A FAST ALGORITHM FOR COMPUTING A MINIMAL DECOMPOSITION OF A METABOLIC FLUX VECTOR IN TERMS OF ELEMENTARY FLUX VECTORS

R.M. Jungers[1], F. Zamorano[2], V.D. Blondel[1], A. Vande Wouwer[2], G. Bastin[1]

[1]Cesame, Department of Mathematical Engineering, Université catholique de Louvain, Belgium,

[2]Service d'Automatique, Faculté polytechnique de Mons, Belgium

Corresponding author: G. Bastin, Cesame, Université catholique de Louvain,
4, Avenue G. Lemaitre, 1348 Louvain-la-Neuve, Belgium,
Georges.Bastin@uclouvain.be

Abstract. The concept of elementary flux vector is valuable in a number of applications of metabolic engineering. For instance, in metabolic flux analysis, each admissible flux vector can be expressed as a non-negative linear combination of a small number of elementary flux vectors. However a critical issue concerns the number of elementary flux vectors which may be huge because it combinatorially increases with the size of the metabolic network. In this paper we present a fast algorithm that randomly computes admissible flux vectors having a minimal decomposition without explicitly enumerating all the elementary flux vectors of the network. The method is illustrated with an experimental case-study on CHO cells where the network has 65329 elementary flux vectors while the admissible flux distributions are expressed as a combination of 22 elementary vectors only.

1 Metabolic networks and elementary flux vectors

The intracellular metabolism of living cells is usually represented by a metabolic network under the form of a directed hypergraph that encodes a set of biochemical reactions taking place within the cell. In this hypergraph, the nodes represent the metabolites and the edges represent the metabolic fluxes.

According to the quasi steady-state paradigm of metabolic flux analysis (MFA) (e.g. [10]), it is assumed that the fluxes are balanced at each internal node. This means that the net sum of production and consumption fluxes, weighted by their stoichiometric coefficients, is zero for each internal metabolite of the network. This is expressed by the algebraic relation:

$$\mathbf{N}\mathbf{v} = \mathbf{0} \qquad \mathbf{v} \geqslant \mathbf{0} \tag{1}$$

where $\mathbf{v} = (v_1, v_2, \ldots, v_m)^T$ is the m-dimensional column vector of fluxes and $\mathbf{N} = [n_{ij}]$ is the $n \times m$ stoichiometric matrix of the metabolic network (m is the number of fluxes and n the number of internal nodes of the network). More precisely, a flux v_j denotes the rate of reaction j and a non-zero n_{ij} is the stoichiometric coefficient of the metabolite i in reaction j.

For a given metabolic network, the set S of possible flux distributions is the set of vectors $\mathbf{v}$ that satisfy the linear system (1). This set S is the pointed polyhedral cone resulting from the intersection of the kernel of $\mathbf{N}$ with the nonnegative orthant. This implies that there exists a set of elementary flux vectors $\mathbf{e}_i$ ([11]) which are the edges (or extremal rays) of the polyhedral cone and such that any flux distribution $\mathbf{v}$ can be expressed as a non-negative linear combination of the vectors $\mathbf{e}_i$ which form therefore a *unique* convex basis (see e.g. [12]) of the flux space S:

$$\mathbf{v} = w_1\mathbf{e}_1 + w_2\mathbf{e}_2 + \cdots + w_q\mathbf{e}_q \quad w_i \geqslant 0. \tag{2}$$

The $m \times q$ non-negative matrix $\mathbf{E}$ with column vectors $\mathbf{e}_i$ obviously satisfies $\mathbf{N}\mathbf{E} = 0$ and (2) can be written in matrix form as

$$\mathbf{v} = \mathbf{E}\mathbf{w} \quad \text{with} \quad \mathbf{w} \triangleq (w_1, w_2, \ldots, w_q)^T. \tag{3}$$

2 Metabolic flux analysis

Metabolic flux analysis (MFA) is the exercise of calculating the admissible flux distributions $\mathbf{v}$ that satisfy the steady state balance equation $\mathbf{N}\mathbf{v} = \mathbf{0}$ together with an additional set of linear constraints added by using experimental measurements. Here we consider the case where the measurements are collected in a vector $\mathbf{v}_m$ which is a linear function of the unknown flux distribution $\mathbf{v}$ and is expressed as

$$\mathbf{v}_m = \mathbf{P}\mathbf{v} \tag{4}$$

where $\mathbf{P}$ is a given $p \times m$ full-rank matrix. In addition, it is assumed that $\mathbf{P}\mathbf{e}_i \neq 0 \; \forall i$ or, in other terms, that the elementary flux vectors $\mathbf{e}_i$ do not belong to the kernel of the matrix $\mathbf{P}$. Then, from equations (1)-(4), we have the following fundamental equation of metabolic flux analysis

$$\Sigma \begin{pmatrix} \mathbf{v} \\ 1 \end{pmatrix} = \mathbf{0} \quad \text{with} \quad \Sigma \triangleq \begin{pmatrix} \mathbf{N} & \mathbf{0} \\ \mathbf{P} & -\mathbf{v}_m \end{pmatrix} \quad \text{and} \quad \mathbf{v} \geqslant \mathbf{0}. \tag{5}$$
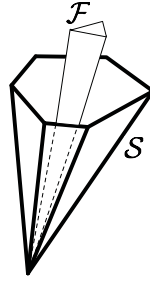
Figure 1: Illustration of the flux spaces S and $\mathscr{F}$.

