Torres N, Voit E, Glez-Alcon C, Rodriguez F: **An indirect optimization method for biochemical systems. Description of method and application to ethanol, glycerol and carbohydrate production in *Saccharomyces cerevisiae*.** *Biotech Bioeng* 1997, **5(55)**:758-772.
11. Shiraishi F, Savageau MA: **The tricarboxylic acid cycle in *Dictyostelium discoideum*. III. Analysis of steady state and dynamic behavior.** *J Biol Chem* 1992, **267(32)**:22926-22933.
12. Savageau M: *Biochemical Systems Analysis. A Study of Function and Design in Molecular Biology* Addison-Wesley, Reading, Massachusetts; 1976.
13. De Atauri P, Curto R, Puigjaner J, Cornish-Bowden A, Cascante M: **Advantages and disadvantages of aggregating fluxes into synthetic and degradative fluxes when modelling metabolic pathways.** *Eur J Biochem* **265(2)**:671-679.
14. Savageau M, Voit E: **Recasting nonlinear differential equations as S-systems: A canonical nonlinear form.** *Math Biosci* 1987:83-115.
15. Dantzig G: *Linear Programming and Extensions* Princeton University Press, Princeton, New Jersey; 1963.
16. Xu G, Shao C, Xiu Z: **A Modified Iterative IOM Approach for Optimization of Biochemical Systems.** eprint arXiv:q-bio/0508038 2005.
17. Vera J, de Atauri P, Cascante M, Torres N: **Multicriteria optimization of biochemical systems by linear programming: application to production of ethanol by *Saccharomyces cerevisiae*.** *Biotechnol Bioeng* 2003, **83(3)**:335-43.
18. Alvarez-Vasquez F, Gonzalez-Alcon C, Torres N: **Metabolism of citric acid production by *aspergillus niger*: model definition, steady-state analysis and constrained optimization of citric acid production rate.** *Biotechnol Bioeng* 2000, **70**:82-108.
19. Polisetty P, Voit E, Gatzke EP: **Yield Optimization of *Saccharomyces cerevisiae* using a GMA Model and a MILP-based piecewise linear relaxation method.** *Proceedings of: Foundations of Systems Biology in Engineering* 2005.
20. Zener C: *Engineering Design by Geometric Programming* John Wiley and Sons, Inc; 1971.
21. Boyd S, Vandenberghe L: *Convex Optimization* Cambridge University Press; 2004.
22. Grant M, Boyd S, Ye Y: **CVX: Matlab Software for Disciplined Convex Programming.** 2005.
23. Koh K, Kim S, Mutapic A, Boyd S: **GGPLAB: A simple Matlab toolbox for Geometric Programming.** 2006. [Version 0.95]
24. Floudas CA: *Deterministic Global Optimization* Kluwer Academic Publishers; 2000.
25. Boyd S, Kim S, Vandenberghe L, Hassibi : **A tutorial on geometric programming.** . [To be published in Optimization and Engineering]
26. Roundtree D, Rigler A: **A penalty treatment of equality constraints in generalized geometric programming.** *Journal of Optimization Theory and Applications* 1982, **38(2)**:169-178.
27. Goldfarb D: **A Family of Variable Metric Updates Derived by Variational Mean.** *Mathematics of Computing* 1970, **24**:23-26.
28. Fletcher D, Powell M: **A rapidly convergent Descent Method for minimization.** *Computer Journal* 1963, **6**:163-168.
29. Curto R, Sorribas A, Cascante M: **Comparative characterization of the fermentation pathway of *Saccharomyces cerevisiae* using biochemical systems theory and metabolic control analysis: model definition and nomenclature.** *Math Biosci* 1995, **130**:25-50.
30. Galazzo J, Bailey J: **Fermentation pathway kinetics and metabolic flux control in suspended and immobilized *Saccharomyces cerevisiae*.** *Enzyme Microb Technol* 1990:162-172.
31. Marin-Sanguino A, Torres NV: **Optimization of tryptophan production in bacteria. Design of a strategy for genetic manipulation of the tryptophan operon for tryptophan flux maximization.** *Biotechnol Prog* 2000, **16(2)**:133-145.
32. Xiu Z, Chang Z, Zeng A: **Nonlinear dynamics of regulation of bacterial trp operon: model analysis of integrated effects of repression, feedback inhibition, and attenuation.** *Biotechnol Prog* 2002, **18(4)**:686-93.

Publish with **BioMed Central** and every scientist can read your work free of charge

"BioMed Central will be the most significant development for disseminating the results of biomedical research in our lifetime."

Sir Paul Nurse, Cancer Research UK

Your research papers will be:

- available free of charge to the entire biomedical community
- peer reviewed and published immediately upon acceptance
- cited in PubMed and archived on PubMed Central
- yours — you keep the copyright

Submit your manuscript here:
http://www.biomedcentral.com/info/publishing_adv.asp

