Inverse Dynamical Problems: An Algebraic Formulation Via MP Grammars

Vincenzo Manca and Luca Marchetti

University of Verona, Department of Computer Science
Strada Le Grazie 15, 37134 Verona, Italy
vincenzo.manca@univr.it luca.marchetti@univr.it

Summary. Metabolic P grammars are a particular class of multiset rewriting grammars introduced in the MP systems' theory for modelling metabolic processes. In this paper, a new algebraic formulation of inverse dynamical problems, based on MP grammars and Kronecker product, is given, for further motivating the correctness of the LGSS (Log-gain Stoichiometric Stepwise) algorithm, introduced in 2010s for solving dynamical inverse problems in the MP framework. At the end of the paper, a section is included that introduces the problem of multicollinearity, which could arise during the execution of LGSS, and that defines an algorithm, based on a hierarchical clustering technique, that solves it in a suitable way.

Key words: Metabolic P systems, dynamical systems, dynamical inverse problems, Kronecker product, stepwise regression.

1 Introduction

Metabolic P (MP) systems are a particular class of cell-like P systems [33, 34, 36, 35] introduced by Vincenzo Manca in 2004, for modelling metabolic processes [29]. An MP system is essentially a particular type of deterministic discrete dynamical system which inherits from the P systems' framework a native similitude with the functioning of a living cell.

MP systems share with P systems the multiset rewriting mechanism as their fundament. However, while P systems are essentially unconventional computational models, MP systems are intended to generate dynamics instead of computations. Namely, their aim in modelling biological phenomena is that of finding the multiset rewriting mechanism underlying an observed biological behaviour.

Metabolic P systems can be considered as the result of a research activity initiated in 1990s with some initial works [15, 28, 30]. They are different, with respect to other "P variants" applied in the context of systems biology [3, 4, 6, 38, 39]. The main difference is in their determinism. In fact, their basis are *MP grammars*, where multiset transformations are regulated by functions in a *deterministic* way [19]. An MP system is an MP grammar equipped with a *temporal interval* τ, a conventional *mole size* ν, and substances masses, which specify the time and population (discrete) granularities respectively [19].

An MP grammar G can be considered as a generator of time series, determined by the following structure ($n, m \in \mathbb{N}$, the set of natural numbers):

$$G = (M, R, I, \Phi)$$

where:

1. $M = \{x_1, x_2, \ldots, x_n\}$ is a finite set of elements called *metabolites*, or *substances.* A *metabolic state* is given by a list of n values, each of which is associated to a metabolite.
2. $R = \{\alpha_j \to \beta_j \mid j = 1, \ldots, m\}$ is a set of *rules*, or *reactions*, with α_j and β_j multisets over M for $j = 1, \ldots, m$.
3. I are *initial values* of metabolites, that is, a list $x_1[0], x_2[0], \ldots, x_n[0]$ providing the *metabolic state at step 0.*
4. $\Phi = \{\varphi_1, \ldots, \varphi_m\}$ is a list of functions, called *regulators*, one for each rule, such that, for $1 \leq j \leq m$, and for some k_j $(0 \leq k_j \leq n)$

$$\varphi_j : \mathbb{R}^{k_j} \to \mathbb{R}.$$

An MP grammar G is *parametric*, when a set P of parameters is added to G, and metabolic states include also elements of P (to which, the state assigns real values), therefore regulators may include parameters as their arguments . If G is parametric, also the time series of parameters has to be provided in order to specify G.

An MP grammar can be easily representable by an *MP graph* [22]. Moreover, the set of the rules of the system can be also represented by a *stoichiometric matrix* $\mathbb{A}$, which gives a sort of “matrix-like representation” of the stoichiometry (see Figure 1).

An MP grammar G defines, for any $x \in M$, a time series

$$(x[i] \mid i \in \mathbb{N}, i > 0)$$

in the following way. Let

$$s[i] = (x_1[i], x_2[i], \ldots, x_n[i])$$

the (row) *state vector* of G at step i, which can be seen as a function from the set of metabolites to $\mathbb{R}$, then the flux $\varphi_j(s_j[i])$ of rule r_j at step i, is given by applying the regulator φ_j to $s_j[i]$, a substate of $s[i]$ associated to r_j, and constituted by k_j components called the *tuners* of r_j.

If we consider the rule r_2 of the MP grammar given in Figure 1, for example, then the flux at step i is calculated by:

$$\begin{aligned}\varphi_2(s_2[i]) &= \varphi_2(A[i], B[i]) \\ &= c_2 A[i]^2 + c_3 B[i]^3\end{aligned}$$

