Kleene's Theorem and the Solution of Metabolic Carbon Labeling Systems

Nicole Isermann Michael Weitzel Wolfgang Wiechert

Department of Simulation, University of Siegen
Paul-Bonatz-Str. 9-11, D-57068 Siegen, Germany
{isermann,weitzel,wiechert}@simtec.mb.uni-siegen.de

Abstract: Carbon Labeling Systems (CLS) are large equation systems that describe the dynamics of labeled carbon atoms in a metabolic network. The rapid solution of these systems is the algorithmic backbone of ^{13}C Metabolic Flux Analysis (MFA) which has become one of the most widely used tools in Metabolic Engineering. A new algorithm is presented for the solution of CLS which is not based on iteration schemes or numerical linear algebra methods but on path tracing of labeled particles. It is shown that the set of all paths from the system input to the internal network nodes directly gives the clue to an explicit solution of CLS. The promising potential of this new solution algorithm are outlined.

1 Metabolic Flux Analysis

In recent years Metabolic Flux Analysis (MFA) by using ^{13}C isotopes has become one of the most widely used tools in Metabolic Engineering [Wi01, W02]. MFA allows to determine quantitatively all fluxes in the central metabolism of a micro organism or higher cell. The metabolic flux maps resulting from this analysis serve to compare different strains of a micro organism to diagnose the effects of a genetic manipulation or even give hints to further improve the production capabilities of a given organism.

MFA is based on a carbon labeling experiment where ^{13}C labeled substrates are fed to the cells. The ^{13}C isotopes are then distributed all over the metabolic network due to the metabolic activity. Finally, the enrichment of ^{13}C isotopes in the intra-cellular metabolite pools tends to an isotopically stationary state, which means that constant fractions of unlabeled and labeled carbon atoms are encountered in all pools. In this state the isotope enrichment is measured by NMR or MS instruments [Sz98]. From this measurement data the metabolic fluxes are estimated based on a mathematical model of the carbon labeling dynamics in the system.

The sole biological knowledge that is required for this procedure is the biochemical structure of the metabolic network, and the carbon atom transitions in each single reaction step. This knowledge is rather well established for central metabolism, but the method can also be used to distinguish between network variances.

1.1 Modelling Carbon Atom Flow

The central concept for the modeling of carbon labeling systems (CLS) is that of an isotopomer of a metabolite. If a certain metabolite has n carbon atoms then there are 2^n differently labeled types of this metabolite, depending on whether each single carbon atom position is a ^{12}C or ^{13}C isotope. These different types are called the isotopomers of a metabolite. Isotopomers are quantified by isotopomer fractions which must necessarily sum up to one, that is 100%, for each metabolite pool. The number of different isotopomers of all metabolites in the central metabolic pathways can be quite high, because metabolites with up to 12 carbon atoms occur. This gives rise to high dimensional equation systems that must be solved in MFA.

The mathematical centerpiece of MFA is given by the stoichiometric balance equations for the metabolic fluxes ($\boldsymbol{N}$ stoichiometric matrix; $\boldsymbol{v}$ flux vector)

$$\boldsymbol{N} \cdot \boldsymbol{v} = \boldsymbol{0} \tag{1}$$

together with the isotopomer labeling balance system [WMI$^+$99, SCNV97].

$$\overline{\boldsymbol{f}}(\boldsymbol{v}, \overline{\boldsymbol{x}}^{inp}, \overline{\boldsymbol{x}}) = \boldsymbol{0} \tag{2}$$

In this generally non-linear equation system the vector $\overline{\boldsymbol{x}}$ combines all isotopomer fractions of all metabolites in the networks. It can have $\dim \overline{\boldsymbol{f}} = \dim \overline{\boldsymbol{x}} > 5000$. The labeling state $\overline{\boldsymbol{x}}^{inp}$ of the substrates fed into the system is known while the flux vector $\boldsymbol{v}$ is to be determined from the experiment. Assuming $\boldsymbol{v}$ to be unknown the equations (1) and (2) allow to compute the labeling state $\overline{\boldsymbol{x}}$ of the system as a function of $\boldsymbol{v}$ (cf. Sec. 1.3).

1.2 Solution of Carbon Labeling Systems

For various reasons the solution of the high dimensional non-linear labeling balances with respect to $\overline{\boldsymbol{x}}$ must be achieved with a high computational efficiency:

1. The computation of the function $\overline{\boldsymbol{x}} = \overline{\boldsymbol{x}}(\boldsymbol{v})$ can be considered as a simulation of the labeling experiment where the metabolic fluxes $\boldsymbol{v}$ are known. Flux analysis essentially is the inverse problem to this simulation step, which means that a large number of simulation steps for different values of $\boldsymbol{v}$ must be done in a parameter fitting algorithm to match the measured data [WSGM97, WMPG01].