For a given metabolic network and a given set of measurements, the solution of the MFA problem is defined as the set $\mathscr{F}$ of admissible flux distributions i.e. the set of non-negative vectors $\mathbf{v}$ that satisfy the homogeneous linear system (5). Each admissible $\mathbf{v}$ must be such that the non-negative vector $(\mathbf{v}^T\ 1)^T$ belongs to the kernel of the matrix Σ. Hence, as emphasized in [7, Chapter 4]-[8], the set $\mathscr{F}$ is a polytope in the positive orthant $\mathbb{R}^m_+$. This means that any admissible flux distribution $\mathbf{v}$ can be expressed as a convex combination of a set of non-negative basis vectors $\mathbf{f}_i$ which are the vertices of this polytope and form therefore a *unique* convex basis of the flux space $\mathscr{F}$. In other words, the solution of the MFA problem is the *admissible flux space* $\mathscr{F}$ defined as

$$\mathscr{F} \triangleq \Big\{ \mathbf{v} : \mathbf{v} = \sum_i \alpha_i \mathbf{f}_i, \ \ \alpha_i \geqslant 0, \ \ \sum_i \alpha_i = 1 \Big\}. \tag{6}$$

The admissible flux space $\mathscr{F}$ is a subset of the possible flux space S. In geometric terms, the polytope $\mathscr{F}$ defines a subcone of the pointed cone S as illustrated in Fig.1.

3   Minimal decomposition of $\mathbf{v} \in \mathscr{F}$ in terms of elementary vectors $\mathbf{e}_i$

For any admissible flux vector $\mathbf{v}$ in the polytope $\mathscr{F}$ satisfying equation (5), it must be emphasized that the decomposition of $\mathbf{v}$ in the convex basis $\{\mathbf{e}_i\}$ is **not** unique. Our aim is to determine minimal decompositions which can be useful in pratical applications of MFA. Using (3), system (5) is equivalent to the system:

$$\begin{pmatrix} \mathbf{NE} \\ \mathbf{PE} \end{pmatrix} \mathbf{w} = \begin{pmatrix} \mathbf{0} \\ \mathbf{v}_m \end{pmatrix} \quad \mathbf{w} \geqslant 0. \tag{7}$$

We observe that the first equation $\mathbf{NEw} = \mathbf{0}$ is trivially satisfied independently of $\mathbf{w}$ since by definition $\mathbf{NE} = 0$. Hence, system (7) may be reduced to the second equation:

$$\mathbf{PEw} = \mathbf{v}_m \quad \mathbf{w} \geqslant \mathbf{0}.$$

or equivalently:

$$\begin{pmatrix} \mathbf{PE} & -\mathbf{v}_m \end{pmatrix} \begin{pmatrix} \mathbf{w} \\ 1 \end{pmatrix} = 0 \quad \mathbf{w} \geqslant \mathbf{0}. \tag{8}$$

In this form, it is clear that the set of admissible weighting vectors $\mathbf{w}$ that satisfy (8) again constitutes a convex polytope that we denote $\mathscr{H}$. Therefore there exists a set of appropriate edge vectors $\mathbf{h}_i$ such that any arbitrary convex combination of the form:

$$\mathbf{w} = \sum_i \beta_i \mathbf{h}_i \quad \beta_i \geqslant 0 \quad \sum_i \beta_i = 1 \tag{9}$$

is necessarily an admissible $\mathbf{w}$ satisfying (8). The convex basis vectors $\mathbf{h}_i$ have a critical property : the number of non-zero entries is equal to the size p of the vector $\mathbf{v_m}$ i.e. the number of measurements (see [3] and Section 3.5 in [7]). From a metabolic viewpoint, each vector $\mathbf{h}_i$ is a particular solution $\mathbf{w}$ of (8) corresponding to an admissible flux distribution $\mathbf{v}$:

$$\mathbf{v} = \mathbf{Eh}_i \quad \mathbf{v} \in \mathscr{F} \tag{10}$$

In this expression, the non-zero entries of the vector $\mathbf{h}_i$ are interpreted as the weights of the respective contributions of the corresponding elementary flux vectors $\mathbf{e}_i$ in the computation of the flux distribution $\mathbf{v}$.

An important issue concerns the number of distinct extremal rays or vertices that are generated when computing the cone S or the polytopes $\mathscr{F}$ and $\mathscr{H}$. This number may become very large because it combinatorially increases with the size of the underlying metabolic network. The Double Description (DD) method ([6]) is the simplest known algorithm for enumerating the extremal rays of a polyhedral cone (see [3] for a review). In the context of metabolic networks it has received various dedicated improvements that are documented in the literature (see e.g. [9], [4] and [5]). In practical applications of MFA, the enumeration of all extremal rays is not necessarily a

critical objective. In many applications it is sufficient to know only one minimal decomposition of some vectors $\mathbf{v} \in \mathscr{F}$ in terms of elementary vectors $\mathbf{e}_i$. It clearly follows from our analysis that such a minimal decomposition involves ℓ terms with $p \leqslant \ell \leqslant (m-n)$. Furthermore, according to (10) there necessarily exist admissible $\mathbf{v}$ having a decomposition that involves only p terms. Computing this decomposition may be very expensive at first sight since the dimension of $\mathbf{E}$ is not bounded by a polynomial in the sizes of $\mathbf{N}$ and $\mathbf{P}$.