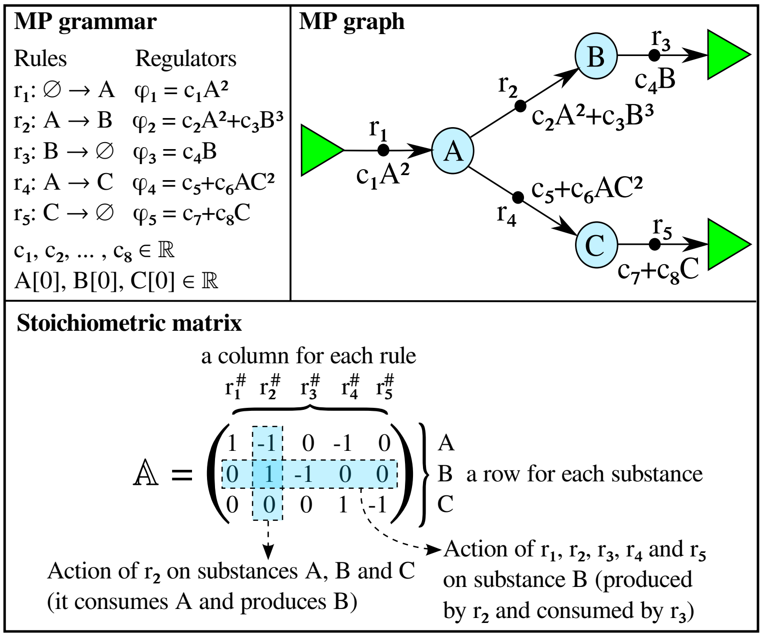

Fig. 1. An example of MP grammar (where $\emptyset$ denotes an empty multiset and substance symbols occurring in regulators denote the corresponding substance quantities), the stoichiometric matrix $\mathbb{A}$ is directly deduced by the MP grammar on the top left corner. The MP graph on the top right corner is obtained by translating the rules in the source-target-edge notation [26].

where c_2, c_3 are given real constants, and A and B are said to be the tuners of the rule r_2.

The value of $x[i+1]$, for each $x \in M$, is given by the following equation, where $\alpha_j(x)$ and $\beta_j(x)$ denote the multiplicities of x in the multiset α_j and β_j, respectively:

$$x[i+1] = x[i] + \sum_{j=1}^{m} [(\beta_j(x) - \alpha_j(x)) \cdot \varphi_j(s_j[i])].$$

More generally, if we denote by $\mathbb{A}$ the stoichiometric matrix of the system and by

$$\Phi[i] = (\varphi_1(s_1[i]), \varphi_2(s_2[i]), \ldots, \varphi_m(s_m[i]))$$

the row vector of fluxes at step i, it can be proved that [16]:

$$(s[i+1] - s[i])^T = \mathbb{A} \times \Phi^T[i] \tag{1}$$

that is, by transposition:

$$s[i+1] - s[i] = \Phi[i] \times \mathbb{A}^T. \tag{2}$$

These last two equations define equivalently the *Equational Metabolic Algorithm (EMA)*. In the following, the MP dynamics we will present are computed in MATLAB[1] by applying EMA. We refer to [19, 20, 18, 21] for a comprehensive presentation of the MP theory.

The dynamics which can be modelled by MP systems can be very complicated even by considering simple MP grammars (i.e. with few substances and linear regulators). In [23] MP systems were successfully applied to the field of real periodical function approximation. The complexity of the dynamics compared to the simplicity of the MP grammar which calculates it by EMA, suggests that MP system theory can be a suitable framework for modelling biological dynamics.

The procedure introduced in [23] to define the models has been widely extended in [25, 26] for defining the *LGSS (Log-Gain Stoichiometric Stepwise)* algorithm, which derives MP grammars generating time series of observed dynamics. LGSS can be applied independently from any knowledge about reaction rate kinetics and it represents the most recent solution, in terms of MP systems, of the *dynamical inverse problem*, that is, of the identification of (discrete) mathematical models of an observed dynamics and satisfying all the constraints required by the specific knowledge about the modelled phenomenon. The LGSS algorithm combines and extends the log-gain principles developed in the MP system theory [16, 17] with the classical method of Stepwise Regression [7], which is a statistical regression technique based on Least Squares Approximation and statistical F-tests [5].

LGSS has been implemented by Luca Marchetti in 2010 as a set of MATLAB functions. We refer to [24, 31, 32, 27] for some successful applications of LGSS and MP systems for discovering the internal regulation logic of phenomena relevant in systems biology.

The starting point of the LGSS algorithm was the search for the right regulators associated to the reactions of an MP grammar which provide the observed time series when dynamics is computed by means of EMA. If we consider the role of each regulator, we realize that it affects the variations of many substances. Therefore regulators are constrained to satisfy altogether, at each step, an algebraic system based on the stoichiometry of the observed phenomenon. The crucial point for regulator determination was a special kind of regression formulated as "stoichiometric expansion" of EMA by means of an initial set of basic functions called regressors.

In the next section we will introduce a new algebraic formulation of the stoichiometric expansion, based on MP grammars and Kronecker product, which better describes and motivates its adoption in LGSS for solving inverse dynamical problems.

[1] See `http://www.mathworks.it/index.html` for details on the MATLAB software.