2. The statistical analysis of the estimated fluxes requires the computation of the flux sensitivities $\partial \overline{\boldsymbol{x}} / \partial \boldsymbol{v}$ [WMI$^+$99]. If this is done with high order finite difference methods a large number of simulation steps is required.

3. The determination of confidence regions for the estimated fluxes is absolutely necessary because biological measurement values are typically rather noisy. In the case of non-linear systems and non-normal error distributions it is possible to calculate

confidence regions by a *Monte-Carlo-Markov-Chain* Method which requires a large number of simulation steps [LW04].

4. MFA is applied to more and more complex systems in recent years. This leads to a growing dimension of the systems and the computational effort becomes critical.

1.3 Existing Solution Algorithms

Several numerical algorithms have been developed to solve the equation system (1) and (2). They can be classified into the following three approaches:

1. Iterative methods based on standard numerical iteration schemes generally produce the wanted results but their convergence can be slow and depends on the flux vector $\boldsymbol{v}$ [SCNV97, YWH04].

2. It could be shown in [WW01] that the state vectors $\overline{\boldsymbol{x}}$ and $\overline{\boldsymbol{x}^{inp}}$ can be linearly transformed in another pair of state vectors $\boldsymbol{x}$ and $\boldsymbol{x}^{inp}$ of the same dimension, in such a way that the transformed equations system has basically the same mathematical structure as Eq. (2).

$$\boldsymbol{f}(\boldsymbol{v}, \boldsymbol{x}^{inp}, \boldsymbol{x}) = \boldsymbol{0} \tag{3}$$

   The new state variables are called cumomer fractions and can be interpreted as labeled carbon atoms, pairs of carbon atoms, triples of carbon atoms and so on – that are traced through the network. Basically, the cumomer network has the same structure as the isotopomer network except for some differences described in [IW03]. However, these differences lead to a powerful mathematical reformulation of the balance equations. Precisely, the cumomer fraction vector can be partitioned into a sequence of vectors $\boldsymbol{x} = \left({}^0\boldsymbol{x}, {}^1\boldsymbol{x}, {}^2\boldsymbol{x}, \ldots\right)$ in such a way that Eq. (3) becomes a cascade of linear equation systems with non-linear inhomogeneous terms ${}^i\boldsymbol{b}$:

$$\begin{array}{rcll} {}^0\boldsymbol{x} & = & 1 & \\ \boldsymbol{0} & = & {}^i\boldsymbol{A}(\boldsymbol{v}) \cdot {}^i\boldsymbol{x} + {}^i\boldsymbol{b}\left(\boldsymbol{v}, \boldsymbol{x}^{inp}, {}^0\boldsymbol{x}, {}^1\boldsymbol{x}, \ldots, {}^{i-1}\boldsymbol{x}\right) & i = 1, 2, \ldots \end{array} \tag{4}$$

   This cascade directly suggests a non-iterative successive solution algorithm that solves one stage of the cascade after the other by e.g. the Gaussian algorithm. However, linear equation solving algorithms generally are of complexity $\mathrm{O}(n^3)$, so that this method becomes more and more time consuming with growing system dimensions [WMI+99].

3. Several authors tried to derive explicit formulas for the solution of the equation system which is possible from the cascade (4) [KPSS99]. It turns out that the explicit solutions are always rational functions of the fluxes [IW03]. However, since computer algebra systems were applied to find explicit solutions this was only possible for small or strongly reduced metabolic networks whereas complex networks lead to a tremendous computational effort.

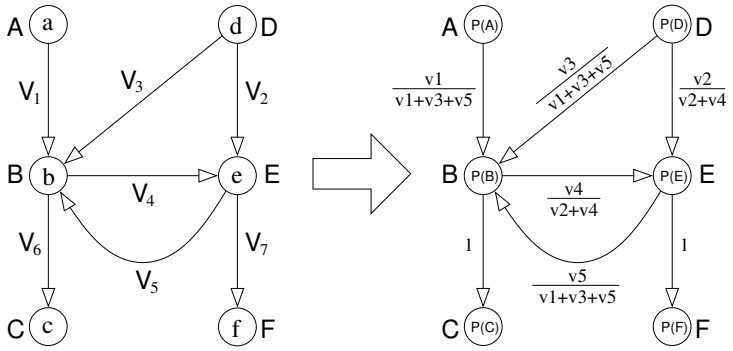

Figure 1: left: an example linear flux network; right: transformation of fluxes into *influx probabilities*