In this paper we present a fast algorithm that randomly computes vectors $\mathbf{v} \in \mathscr{F}$ having such a minimal decomposition from the sole knowledge of the stoichiometric matrices $\mathbf{N}$, $\mathbf{P}$ and the measurement vector $\mathbf{v}_m$ but without explicitly enumerating the extremal rays of the cone S (i.e. the columns of the huge matrix $\mathbf{E}$) and therefore without solving the system (8).

4 The algorithm

Let us first consider the following simple problem: We are given a vector $\mathbf{v}$ that belongs to a cone S, and we would like to express this vector as a linear combination of a few extremal rays of S.

Let us denote $a = \mathbf{u}^T \mathbf{v}$ the sum of the entries in $\mathbf{v}$ ($\mathbf{u}$ denotes the vector whose all entries are equal to one). In the following we will consider without loss of generality the slightly different problem where we are looking for extremal rays $\mathbf{e}_i$ such that $\mathbf{u}^T \mathbf{e}_i = a$. Geometrically speaking, we cut the cone with a plane passing through $\mathbf{v}$ such that the intersection is a bounded polytope whose vertices correspond to extremal rays of the initial cone S. We are thus given a (bounded) polytope, and a vector $\mathbf{v}$ in this polytope and we want to express this vector $\mathbf{v}$ as a convex combination of vertices of the polytope.

The algorithm essentially relies on two observations: first, we do not need to know all the extremal rays, what we only need is a (small) subset, to express $\mathbf{v}$ as a convex combination of them. Second, all the constraints defining the different cones are linear, and so we can make use of Linear Programming (e.g. [2]). More precisely, the problem of finding a vertex of the polytope defined by the equations

$$\mathbf{Mx} = \mathbf{0}, \quad \mathbf{u}^T \mathbf{x} = a, \quad \mathbf{x} \geqslant \mathbf{0}$$

can be done in time polynomial in the number of constraints and the dimension. Indeed, consider the following linear program :

$$
\begin{aligned}
&\min \mathbf{d}^T \mathbf{x} \\
&\text{s.t.} \\
&\mathbf{Mx} = \mathbf{0}, \\
&\mathbf{x} \geqslant \mathbf{0}, \\
&\mathbf{u}^T \mathbf{x} = a.
\end{aligned}
\tag{11}
$$

If $\mathbf{d}$ is not parallel to a constraint of the program (11), then, the solution is a vertex of the corresponding polytope (see for instance [2]). So in practice, if $\mathbf{d}$ is a random direction, an extremal ray is found with probability one.

Let us now present our algorithm which proceeds iteratively by projecting $\mathbf{v}$ on faces $\mathscr{P}_i$ of the polytope $\mathscr{P}$ described by the constraints of the program (11). Since the dimension of the faces $\mathscr{P}_i$ strictly decreases at each step, the algorithm takes at most $k-1$ steps, where k is the dimension of the cone S.

Take any extremal ray $\mathbf{e}_1$ of the cone S (for instance by solving the linear program (11)); then the vector $\mathbf{v}$ can be written as the convex combination of $\mathbf{e}_1$ and of a vector $\mathbf{v}_1$, which belongs to a face $\mathscr{P}_1$ of S: $\mathbf{v} = \gamma_1 \mathbf{e}_1 + (1 - \gamma_1) \mathbf{v}_1$. These quantities $\mathbf{v}_i, \gamma_i$ are easy to compute, as $\mathbf{v}_1$ is the solution $\mathbf{x}^*$ of the Linear Program

$$
\begin{aligned}
&\max \mu \\
&\text{s.t.} \\
&\mathbf{Mx} = \mathbf{0}, \\
&\mathbf{x} \geqslant \mathbf{0}, \\
&\mathbf{u}^T \mathbf{x} = a, \\
&\mathbf{v} + \mu (\mathbf{v} - \mathbf{e}_1) = \mathbf{x}.
\end{aligned}
\tag{12}
$$

The geometric meaning of this linear program is as follows: starting fom the vector $\mathbf{v}$ one tries to find a point $\mathbf{x}$ which is diametrically opposite to $\mathbf{e}_1$ and as far as possible from $\mathbf{v}$. Clearly this point will be on a face of the polytope (because if it is not, it is possible to go further). Here μ represents the distance from $\mathbf{v}$ to $\mathbf{x}$.

Now $\mathscr{P}_i$ is a new polyhedron, and we still can express $\mathbf{v}_i$ as a convex combination of a vertex of $\mathscr{P}_i$ (which is also a vertex of S) and a point $\mathbf{v}_{i+1}$ that belongs to a face $\mathscr{P}_{i+1}$ of $\mathscr{P}_i$ (which is also a face of S, but of dimension

strictly smaller than $\dim \mathscr{P}_i$). Thus, after $k' \leq k-1$ steps, the dimension of $\mathscr{P}_{k'}$ is equal to 0, which means that $\mathbf{v}_{k'}$ is actually a vertex of $\mathscr{P}$ which we denote $\mathbf{e}_{k'+1}$. Thus, $\mathbf{v}_{k'-1} = \gamma_{k'}\mathbf{e}_{k'} + (1-\gamma_{k'})\mathbf{e}_{k'+1}$. Finally we can write:

$$\begin{aligned} \mathbf{v} &= \mathbf{v}_0 \\ &= \gamma_1\mathbf{e}_1 + (1-\gamma_1)(\gamma_2\mathbf{e}_2 + (1-\gamma_2)(\dots(\gamma_{k'}\mathbf{e}_{k'} + (1-\gamma_{k'})\mathbf{e}_{k'+1}))) \\ &= \sum_1^{k'+1} w_i\mathbf{e}_i, \end{aligned}$$

with $\sum w_i = 1$. Finally, as the dimension of the cone S is equal to $k = m-n$, we obtain at most $m-n$ extremal vectors $\mathbf{e}_i$. We have thus found the decomposition in polynomial time, which is a dramatic improvement compared to the naive brute force approach that requires the enumeration of all vectors $\mathbf{e}_i$.