2 Stoichiometric expansion

Given a system with n variables $x_1, x_2, \ldots, x_n$, let us suppose to know the time series of these variables along time points $0, 1, \ldots, t$. Let

$$s[i] = (x_1[i], x_2[i], \ldots, x_n[i])$$

the (row) state vector at time i, and

$$x_j[i+1] - x_j[i] = \mathit{\Delta}_j[i]$$

for $j = 1, 2, \ldots, n$, then

$$s[i+1] - s[i] = (\mathit{\Delta}_1[i], \mathit{\Delta}_2[i], \ldots, \mathit{\Delta}_n[i])$$

whence, from equation (2), we get

$$\mathit{\Phi}[i] \times \mathbb{A}^T = (\mathit{\Delta}_1[i], \mathit{\Delta}_2[i], \ldots, \mathit{\Delta}_n[i]). \tag{3}$$

For the determination of the regulators which provide the best approximate solution of the system (3), which has m unknowns (the m components of the flux vector $\mathit{\Phi}[i]$), LGSS applies a procedure called *stoichiometric expansion.* Let us assume that the regulators we are searching for can be expressed as linear combinations of some basic regressors $g_1, g_2, \ldots, g_d$ which usually include constants, powers, and products of substances, plus some basic functions which are considered suitable in the specific cases under investigation:

$$\begin{aligned} \varphi_1 &= c_{1,1}g_1 + c_{1,2}g_2 + \ldots + c_{1,d}g_d \\ \varphi_2 &= c_{2,1}g_1 + c_{2,2}g_2 + \ldots + c_{2,d}g_d \\ \ldots &= \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ \varphi_m &= c_{m,1}g_1 + c_{m,2}g_2 + \ldots + c_{m,d}g_d. \end{aligned} \tag{4}$$

Let us consider the *t-expansion* $\mathit{\Phi}_1^t, \mathit{\Phi}_2^t, \ldots, \mathit{\Phi}_m^t$ of regulators as the vectors constituted by the right members of equations (4) evaluated along t steps (where the values of all the variables of the system are supposed to be known):

$$\begin{aligned} \mathit{\Phi}_1^t &= c_{1,1}G_1^t + c_{1,2}G_2^t + \ldots + c_{1,d}G_d^t \\ \mathit{\Phi}_2^t &= c_{2,1}G_1^t + c_{2,2}G_2^t + \ldots + c_{2,d}G_d^t \\ \ldots &= \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ \mathit{\Phi}_m^t &= c_{m,1}G_1^t + c_{m,2}G_2^t + \ldots + c_{m,d}G_d^t. \end{aligned} \tag{5}$$

Now, let $C_1^d, C_2^d, \ldots, C_m^d$ be the unknown column vectors, of dimension d, constituted by the coefficients of the regressors providing the linear combinations of regulators $\varphi_1, \varphi_2, \ldots, \varphi_m$ we are searching for, and

$$\mathbb{C} = (C_1^d, C_2^d, \ldots, C_m^d)$$

the matrix having these vectors as columns. Moreover, let $\mathit{\Delta}_1^t, \mathit{\Delta}_2^t, \ldots, \mathit{\Delta}_n^t$ be the column vectors of dimension t constituted by substance variations of substances, from step i to step $i+1$, for $0 \leq i \leq t-1$, and

$$\mathit{\Delta} = (\mathit{\Delta}_1^t, \mathit{\Delta}_2^t, \ldots, \mathit{\Delta}_n^t)$$

the matrix having these vectors as columns. Let also $\mathit{\Phi}^t$ be the following matrix constituted by m column vectors of t elements:

$$\mathit{\Phi}^t = (\mathit{\Phi}_1^t, \mathit{\Phi}_2^t, \ldots, \mathit{\Phi}_m^t).$$

Finally, let

$$\mathbb{G} = (G_1^t, G_2^t, \ldots, G_d^t)$$

the matrix, of dimension $t \times d$, having as columns the vectors obtained by evaluating the regressors $g_1, g_2, \ldots, g_d$ on the t observed time points. With the notation above, the system of equations (5) becomes:

$$\mathbb{G} \times \mathbb{C} = \mathit{\Phi}^t. \tag{6}$$

Now, it easily follows from (3) that:

$$\mathit{\Phi}^t \times \mathbb{A}^T = \mathit{\Delta} \tag{7}$$

where the exponent T denotes the matrix transposition. Therefore, by combining equations (6) and (7), we finally obtain the *t-expansion* of the system (3) as:

$$\mathbb{G} \times \mathbb{C} \times \mathbb{A}^T = \mathit{\Delta}. \tag{8}$$

The coefficients of $\mathbb{C}$ are the unknowns which needs to be estimated by LGSS. We show now that they can be obtained by a Least Square Estimation deduced by equation (8), by using *direct product* $\otimes$ between matrices, also called *Kronecker product* [9, 10, 40, 41], which results a special case of *tensor product* used in linear algebra and in mathematical physics.

Given two real matrix A, B of dimension $n \times m$ and $t \times d$ respectively, then the *direct product*:

$$A \otimes B$$

is the matrix, of dimension $nt \times md$, constituted by nm blocks $B_{i,j}$, such that, if $A = (a_{i,j} \mid 1 \leq i \leq n,\ 1 \leq j \leq m)$, then $B_{i,j} = a_{i,j}B$ (in $B_{i,j}$ all the elements of B are multiplied by $a_{i,j}$, see Figure 2).

The Kronecker product is bilinear and associative, that is, it satisfies the following equations:

$$\begin{aligned} A \otimes (B + C) &= (A \otimes B) + (A \otimes C) \\ (A + B) \otimes C &= (A \otimes B) + (A \otimes C) \\ (kA) \otimes B &= A \otimes (kB) = k(A \otimes B) \\ (A \otimes B) \otimes C &= A \otimes (B \otimes C). \end{aligned}$$

$$\begin{pmatrix} a & b & c \\ d & e & f \end{pmatrix} \otimes \begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix} = \begin{pmatrix} a\begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix} & b\begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix} & c\begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix} \\ \\ d\begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix} & e\begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix} & f\begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix} \end{pmatrix}$$

Fig. 2. An example of Kronecker product of two matrices.