This contribution presents a new algorithm for solving the balance equations which is of type 3, but is able to produce a complete explicit solution in a recursive form without any restrictions on the network size. The new algorithm is based on a complete path tracing of labeled particles through the metabolic network. This is also the way a biologist deals with CLS and thus is close to the way of thinking in biochemistry. However, the attempt to derive a solution algorithm from this path tracing intuition has not been successful up to now. The reason is that many cycles occur in the metabolic network and path tracing involves an infinite number of possible paths that connect the input substrates of the system with any other pool. A solution can only be successful if this infinite set of paths can be represented in a finite way. It will turn out in the following, that this finite representation is directly being achieved by the application of KLEENE's theorem using the classical algorithm for the derivation of the regular expression from a finite automaton.

2 An illustrative Example

2.1 Balance Equations

Figure 1 shows the graph of a simple linear flux network. All metabolites have just one carbon atom and therefore all reactions between metabolites are mono-molecular, which reduces cascade (4) to a single stage – nevertheless a representative example, because the stages of the real cascade are made up likewise. Each node corresponds to a set of carbon atoms, called a *pool*, in which each atom can be characterized as being labeled (^{13}C) or unlabeled (^{12}C). Each carbon atom pool can be described by the percentage of labeled carbon atoms. The directed edges in the graph correspond to chemical reactions between the different metabolite pools. In the following, a sans-serif upper case letter (e.g. A) stands for a node, and a lower case letter (e.g. a) denotes the corresponding labeling fraction. Edges i are labeled with a V_i whereas the reaction's flux value is denoted by v_i.

Based on the graph in Fig. 1, the stoichiometric equations (5, left) are formulated first.

They describe the underlying system in a metabolic stationary state. Their predication is that the overall in- and out-flux of every single metabolite pool, excluding input- and output pools, have to compensate.

$$\begin{array}{lrcl} \mathsf{B}: & v_1 + v_3 + v_5 & = & v_4 + v_6 \\ \mathsf{E}: & v_2 + v_4 & = & v_5 + v_7 \end{array} \Rightarrow \begin{array}{rcl} v_1 a + v_3 d + v_5 e & = & (v_4 + v_6)\, b \\ v_2 d + v_4 b & = & (v_5 + v_7)\, e \end{array} \tag{5}$$

Given the labeling fractions in the input-pools A and C it is now possible to quantify the stationary ^{13}C labeling fractions of the inner pools. Because of the linearity of the example network every labeling in- and outflux can be described as a product of a labeling fraction and a flux value. The flux value describes the total transported material (^{12}C and ^{13}C) and the labeling fraction the percentage of the involved ^{13}C atoms. With help of the stoichiometry this leads to the labeling balance equations (5, right). With knowledge of the flux values, which have to satisfy the stoichiometry (5, left), a linear system of equations (6) can be obtained from which the unknown fractions *b, e* in the nodes B and E can be computed (note that $c = b$ and $f = e$):

$$\begin{pmatrix} v_4 + v_6 & -v_5 \\ -v_4 & v_5 + v_7 \end{pmatrix} \cdot \begin{pmatrix} b \\ e \end{pmatrix} = \begin{pmatrix} v_1 a + v_3 d \\ v_2 d \end{pmatrix} \tag{6}$$

This linear system can be solved by any common method like the GAUSS algorithm. However, we here introduce a new approach and compare both approaches in section 4.

2.2 From Flux to Flux Probability

The basis for path tracing is a change of viewpoint from fluxes to *flux probabilities*. The idea comes from Eq. (5, right) by solving the equations for the balanced labeling fraction. Pool B for example yields:

$$b = \frac{v_1}{v_4 + v_6}\, a + \frac{v_3}{v_4 + v_6}\, d + \frac{v_5}{v_4 + v_6}\, e = \frac{1}{v_1 + v_3 + v_5}\, (v_1\, a + v_3\, d + v_5\, e) \tag{7}$$