We now would like to express a vector $\mathbf{v}$ in $\mathscr{F}$ (that is a vector compatible with the measurements in $\mathbf{v}_m$) as a linear combination of extremal rays of S. Moreover we would like to minimize the number of extremal rays in this expression. Equation (9) and the remark below ensure us that there is such a vector $\mathbf{v}$ that can be expressed as a combination of only p extremal rays $\mathbf{e}_i$ of S. To see this, consider the expression (8) of the polytope $\mathscr{H}$, which describes the set of admissible values of $\mathbf{w}$. It can be defined by only p equalities, so that $dim(\mathbf{w}) - p$ inequality constraints can be activated to define an extremal ray $\mathbf{h}_i$ of $\mathscr{H}$. In conclusion, there are admissible vectors $\mathbf{w}$ (the extremal rays of $\mathscr{H}$), that only contain at most p nonzero values. However, if one does not want to compute the matrix $\mathbf{E}$ of extremal rays of S, this is not an easy task a priori to find such a minimal representation. Indeed, the dimension of $\mathbf{w}$ is exponential in the size of the problem.

In order to compute such a "good" vector $\mathbf{v}$ and its corresponding decomposition, we introduce yet another cone $\mathscr{K} \subset \mathbb{R}^p$. This cone is the projection of S by the matrix $\mathbf{P}$:

$$\mathscr{K} = \{\mathbf{y} = \mathbf{Pv} : \mathbf{v} \geqslant \mathbf{0}, \mathbf{Nv} = \mathbf{0}\}.$$

The idea of the algorithm is as follows: We know that the vector $\mathbf{v}_m$ is in $\mathscr{K}$, and we will express this vector as a convex combination of p vectors, which are the projection of extremal rays $\mathbf{e}_i$ under the matrix $\mathbf{P}$. We start from an extremal ray $\mathbf{e}_1$ of the cone $\mathscr{F}$ (for instance by applying the Linear Program (11)); then the vector $\mathbf{v}_m = \mathbf{y}_0$ can be written as the convex combination of $\mathbf{Pe}_1$ and a vector $\mathbf{y}_1$, which belongs to a face $\mathbf{P}_1$ of $\mathscr{K}$: $\mathbf{v}_m = \alpha_1\mathbf{Pe}_1 + (1-\alpha_1)\mathbf{y}_1$. This vector $\mathbf{v}_1$ is easy to find with a line search in the cone $\mathscr{K}$ as in Program (12). Now, at each step, find an extremal ray $\mathbf{e}_i$ of $\mathscr{K}$ which is mapped to a face $\mathbf{P}_{i-1}$ of $\mathscr{K}$. Then $\mathbf{y}_{i-1}$ can be expressed as a convex combination of $\mathbf{Pe}_i$ and a vector $\mathbf{y}_i$ that belongs to a face $\mathbf{P}_i$ of $\mathbf{P}_{i-1}$. Since the dimension of $\mathbf{P}_i$ strictly decreases at each step, after $t \leq p$ steps the point $\mathbf{y}_t$ is actually an extremal ray of $\mathscr{K}$, and is thus the projection of an extremal ray $\mathbf{e}_{(t+1)}$ of S. Finally we have the relations:

$$\mathbf{v}_m = \sum_1^t \lambda_i\mathbf{Pe}_i = \mathbf{P}(\sum_1^t \lambda_i\mathbf{e}_i), \tag{13}$$

and thus the vector

$$\mathbf{v} = \sum_1^t \lambda_i\mathbf{e}_i$$

is a convex combination of at most p extremal vectors of S that satisfies (5).

5   Case study

As a matter of illustration and motivation to the methodology presented above, we consider the example of chinese hamster ovary (CHO) cells cultivated in batch mode in stirred flasks in a serum-free medium ([1]). During the growth phase, we assume that the cell metabolism is described by the metabolic network presented in Appendix A. The network involves the Glycolysis pathway, the Pentose-Phosphate pathway, the Krebs cycle, the amino-acid metabolism, the urea cycle as well as the nucleotide, protein and lipid synthesis (see [13] for further motivation and details).

For this network we have $m = 82$ fluxes and $n = 53$ internal metabolites, and there are 65329 elementary flux vectors $\mathbf{e}_i$ (i.e the polyhedral cone S has 65329 edges).

Moreover, there are $p = 22$ extra-cellular species whose degradation or accumulation rates in the culture medium are measured and collected in the vector $\mathbf{v}_m$ given in Table 1.