Moreover, matrix direct product verifies also the following equations:

$$\begin{aligned} (A \otimes B) \times (C \otimes D) &= (A \times C) \otimes (B \otimes D) \\ (A \otimes B)^T &= A^T \otimes B^T \\ (A \otimes B)^{-1} &= A^{-1} \otimes B^{-1} \end{aligned}$$

where the exponent T denotes transposition and the last equation holds only when the involved matrices are invertible.

Let us denote by $vec(W)$ the *vectorization* of the matrix W, obtained by concatenating in a unique column vector all the columns of W in their order. Then, a general property of matrix direct product asserts that [10]:

$$A \times X \times B = Y \quad iff \quad (B^T \otimes A) \times vec(X) = vec(Y). \tag{9}$$

Therefore, if we apply equivalence (9) to equation (8) we obtain:

$$(\mathbb{A} \otimes \mathbb{G}) \times vec(\mathbb{C}) = vec(\varDelta) \tag{10}$$

where the *stoichiometric matrix* is multiplied, by Kronecker product, with the *regressor matrix* and the result is multiplied with the vectorization of the *regressor coefficient matrix*, and then equated to the vectorization of the *substance variation matrix*, by providing nt equations with md unknown values. The system of equations given in (10) is the *stoichiometric expanded system* calculated by LGSS.

According to the Least Square approximation method [43, 13], if $nt \geq md$, then the best approximation to $vec(C)$, minimizing the difference between the two members of equation (10), is given by the following vector:

$$\big((\mathbb{A} \otimes \mathbb{G})^T \times (\mathbb{A} \otimes \mathbb{G})\big)^{-1} \times (\mathbb{A} \otimes \mathbb{G})^T \times vec(\varDelta). \tag{11}$$

Some constraints may be imposed to the fluxes provided by regulators, which may be of general nature, or may be specific to some classes of systems (for example, fluxes should not be negative, and the sum of fluxes of all reactions consuming a substance x cannot exceed the quantity of x). In Figure 3 are represented the regressor matrix $\mathbb{G}$ and the substance variation matrix $\varDelta$ which are used by LGSS for least-squares approximating the coefficients $c_1, \ldots, c_8$ of the MP grammar given in Figure 1.

$$\textbf{Regressor matrix} \qquad \qquad \textbf{Substance variation matrix}$$

$$\mathbb{G} = \left(\begin{array}{cccccc} 1 & (A[0])^2 & B[0] & (B[0])^3 & C[0] & A[0]\cdot(C[0])^2 \\ 1 & (A[1])^2 & B[1] & (B[1])^3 & C[1] & A[1]\cdot(C[1])^2 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & (A[t-1])^2 & B[t-1] & (B[t-1])^3 & C[t-1] & A[t-1]\cdot(C[t-1])^2 \end{array}\right) \quad \Delta = \left(\begin{array}{ccc} A[1]-A[0] & B[1]-B[0] & C[1]-C[0] \\ A[2]-A[1] & B[2]-B[1] & C[2]-C[1] \\ \dots & \dots & \dots \\ A[t]-A[t-1] & B[t]-B[t-1] & C[t]-C[t-1] \end{array}\right)$$

$$\underbrace{\qquad}_{[1]^t}\ \underbrace{\qquad}_{[A^2]^t}\ \underbrace{\qquad}_{[B]^t}\ \underbrace{\qquad}_{[B^3]^t}\ \underbrace{\qquad}_{[C]^t}\ \underbrace{\qquad\quad}_{[AC^2]^t} \qquad \underbrace{\qquad\quad}_{[\Delta_A]^t}\ \underbrace{\qquad\quad}_{[\Delta_B]^t}\ \underbrace{\qquad\quad}_{[\Delta_C]^t}$$

Fig. 3. The regressor matrix $\mathbb{G}$ and the substance variation matrix $\varDelta$ used for approximating the coefficients $c_1, c_2, \dots, c_8$ of the MP grammar given in Figure 1.

However, the approximation given by (11) cannot in general be considered the best way for solving the inverse dynamical problem. In fact, apart the computational cost of considering all the d regressors at same time, several reasons suggest to follow a gradual strategy in the determination of a subset of regressors and their corresponding coefficient which provide the best approximation to the given dynamics. There are two main requirements which are essential for an appropriate application of least squares method: the linear independence among the regressor expansions and the parsimony of the set of regressors. In other words, the best approximation is obtained by determining a *parsimonious* set of linearly independent regressors ensuring an error under a given threshold.

Linear independence is a requirement of least squares method and is solved by considering systems of equations which have been stoichiometric expanded. The parsimony of the model, instead, avoids problems of *overfitting*. In fact, the more regressors are considered in the model, the less is the degree of freedom left for the error [1]. This implies that solution fits very well with the dynamics on the observation points, but it is too constrained to them for behaving in a satisfactory way outside them (i.e. the model fits well the data, but it has not predictive power, see Figure 4 for an example).

In order to cope with the requirements explained above, LGSS integrates the least squares approximation of stoichiometric expanded systems with a regression strategy based on a step-wise approach as defined in [26]. Such kind of approach permits to define the model, step by step, by inserting into the model only those expanded regressors (among the columns of the matrix given by the direct product $\mathbb{A}\otimes\mathbb{G}$) which satisfy specific statistical tests. In this way, we can obtain MP models which fit the dynamics and that comprehends a small set of regressors.

3 Problems related to the regression in LGSS

The stepwise approach adopted in LGSS is based on the assumptions which are at the basis of the classical multiple regression model [1]. These assumptions concern with some properties of the expanded regressors (i.e. they must be linearly independent and, possibly, not correlated[2]) and with the probability distribution of the errors associated to observations in considered time series (i.e. the errors should be normally distributed with mean zero). When one or more of these assumptions are not completely satisfied, some mistakes can occur in the definition of the regulators. In particular, there are several problems which we need to be aware of in the context of multiple regression. Some of them have been discussed in [26] and can be solved by substituting the ordinary least squares with other estimation methods based on the *weighted least squares* [42] or on the *generalized least squares* [12].

Here we focus on solving the problem of multicollinearity, which consists in having regressors that are highly correlated among them. This is the most common problem occurring in LGSS and also one of the most difficult to be solved [1]. When we develop a new MP model, we hope to have a strong correlation between each expanded regressor and the dependent variable $vec(\mathit{\Delta})$, but we do not want to have expanded regressors correlated among them. In fact, this phenomenon may cause errors in the selection of the right set of regressors during the execution of the stepwise regression. In the case of perfect collinearity, the regression algorithm breaks down completely (because the matrix given by the direct product $\mathbb{A} \otimes \mathbb{G}$ has not maximum rank). Since in LGSS usually regulators are assumed to be linear combinations of polynomial regressors, then it is very common to meet multicollinearity problems.

[2] The correlation between regressors is intended to be calculated by means of the *Pearson's correlation coefficient* [37], which ranges from −1 to 1 and provides a measure of dependence between the behaviours of two magnitudes (−1: perfect anti-correlation; 0: no correlation; 1: perfect correlation).

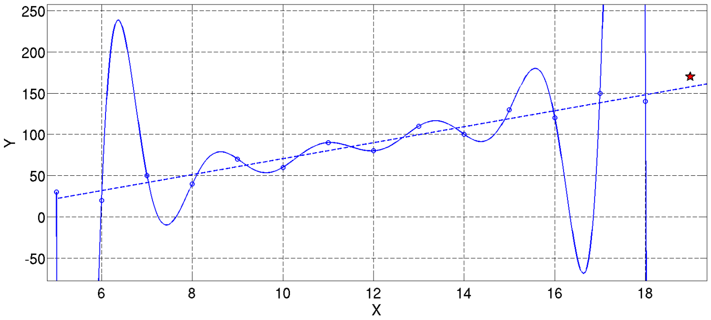

Fig. 4. Comparison between the predictive power of two regression models: a 13-degree polynomial $\widehat{Y} = c_0 + c_1X + c_2X^2 + \ldots + c_{13}X^{13}$ (depicted by the continuous line) and a least squares line (depicted by the dotted line). The dataset used to calculate the models are the 14 points depicted as blue circles, the last point represented by the red star is the value of Y we want to predict with our models. The 13-degree polynomial is a perfect example of model which overfits the data: in fact, it provides a perfect fit for all the points of the dataset, but it completely fails the prediction of Y in the 15th data point.

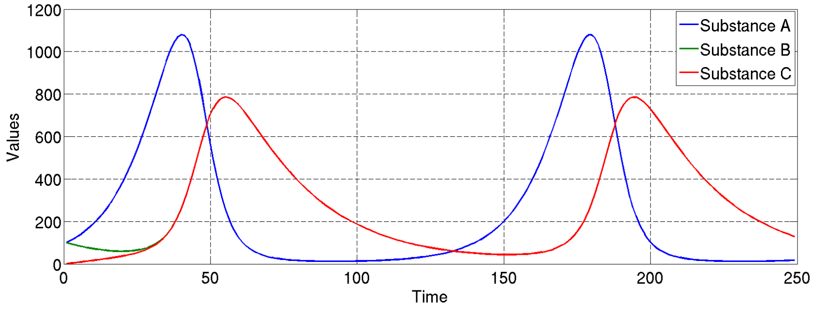

Fig. 5. Sirius' dynamics.

As an example, let us consider the dynamics given in Figure 5 related to a synthetic oscillator, introduced in [16] and very often considered in the MP theory, called *Sirius*. The oscillator is made of three substances A, B and C and five reactions whose regulators are supposed to depend on the set of substances given in Table 1.

For the metabolic oscillator Sirius, we want to apply LGSS for discovering the formulae of the regulators by assuming that they will be given by linear combinations of polynomial functions on substance quantities of degree less than or equal to 3. Since we do not know the right set of regressors which will be used, we should start LGSS by considering the set of all possible regressors given in Table 2. If we do this, however, the problem of multicollinearity arises since there

Set of the reactions	Dependence of the corresponding regulators
$r_1 : \emptyset \to A$	A, B and C
$r_2 : A \to B$	A and C
$r_3 : A \to C$	A and B
$r_4 : B \to \emptyset$	only B
$r_5 : C \to \emptyset$	only C

Table 1. The reactions (and the corresponding tuners) of the Sirius oscillator.

r_1 :	Constant, A, B, C, A^2, B^2, C^2, AB, AC, BC, A^3, B^3, C^3, A^2B, A^2C, B^2C, AB^2, AC^2, BC^2, ABC.
r_2 :	Constant, A, C, A^2, C^2, AC, A^3, C^3, A^2C, AC^2.
r_3 :	Constant, A, B, A^2, B^2, AB, A^3, B^3, A^2B, AB^2.
r_4 :	Constant, B, B^2, B^3.
r_5 :	Constant, C, C^2, C^3.