The algorithm of Section 4 is then implemented with these data. We present a trial where the resulting admissible flux vector $\mathbf{v}$ is given in Table 2. It can be checked to satisfy (5) and to be is fully consistent with the experimental

Glucose	-0,187130	Glutamine	-0,050246
Threonine	-0,001184	Lysine	-0,002125
Valine	-0,001956	Isoleucine	-0,001528
Leucine	-0,002601	Phenylalanine	-0,000998
Methionine	-0,000724	Asparagine	-0,001278
Arginine	-0,002142	Proline	-0,002142
Histidine	-0,003298	Tyrosine	-0,007610
Aspartate	-0,000318	Cysteine	-0,000923
Glycine	0,002230	Serine	-0,000923
Glutamate	-0,009548	Ammonia	0,045712
Lactate	0,344510	Alanine	0,008808

Table 1: Vector of measurements $\mathbf{v}_m$ (mM/(h $\times 10^9$ cells)), with a "-" sign for degradation and a "+" sign for accumulation.

data of Table 1. Furthermore, the algorithm provides the minimal decomposition of $\mathbf{v}$ as a non-negative linear combination of the 22 elementary flux vectors $\mathbf{e}_i$ given in Tables 3 and 4.

Let us insist that the obtained vector $\mathbf{v}$ is obviously just one possible solution among many others with a minimal decomposition. If the algorithm is re-run with the same initial data, it will find other solutions with a minimal decomposition because it makes use of random searching directions. Complementary results on the metabolic flux analysis of CHO cells can be found in the companion paper [13].

6 References

[1] J.S. Ballez, J. Mols, J. Burteau, S.N. Agathos, and Y-J. Schneider. Plant protein hydrolysates support CHO-320 cells proliferation and recombinant ifn-gamma production in suspension and inside microcarriers in protein-free media. *Cytotechnology*, 44:103–114, 2004.

[2] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, New-York, USA, 2004.

[3] K. Fukuda and A. Prodon. The double description method revisited. In R. Euler and M. E. Deza I. Manoussakis, editors, *Combinatorics and Computer Science*, volume 1120 of *Lecture Notes in Computation Sciences*, pages 91–111. Springer-Verlag, 1996.

[4] J. Gagneur and S. Klamt. Computation of elementary modes : a unifying framework and the new binary approach. *BMC Bioinformatics*, 5:175, 2004.

[5] S. Klamt, J. Gagneur, and A. von Kamp. Algorithmic approaches for computing elementary modes in large biochemical networks. *IEE Proceedings Systems Biology*, 152:249–255, 2005.

[6] T.S. Motzkin, H. Raiffa, G.L. Thompson, and R.M. Thrall. The double description method. In H.W. Kuhn and A.W. Tucker, editors, *Contribution to the Theory of Games Vol. II*, volume 28 of *Annals of Mathematical Studies*, pages 51–73, Princeton, New Jersey, 1953. Princeton University Press.

[7] A. Provost. *Metabolic design of dynamic bioreaction models*. PhD thesis, Faculty of Engineering, Université Catholique de Louvain, November 2006.

[8] A. Provost and G. Bastin. Metabolic flux analysis : an approach for solving non-stationary underdetermined systems. In *CD-Rom Proceedings 5th MATHMOD Conference*, Paper 207 in Session SP33, Vienna, Austria, 2006.

[9] R. Schuster and S. Schuster. Refined algoritm and computer program for calculating all non-negative fluxes admissible in steady states of biochemical reaction systems with or without some flux rates fixed. *Computer Applications in the Biosciences*, 9(1):79–85, February 1993.

[10] G. Stephanopoulos, J. Nielsen, and A. Aristidou. *Metabolic Engineering : Principles and Methodologies*. Academic Press, San Diego, 1998.

[11] R. Urbanczik. Enumerating constrained elementary flux vectors of metabolic networks. *IET Systems Biology*, 1(5):274–279, 2007.

[12] H. Weyl. The elementary theory of convex polyhedra. In *Contributions to the Theory of Games Vol. I*, Annals of Mathematical Studies, pages 3–18, Princeton, New Jersey, 1950. Princeton University Press.

[13] F. Zamorano, A. Vande Wouwer, and G. Bastin. Metabolic flux interval analysis of CHO cells. In *Proceedings MATHMOD 2009 - 6th Vienna International Conference on Mathematical Modelling*, Vienna, Austria, February 2009.

Appendix A. Metabolic network.

Glycolisis
v1: Glu + ATP → G6P + ADP
v2: G6P + ATP → DHAP + G3P + ADP
v3: DHAP → G3P
v4: G3P + OxP + P_i + ADP → (3)PG + RdP + ATP
v5: (3)PG + ADP → Pyr + ATP

Krebs Cycle
v6:Pyr + OxP → AcCoA + CO_2 + RdP
v7:AcCoA + Oxal → Cit
v8:Cit + OxP → αKG + CO_2 + RdP
v9:αKG + OxP → SucCoA + CO_2 + RdP
v10:SucCoA + ADP + P_i → Succ + ATP
v11:Succ → Fum
v12:Fum → Mal
v13:Mal + OxP → Oxal + RdP

Pyruvate Fates
v14:Pyr + RdP → Lact + OxP
v15:Pyr + Glu → Ala + αKG

Pentose Phosphate Pathway
v16: G6P+ 2 OxP → R5P+ 2 RdP+CO_2
v17: 3 R5P → 2.5 G6P+ 0.5 P_i