Table 2. The set of possible regressors for each regulator of Sirius.

are many regressors which are highly correlated with each other (for example the two regressors A^2 and A^3, whose correlation coefficient is equal to 0.98).

In order to overcome the problem, LGSS computes the *variance inflation factor* (VIF) for each regressor [1], which gives an idea of the degree of multicollinearity introduced by a regressor, when some other regressors are already in the regression equation. In LGSS the user can select a threshold value for the variance inflation factor, in order to avoid the insertion of collinear regressors. This solution, however, may affect the performance of the algorithm since the computing of VIF requires many additional computations [26].

Of course, a way to overcome the problem of multicollinearity is to drop collinear variables before launching the regression phase of LGSS. In the following we define an algorithm, based on a hierarchical clustering technique [11], which permits to cluster the time series of the regressors associated to the same reaction and to select those which are less correlated and that best satisfy the log-gain principle [16, 17], a principle developed in the MP theory based on a general criterion concerning the variations of quantities involved in biological phenomena [2]. The algorithm is based on the following procedure (we refer to [25] for details concerning the calculation of the log-gain score used in the algorithm).
For each reaction r in the MP system:

1. start by associating the time series of regressors of reaction r to different clusters.
2. Compute distances (similarities) between clusters. We consider the distance between one cluster and another cluster to be equal to the average distance from any time series of one cluster to any time series of the other cluster. The distance $d(g_1, g_2)$ of two regressors g_1, g_2 is given by the following two equations, where $corr(G_1^t, G_2^t)$ denotes the value of the Pearson's correlation coefficient [37] between the time series obtained by evaluating the two regressors on the t observed time points:
$$d(g_1, g_2) = 1 - |corr(G_1^t, G_2^t)|$$
if we do not want to distinguish between positive and negative correlations, and
$$d(g_1, g_2) = 1 - corr(G_1^t, G_2^t)$$
when we need to consider this.
3. Find the closest (most similar) pair of clusters and merge them into a single cluster (so that now we have one cluster less).
4. Repeat steps 2 and 3 until all the distances between clusters are greater than a user defined threshold value.
5. For each cluster computed, calculate the log-gain score of each regressor included (as defined in [25]) and select the one which have higher log-gain score. This permits to discard those regressors whose time series express changes which are not realistic in biology. The set of regressors for r which will be considered during the regression phase of LGSS are those collected at this step.

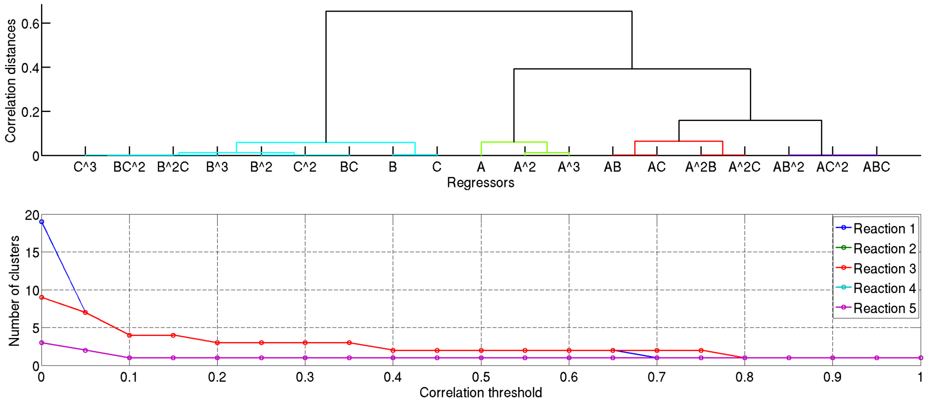

Fig. 6. On the top: the dendrogram which represents the clusters computed for the regressors of the rule r_1 of Sirius, by considering a threshold value of 0.1. On the bottom: the chart which displays the number of the computed clusters, for each reaction of Sirius, with respect to different threshold values of the maximum correlation distance between clusters.

r_1 : Constant, A, B, AB, AB^2.
r_2 : Constant, A, C, AC, AC^2.
r_3 : Constant, A, B, AB, AB^2.
r_4 : Constant, B.
r_5 : Constant, C.

Table 3. The set of possible regressors for each regulator of Sirius after the execution of the clustering algorithm. The total number of regressors is decreased of the 60% with respect to the set considered in Table 2.

The algorithm described above was included in LGSS and permits to solve the problem of multicollinearity without affecting the performance of the regression. In fact, LGSS launches this algorithm before starting the regression phase. The application of the algorithm, to the set of 48 regressors of Table 2, permits to reduce of more than the 60% the total set of regressors, by considering a threshold value of 0.1 (see Figure 6). The new set of regressors is given in Table 3.

The MP grammar computed by LGSS, starting from the set of regressors in Table 3, is given in Table 4 (see also Figure 7). This MP grammar is much better than the one given in Table 5, provided by LGSS with the initial set of possible regressors of Table 2. In fact, the new model uses less regressors (with lower degree) and permits a more clear comprehension of the regulative role of each substance of the system.

$r_1 : \emptyset \rightarrow A$	$\varphi_1 = 0.047 + 0.087A$
$r_2 : A \rightarrow B$	$\varphi_2 = 0.002A + 0.0002AC$
$r_3 : A \rightarrow C$	$\varphi_3 = 0.002A + 0.0002AB$
$r_4 : B \rightarrow \emptyset$	$\varphi_4 = 0.04B$
$r_5 : C \rightarrow \emptyset$	$\varphi_5 = 0.04C$

Table 4. The MP grammar of the Sirius oscillator, the dynamics is given in Figure 7.

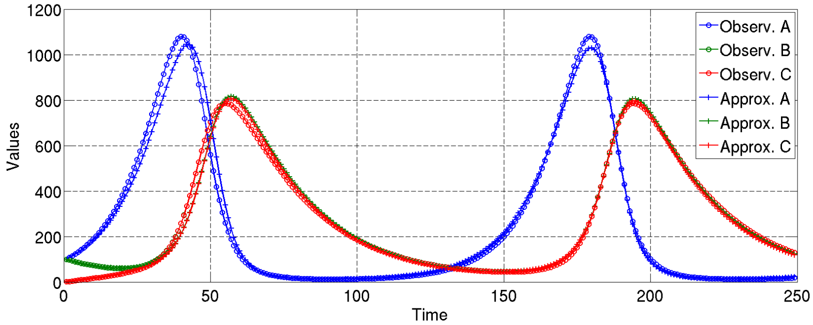

Fig. 7. Sirius' dynamics calculated by means of the MP grammar given in Table 4.

4 Conclusions

In this paper, a new algebraic formulation of inverse dynamical problems, based on MP grammars and Kronecker product, has been given, which provides, in general terms, the logic underlying the LGSS algorithm and proves its correctness in the approximate solution of inverse dynamical problems.

Even if computational tools are available for evaluating unknown parameters of ODE models [14, 8], LGSS seems to point out a more general methodology. In fact, LGSS not only discovers unknowns parameters, but suggests also the form of regulators as a combination of basic functions. This possibility could be very important in the case where the knowledge about the phenomenon under investigation is so poor that no clear idea is available about the kind of model underlying the observed behaviour.

The LGSS algorithm introduces a new perspective in the analysis of time series produced by the variables of a system evolving in time. This perspective can be defined as a generative one, where phenomena observed in time are reconstructed in terms of variable influences/transformations determined by the internal global states of the system. This approach is of course relevant in systems biology, but has a wide field of applications. In fact, in all the cases where some variables change according to some mutual relationship, due to mutual and systemic logic involving them, we are allowed to apply this general paradigm of discrete mathematical analysis.

$r_1 : \emptyset \rightarrow A$	$\varphi_1 = 1.06 + 0.082A$
$r_2 : A \rightarrow B$	$\varphi_2 = 3.6 \cdot 10^{-6}A^2 + 0.0002AC$
$r_3 : A \rightarrow C$	$\varphi_3 = 0.004B + 8.9 \cdot 10^{-7}A^2 + 0.0001AB + 3.3 \cdot 10^{-8}A^2B$
$r_4 : B \rightarrow \emptyset$	$\varphi_4 = 0.04B$
$r_5 : C \rightarrow \emptyset$	$\varphi_5 = 0.04C$

Table 5. The MP grammar of the Sirius oscillator computed by LGSS starting from the set of regressors given in Table 2. Due to the problem of the multicollinearity of regressors, the MP grammar is more complicated than the one given in Table 4.

References

[1] A. D. Aczel and J. Sounderpandian. *Complete Business Statistics.* Mc Graw Hill, International Edition, 2006.

[2] L.von Bertalanffy. *General Systems Theory: Foundations, Developments, Applications.* George Braziller Inc., New York, 1967.

[3] G. Ciobanu, Gh. Păun, and M.J. Pérez-Jiménez (Eds.), editors. *Applications of Membrane Computing.* Springer, 2006.

[4] D.W. Corne and P. Frisco. Dynamics of HIV infection studied with cellular automata and conformon-P systems. *Biosystems*, 91(3):531–544, 2008.

[5] N. Draper and H. Smith. *Applied Regression Analysis, 2nd Edition.* John Wiley & Sons, New York, 1981.

[6] M. Gheorghe, N. Krasnogor, and M. Camara. P systems applications to systems biology. *Biosystems*, 91(3):435–437, 2008.

[7] R.R. Hocking. The Analysis and Selection of Variables in Linear Regression. *Biometrics*, 32, 1976.

[8] S. Hoops, S. Sahle, R. Gauges, C. Lee, and J. Pahle. COPASI-a COmplex PAthway SImulator. *Bioinformatics*, 22(24), 2006.

[9] R.A. Horn and C.R. Johnson. *Topics in Matrix Analysis.* Cambridge University Press, 1991.

[10] A.K. Jain. *Fundamentals of Digital Image Processing.* Prentice Hall, 1989.