Anaplerotic Reaction
v18: Mal+ OxP → Pyr+ CO_2+RdP

Amino Acid Metabolism
v19:Glu + OxP → αKG + NH_4^+ + RdP
v20:Oxal + Glu → Asp + αKG
v21: Gln → Glu + NH_4^+
v22: Thr+OxP → Gly+AcCoA+RdP
v23: Gly+ OxP → CO_2 + NH_4^+ +RdP
v24: (3)PG + OxP + Glu → Ser + αKG +RdP +P_i
v25: Ser → Gly
v26: Ser → Pyr + NH_4^+
v27: Thr → αKb + NH_4^+
v28: αKb + OxP → PropCoA + RdP +CO_2
v29: PropCoA + CO_2 +ATP → SucCoA +ADP + P_i
v30: Lys + 2 αKG + OxP → αKa + 2 Glu +RdP v31: αKa + 2 OxP → AcetoAcCoA + 2 RdP+ 2 CO_2
v32: AcetoAcCoA → 2 AcCoA
v33: Val + αKG → αKv + Glu
v34: αKv + 3 OxP → PropCoA + 2 CO_2 + 3 RdP
v35: Ile + αKG → (3)Methyl(2)oxovalerate+Glu
v36: (3)Methyl(2)oxovalerate+ 2 OxP → AcCoA + PropCoA + CO_2 + 2 RdP
v37: Leu + αKG → αKi + Glu
v38: αKi + OxP + ATP → AcCoA + AcetoAc + RdP + ADP +P_i
v39: AcetoAc + SucCoA → AcetoAcCoA + Succ
v40: Phe + RdP → Tyr + OxP
v41: Tyr + αKG → Fum + Glu + AcetoAc +CO_2
v42: Met + ATP → HomoCys +AMP + P_i
v43: HomoCys + Ser → αKb + Cys + NH_4^+
v44: Cys → Pyr + NH_4^+
v45: Asn → Asp + NH_4^+
v46: Arg → Ornitine + Urea
v47: Ornitine + αKG → GluγSA + Glu
v48: Pro → GluγSA
v49: GluγSA + OxP → Glu +RdP
v50: His → Glu + NH_4^+
v51: Asp_{ext} → Asp
v52: Cys_{ext} → Cys

v53: Gly → Gly$_{ext}$
v54: Ser$_{ext}$ → Ser
v55: Glu$_{ext}$ → Glu
v56: Tyr$_{ext}$ → Tyr
v57: Ala → Ala$_{ext}$

Protein Synthesis
v58: 0.023 His + 0.053 Ile + 0.091 Leu + 0.059 Lys + 0.023 Met + 0.039 Phe + 0.059 Thr + 0.014 Trp + 0.066 Val + 0.051 Arg + 0.019 Cys + 0.042 Gln + 0.072 Gly + 0.052 Pro + 0.032 Tyr + 0.78 Ala + 0.043 Asn + 0.053 Asp + 0.063 Glu + 0.068 Ser + 3 ATP → Protein + AMP + Pp$_i$ + 2 ADP + 2P$_i$

Nucleotide Synthesis
v59: R5P + ATP → PRPP + AMP
v60: PRPP + 2 Gln + Asp + Gly + 4 ATP + CO$_2$ → IMP + 2 Glu + Fum + 4 ADP + 4 P$_i$ + Pp$_i$
v61: IMP + Asp + 3ATP → ATP$_{RN}$ + Fum + 3ADP + P$_i$
v62: IMP + Gln + 3ATP + OxP → GTP$_{RN}$ + Glu + 2ADP + AMP + Pp$_i$ + RdP
v63: CO$_2$ + NH$_4^+$ + Asp + 2 ATP + OxP → Orotate + RdP + 2ADP + 2P$_i$
v64: Orotate + PRPP + ATP → UTP$_{RN}$ + CO$_2$ + 2ADP + Ppi
v65: UTP$_{RN}$ + Gln + ATP → CTP$_{RN}$ + Glu + ADP + P$_i$
v66: 0.285 ATP$_{RN}$ + 0.285 UTP$_{RN}$ + 0.215 GTP$_{RN}$ + 0.215 CTP$_{RN}$ → RNA
v67: ATP$_{RN}$ → dATP
v68: GTP$_{RN}$ → dGTP
v69: UTP$_{RN}$ → dTTP
v70: CTP$_{RN}$ → dCTP
v71: 0.285 dATP + 0.285 dTTP + 0.215 dGTP + 0.215 dCTP → DNA

Lipid Synthesis
v72: DHAP + RdP → Glyc3P + OxP
v73: Glyc3P + 18 AcCoA + 21 ATP + 33 RdP → PA + 16 (ADP + P$_i$) + 33 OxP + 5 (AMP + Pp$_i$)
v74: PA → (1,2)DG + P$_i$
v75: Eth + (1,2)DG + 2 ATP → PE + ADP + P$_i$ + AMP + Pp$_i$
v76: Chol + 1,2)DG + 2 ATP → PC + ADP + P$_i$ + AMP + Pp$_i$
v77: PE + Ser → PS + Eth
v78: 8 AcCoA + 8 ATP + 15 RdP + Ser → Sphg + 7(ADP + P$_i$) + 15 OxP + CO$_2$ + AMP + Pp$_i$
v79: Sphg + 8 AcCoA + 8 ATP + 14 RdP → Cer + 7(ADP + P$_i$) + 14 OxP + AMP + Pp$_i$
v80: Cer + PC → SM + (1,2)DG
v81: 6 AcetoAcCoA + 6 AcCoA + 18 ATP + 14 RdP → Cholesterol + 14 OxP + 18 ADP + 4 Pp$_i$ + 6P$_i$ + 6 CO$_2$
v82: 0.5 PC + 0.2 PE + 0.075 PS + 0.075 SM + 0.15 Cholesterol → Membrane Lipid

v1	0.18713	v21	0.01570	v41	0.00849	v61	0.00687	v81	0.00025
v2	0.14891	v22	0.00000	v42	0.00068	v62	0.00518	v82	0.00167
v3	0.14762	v23	0.00000	v43	0.00068	v63	0.01205		
v4	0.29653	v24	0.04879	v44	0.00157	v64	0.01205		
v5	0.24774	v25	0.01440	v45	0.00120	v65	0.00518		
v6	0.00000	v26	0.03426	v46	0.00205	v66	0.00000		
v7	0.00000	v27	0.00108	v47	0.00205	v67	0.00687		
v8	0.00000	v28	0.00176	v48	0.00205	v68	0.00518		
v9	0.06698	v29	0.00504	v49	0.00410	v69	0.00687		
v10	0.06109	v30	0.00202	v50	0.00326	v70	0.00518		
v11	0.07202	v31	0.00202	v51	0.00023	v71	0.02411		
v12	0.09943	v32	0.01145	v52	0.00089	v72	0.00129		
v13	0.02953	v33	0.00184	v53	0.00235	v73	0.00129		
v14	0.34451	v34	0.00184	v54	0.00013	v74	0.00129		
v15	0.00894	v35	0.00144	v55	0.00944	v75	0.00046		
v16	0.10879	v36	0.00144	v56	0.00755	v76	0.00096		
v17	0.02823	v37	0.00244	v57	0.00881	v77	0.00013		
v18	0.06988	v38	0.00244	v58	0.00173	v78	0.00013		
v19	0.00000	v39	0.01093	v59	0.02411	v79	0.00013		
v20	0.02955	v40	0.00093	v60	0.01205	v80	0.00013		

Table 2: A vector **v** of admissible metabolic flux rates (mM/(h × 10^9 cells)) consistent with the metabolic network and the experimental data of Table 1.

	e1	e2	e3	e4	e5	e6	e7	e8	e9	e10	e11	e12	e13	e14	e15	e16	e17	e18	e19	e20	e21	e22
v1	31	3	0	0	134.5	3	3	551	372.5	134.5	372.5	461.75	253.5	1254.42	3	17.44	23.93	354.5	3	3	3	23.93
v2	31	0	0	0	134.5	3	3	551	372.5	134.5	372.5	212.1	253.5	265.68	3	3.49	9.98	354.5	3	3	3	9.98
v3	0	0	0	0	103.5	3	3	520	341.5	103.5	341.5	181.1	222.5	234.68	3	3.49	9.98	323.5	3	3	3	9.98
v4	31	0	0	0	238	6	6	1071	714	238	714	393.2	476	500.37	6	6.98	19.95	678	6	6	6	19.95
v5	31	0	0	0	0	6	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
v6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
v7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
v8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
v9	0	0	3	0	0	0	6	357	357	0	0	357	238	1413.89	6	24.91	19.95	0	6	6	6	19.95
v10	0	0	3	0	0	0	6	0	0	0	0	0	0	1056.89	6	24.91	19.95	606	6	6	6	19.95
v11	0	0	3	0	238	0	6	357	357	238	0	357	238	1413.89	6	24.91	19.95	642	6	6	6	19.95
v12	0	0	3	0	238	0	6	714	714	238	0	909.98	238	2547.04	6	35.86	30.91	678	6	6	6	30.91
v13	0	0	0	0	0	0	0	0	0	0	0	320.8	0	1270.52	0	17.93	0	0	0	0	0	0
v14	0	0	3	3	476	6	6	1779	1422	470	708	851.55	708	1276.52	12	17.93	43.88	1350	12	12	12	43.88
v15	714	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
v16	0	18	0	0	0	0	0	0	0	0	0	249.65	0	988.73	0	13.95	13.95	0	0	0	0	13.95
v17	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
v18	0	0	3	0	238	0	6	714	714	238	714	589.17	238	1276.52	6	17.93	30.91	678	6	6	6	30.91
v19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
v20	0	0	3	0	0	0	0	0	0	0	0	320.8	0	1270.52	0	17.93	0	0	6	6	6	0
v21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4.95	0	0	0	0	0	0
v22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
v23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
v24	0	0	0	0	238	0	6	1071	714	238	714	393.2	476	500.37	6	6.98	19.95	678	6	6	6	19.95
v25	0	0	0	0	232	0	0	0	0	0	0	124.83	0	494.37	0	6.98	6.98	0	0	0	0	6.98
v26	683	0	0	0	0	0	6	1065	708	232	708	262.37	470	0	6	0	12.98	672	6	6	6	12.98
v27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
v28	0	0	0	0	238	0	0	0	0	238	0	0	0	0	0	0	0	0	0	0	0	0
v29	0	0	0	0	238	0	0	0	0	238	0	0	0	0	0	0	0	642	0	0	0	0
v30	357	0	0	0	0	0	0	0	0	238	357	0	0	0	0	0	0	0	0	0	0	0
v31	357	0	0	0	0	0	0	0	0	0	357	0	0	0	0	0	0	0	0	0	0	0
v32	321	0	0	0	202	0	0	321	321	202	321	321	202	321	0	0	0	0	0	0	0	0
v33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
v34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
v35	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	642	0	0	0	0
v36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	642	0	0	0	0
v37	0	0	0	0	238	0	0	0	0	238	0	0	238	0	0	0	0	0	0	0	0	0
v38	0	0	0	0	238	0	0	0	0	238	0	0	238	0	0	0	0	0	0	0	0	0
v39	0	0	0	0	238	0	0	357	357	238	0	357	238	357	0	0	0	36	0	0	0	0
v40	0	0	0	0	0	0	0	0	0	0	0	0	0	357	0	0	0	36	0	0	0	0
v41	0	0	0	0	0	0	0	357	357	0	0	357	0	357	0	0	0	36	0	0	0	0