[11] S.C. Johnson. Hierarchical Clusterin Schemes. *Psychometrika*, 2:241–254, 1967.

[12] T. Karya and H. Kurata. *Generalized Least Squares.* Wiley, 2004.

[13] D.G. Luenberger. *Optimization by Vector Space Methods.* John Wiley & Sons Inc., 1969.

[14] T. Maiwald and J. Timmer. Dynamical modeling and multi-experiment fitting with PottersWheel. *Bioinformatics*, 24(18):2037–2043, 2008.

[15] V. Manca. String Rewriting and Metabolism: A logical perspective. In *Computing with Bio-Molecules*, pages 36–60. Springer-Verlag, 1998.

[16] V. Manca. The metabolic algorithm for P systems: Principles and applications. *Theoretical Computer Science*, 404:142–155, 2008.

[17] V. Manca. *Algorithmic Bioprocesses*, chapter 28: Log-Gain Principles for Metabolic P Systems, pages 585–605. Natural Computing. Springer-Verlag, 2009.

[18] V. Manca. From P to MP Systems. *WMC 2009, LNCS*, 5957:74–94, 2009.

[19] V. Manca. Fundamentals of Metabolic P Systems. In *[36]*, chapter 19, pages 475–498. Oxford University Press, 2010.

[20] V. Manca. Metabolic P Dynamics. In *[36]*, chapter 20, pages 499–528. Oxford University Press, 2010.

[21] V. Manca. Metabolic P systems. *Scholarpedia*, 5(3):9273, 2010.

[22] V. Manca and L. Bianco. Biological networks in metabolic P systems. *BioSystems*, 91(3):489–498, 2008.

[23] V. Manca and L. Marchetti. Metabolic approximation of real periodical functions. *The Journal of Logic and Algebraic Programming*, 79:363–373, 2010.

[24] V. Manca and L. Marchetti. Goldbeter's Mitotic Oscillator Entirely Modeled by MP Systems. *CMC 2010, LNCS 6501*, pages 273–284, 2010.

[25] V. Manca and L. Marchetti. Log-Gain Stoichiometic Stepwise regression for MP systems. *Int. Journal of Foundations of Computer Science*, 22(1):97–106, 2011.

[26] V. Manca and L. Marchetti. Solving Dynamical Inverse Problems by means of Metabolic P Systems. *BioSystems*, 2012. DOI:10.1016/j.biosystems.2011.12.006.

[27] V. Manca and L. Marchetti. Application of the MP theory to systems biology. In *Proceedings of the International Conference on Bio-inspired Systems and Signal Processing*, pages 303–308. SciTePress, 2012. DOI: 10.5220/0003852003030308.

[28] V. Manca and M.D. Martino. From String Rewriting to Logical Metabolic Systems. In *Grammatical Models of Multiagent Systems vol. 8*, pages 297–315. Gordon and Breach Science Publishers, 1999.

[29] V. Manca, L. Bianco, and F. Fontana. Evolutions and Oscillations of P systems: Theoretical Considerations and Application to biological phenomena. *WMC5 2004, LNCS*, 3365:63–84, 2005.

[30] V. Manca, G. Franco, and G. Scollo. State Transition Dynamics: basic concepts and molecular computing perspectives. In *Molecular Computational Models*, pages 32–55. IDEA Group INC., 2005.

[31] V. Manca, L. Marchetti, and R. Pagliarini. MP Modelling of Glucose-Insulin Interactions in the Intravenous Glucose Tolerance Test. *Int. Journal of Natural Computing Research*, 2(3):13–24, 2011.

[32] L. Marchetti and V. Manca. A methodology based on MP theory for gene expression analysis. *CMC 2011, LNCS 7184*, pages 300–313, 2012.

[33] Gh. Păun. Computing with membranes. *J. Comput. System Sci.*, 61(1): 108–143, 2000.

[34] Gh. Păun. *Membrane Computing. An Introduction.* Springer, Berlin, 2002.

[35] Gh. Păun. A quick introduction to membrane computing. *Journal of Logic and Algebraic Programming*, 79(6):291–294, 2010.

[36] Gh. Păun, G. Rozenberg, and A. Salomaa, editors. *Handbook of Membrane Computing*. Oxford University Press, 2010.

[37] K. Pearson. Notes on the History of Correlation. *Biometrika*, 13(1):25–45, 1920.

[38] F.J. Romero-Campero and M.J. Pérez-Jiménez. Modelling gene expression control using P systems: The Lac Operon, a case study. *Biosystems*, 91(3): 438–457, 2008.

[39] A. Spicher, O. Michel, M. Cieslak, J.L. Giavitto, and P. Prusinkiewicz. Stochastic P systems and the simulation of biochemical processes with dynamic compartments. *Biosystems*, 91(3):458–472, 2008.

[40] W.H. Steeb. *Matrix Calculus and Kronecker Product with Applications and C++ Programs*. World Scientific Publishing, 1997.

[41] W.H. Steeb. *Problems and Solutions in Introductory and Advanced Matrix Calculus*. World Scientific Publishing, 2006.

[42] T. Strutz. *Data Fitting and Uncertainty. A practical introduction to weighted least squares and beyond*. Vieweg+Teubner, 2010.

[43] J. Wolberg. *Data Analysis Using the Method of Least Squares: Extracting the Most Information from Experiments*. Springer, 2005.