Table 3: Elementary vectors $\mathbf{e}_i$ (first 41 entries) of the minimal decomposition of the flux vector $\mathbf{v}$ of Table 2. The integer entries are exact stoichiometric coefficients. The other entries are truncated to the 2nd decimal.

	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8	e_9	e_{10}	e_{11}	e_{12}	e_{13}	e_{14}	e_{15}	e_{16}	e_{17}	e_{18}	e_{19}	e_{20}	e_{21}	e_{22}
v42	0	0	0	0	238	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
v43	0	0	0	0	238	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
v44	0	0	0	3	238	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
v45	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	17.93
v46	0	0	0	0	0	0	0	357	0	0	0	0	0	0	0	0	0	0	0	0	0	0
v47	0	0	0	0	0	0	0	357	0	0	0	0	0	0	0	0	0	0	0	0	0	0
v48	0	0	0	0	0	0	6	0	357	0	0	0	0	0	0	0	0	0	0	0	0	0
v49	0	0	0	0	0	0	6	357	357	0	0	0	0	0	0	0	0	0	0	0	6	0
v50	0	0	0	0	0	0	0	0	0	0	0	0	238	0	0	0	0	0	0	0	0	0
v51	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	17.93	0	0	0	0	0
v52	0	0	0	3	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
v53	0	0	0	0	232	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
v54	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0
v55	689	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
v56	0	0	0	0	0	0	0	357	357	0	0	357	0	0	0	0	0	0	0	0	0	0
v57	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
v58	9153.85	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
v59	0	0	0	0	0	0	0	0	0	0	0	249.65	0	988.73	0	13.95	13.95	0	0	0	0	13.95
v60	0	0	0	0	0	0	0	0	0	0	0	124.83	0	494.37	0	6.98	6.98	0	0	0	0	6.98
v61	0	0	0	0	0	0	0	0	0	0	0	71.15	0	281.79	0	3.98	3.98	0	0	0	0	3.98
v62	0	0	0	0	0	0	0	0	0	0	0	53.67	0	212.58	0	3	3	0	0	0	0	3
v63	0	0	0	0	0	0	0	0	0	0	0	124.83	0	494.37	0	6.98	6.98	0	0	0	0	6.98
v64	0	0	0	0	0	0	0	0	0	0	0	124.83	0	494.37	0	6.98	6.98	0	0	0	0	6.98
v65	0	0	0	0	0	0	0	0	0	0	0	53.67	0	212.58	0	3	3	0	0	0	0	3
v66	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
v67	0	0	0	0	0	0	0	0	0	0	0	71.15	0	281.79	0	3.98	3.98	0	0	0	0	3.98
v68	0	0	0	0	0	0	0	0	0	0	0	53.67	0	212.58	0	3	3	0	0	0	0	3
v69	0	0	0	0	0	0	0	0	0	0	0	71.15	0	281.79	0	3.98	3.98	0	0	0	0	3.98
v70	0	0	0	0	0	0	0	0	0	0	0	53.67	0	212.58	0	3	3	0	0	0	0	3
v71	0	0	0	0	0	0	0	0	0	0	0	249.65	0	988.73	0	13.95	13.95	0	0	0	0	13.95
v72	31	0	0	0	31	0	0	31	31	31	31	31	31	31	0	0	0	31	0	0	0	0
v73	31	0	0	0	31	0	0	31	31	31	31	31	31	31	0	0	0	31	0	0	0	0
v74	31	0	0	0	31	0	0	31	31	31	31	31	31	31	0	0	0	31	0	0	0	0
v75	11	0	0	0	11	0	0	11	11	11	11	11	11	11	0	0	0	11	0	0	0	0
v76	23	0	0	0	23	0	0	23	23	23	23	23	23	23	0	0	0	23	0	0	0	0
v77	3	0	0	0	3	0	0	3	3	3	3	3	3	3	0	0	0	3	0	0	0	0
v78	3	0	0	0	3	0	0	3	3	3	3	3	3	3	0	0	0	3	0	0	0	0
v79	3	0	0	0	3	0	0	3	3	3	3	3	3	3	0	0	0	3	0	0	0	0
v80	3	0	0	0	3	0	0	3	3	3	3	3	3	3	0	0	0	3	0	0	0	0
v81	6	0	0	0	6	0	0	6	6	6	6	6	6	6	0	0	0	6	0	0	0	0
v82	40	0	0	0	40	0	0	40	40	40	40	40	40	40	0	0	0	40	0	0	0	0

Table 4: Elementary vectors $\mathbf{e}_i$ (last 41 entries) of the minimal decomposition of the flux vector $\mathbf{v}$ of Table 2. The integer entries are exact stoichiometric coefficients. The other entries are truncated to the 2nd decimal.