If the common denominator $v_4 + v_6$ is replaced with the equivalent expression $v_1 + v_3 + v_5$ from Eq. (5), the labeling fraction b of node B is solely described by the incoming flux values and the labeling fractions carried along. A physical interpretation for the occurring quotients are the probabilities that a labeled carbon atom in pool B originates from one of the pools A, D or E. These *influx probabilities* can also be understood as conditional probabilities. Under this perspective it is convenient to interpret the labeling fractions $a, b, d, e \in [0, 1]$ as probabilities $P(\mathsf{A})$, $P(\mathsf{B})$, $P(\mathsf{D})$, $P(\mathsf{E})$, as well. This leads to an alternative formulation for Eq. (7):

$$P(\mathsf{B}) = P(\mathsf{A}) \cdot P(\mathsf{A}|\mathsf{B}) + P(\mathsf{D}) \cdot P(\mathsf{D}|\mathsf{B}) + P(\mathsf{E}) \cdot P(\mathsf{E}|\mathsf{B}) \tag{8}$$

By using this simple procedure, every balance equation can be converted into an equation of probabilities. Equation (8) resembles the equations customary in context of the *Markov Chain* methods and is known as the law of complete probability.

2.3 Regular Expressions describe Paths

In preparation for path tracing all fluxes need to be transformed into *influx probabilities*, as shown in Fig. 1. From now on, paths will be described as *edge-words* – i.e. words, whose symbols are edges of the network graph (e.g. $\mathsf{V}_1\mathsf{V}_4\mathsf{V}_5\mathsf{V}_6$). Interpreting the discussed graphs as state transition graphs of finite state machines, certain sets (regular languages over the alphabet given by the set of edges) of these paths can be compactly represented as regular expressions. An example is the infinite set of the paths from A to C, which can be represented by the regular expression $\mathsf{V}_1(\mathsf{V}_4\mathsf{V}_5)^\star\mathsf{V}_6$. If R is such a regular expression then $L(R)$ is the set of paths generated by R. By abuse of notations, the L will be omitted in the following so that the regular expression R directly represents the set of paths it generates.

To determine a labeling fraction of a yet indeterminate pool all arriving reaction paths (including their probabilities), coming from the input pools (A and D in the running example), must be known. For instance, to determine the labeling fraction e of pool E the following paths have to be considered (amongst others):

$$\begin{array}{ll} \mathsf{A} \to \mathsf{E}: & \mathsf{V}_1\mathsf{V}_4 \\ \mathsf{D} \to \mathsf{E}: & \mathsf{V}_2,\ \mathsf{V}_3\mathsf{V}_4 \end{array} \tag{9}$$

The probability to walk through the graph along a certain path (e.g. $\mathsf{V}_1\mathsf{V}_4$) equals the product of the single step probabilities. Correspondingly, a probability can be assigned to each path:

$$\begin{array}{lcl} P(\mathsf{V}_1\mathsf{V}_4) = P(\mathsf{V}_1)\cdot P(\mathsf{V}_4) & = & P(\mathsf{A}|\mathsf{B})\cdot P(\mathsf{B}|\mathsf{E}) \\ P(\mathsf{V}_2) & = & P(\mathsf{D}|\mathsf{E}) \\ P(\mathsf{V}_3\mathsf{V}_4) = P(\mathsf{V}_3)\cdot P(\mathsf{V}_4) & = & P(\mathsf{D}|\mathsf{B})\cdot P(\mathsf{B}|\mathsf{E}) \end{array} \tag{10}$$

The paths V_2 and $\mathsf{V}_3\mathsf{V}_4$ are *parallel*, since they describe different ways to reach node E, starting at node D (cf. Fig. 1). Therefore, the joint probability to reach node E via one or the other path equals the sum of probabilities of both paths:

$$P(\mathsf{V}_2 + \mathsf{V}_3\mathsf{V}_4) = P(\mathsf{V}_2) + P(\mathsf{V}_3\mathsf{V}_4) \tag{11}$$

Here, the different meanings of the '+'-operator must not be confused: on the left is signifies a set union of regular languages, while on the right it really means the addition of two probability values.

In this example the complete set of paths not only consists of the three mentioned paths, but is in fact infinite. This is due to of the cycle through nodes B and E, i.e. paths can loop arbitrary often before they finally end in node E. Exactly as in the context of the traditional regular expressions this infinite set of paths can be described compactly by help of the KLEENE-operator $(\star)$. The complete set of paths arriving in node E are:

$$\begin{array}{l} \textbf{paths } \mathsf{D} \to \mathsf{E}: \\ \{\mathsf{V}_2,\ \mathsf{V}_2\mathsf{V}_5\mathsf{V}_4,\ \mathsf{V}_2\mathsf{V}_5\mathsf{V}_4\mathsf{V}_5\mathsf{V}_4,\ \ldots\} \cup \{\mathsf{V}_3\mathsf{V}_4,\ \mathsf{V}_3\mathsf{V}_4\mathsf{V}_5\mathsf{V}_4, \mathsf{V}_3\mathsf{V}_4\mathsf{V}_5\mathsf{V}_4\mathsf{V}_5\mathsf{V}_4,\ \ldots\} \\ \qquad \equiv \mathsf{V}_2(\mathsf{V}_5\mathsf{V}_4)^\star + \mathsf{V}_3(\mathsf{V}_4\mathsf{V}_5)^\star\mathsf{V}_4 \equiv (\mathsf{V}_2 + \mathsf{V}_3\mathsf{V}_4)(\mathsf{V}_5\mathsf{V}_4)^\star \\ \textbf{paths } \mathsf{A} \to \mathsf{E}: \\ \qquad \{\mathsf{V}_1\mathsf{V}_4,\ \mathsf{V}_1\mathsf{V}_4\mathsf{V}_5\mathsf{V}_4,\ \cdots\} \equiv \mathsf{V}_1(\mathsf{V}_4\mathsf{V}_5)^\star\mathsf{V}_4 \end{array} \tag{12}$$

As mentioned before, the probability values of parallel paths have to be added. Therefore the infinite set of edge-words described by a regular expression over the set of edges (from now on called *regular path* for short) containing the KLEENE-operator generates an infinite sum of probabilities. Example:

$$\begin{aligned} P\left(\mathsf{V}_2(\mathsf{V}_5\mathsf{V}_4)^\star\right) &= \sum_{\varphi\in\mathsf{V}_2(\mathsf{V}_5\mathsf{V}_4)^\star} P(\varphi) = P(\mathsf{V}_2)\cdot\sum_{\varphi\in(\mathsf{V}_5\mathsf{V}_4)^\star} P(\varphi) \\ &= P(\mathsf{V}_2)\cdot\sum_{i=0}^{\infty}\left[P(\mathsf{V}_5)P(\mathsf{V}_4)\right]^i = \frac{P(\mathsf{V}_2)}{1-P(\mathsf{V}_5)P(\mathsf{V}_4)} \end{aligned} \tag{13}$$

Here, the cycle denoted by the KLEENE-operator generates an infinite geometric series of which probability value can be expressed compactly by the corresponding series value.

3 General Path Tracing Algorithm

3.1 Path Evaluation Rules

Summarizing, let $G=(V,E,P)$ be a graph of a flux network labeled with probabilities. Let V be a set of nodes (=pools), $E\subseteq V\times V$ a set of directed edges, $\mathsf{X},\mathsf{Y}\in V$ and Q,R regular expressions over an alphabet E. Let $P: E\mapsto[0,1]$ be the probability value assigned to every edge. Then the ability to evaluate an arbitrary regular path into a probability value follows by the recursive rules given in Tab. 1. In particular, the definitions

Table 1: recursive rules for evaluation of regular paths

regular path type	**evaluation rule**		
reg. paths of length one (edges)	$P(\mathsf{V_i})$	:=	$P(\mathsf{X}\vert\mathsf{Y})$ with $\mathsf{V_i}=(\mathsf{X},\mathsf{Y})\in E$
concatenated regular paths	$P(QR)$	:=	$P(Q)\cdot P(R)$
parallel regular paths	$P(Q+R)$	:=	$P(Q)+P(R)$
cyclic regular paths	$P(R^\star)$	:=	$1\,/[1-P(R)]$

give a finite, recursive method to evaluate any regular path R by $P(R)=\sum_{\varphi\in R}P(\varphi)$. For example, the evaluation of the regular paths in (12) yields in an explicit solution for the labeling fraction of node E with a clear discrimination of the labeling proportions contributed from the input pools A and D:

$$P(\mathsf{E}) = P(\mathsf{A})\cdot\frac{P(\mathsf{V}_1)P(\mathsf{V}_4)}{1-P(\mathsf{V}_5)P(\mathsf{V}_4)} + P(\mathsf{D})\cdot\frac{P(\mathsf{V}_2)+P(\mathsf{V}_3)P(\mathsf{V}_4)}{1-P(\mathsf{V}_5)P(\mathsf{V}_4)} \tag{14}$$

3.2 An Implementation based on the Kleene Algorithm

An implementation of the sketched method has to compute regular expressions for all paths between nodes with known labeling fraction and nodes with unknown labeling fraction. A priori, the only known labeling fractions belong to the input nodes. The straight forward method to determine a node's yet unknown labeling fraction is to evaluate the regular expressions representing the sets of paths coming from the different input nodes.

The problem of determining a regular expression representing a set of paths between two nodes can solved by the KLEENE algorithm, which was presented in 1956 within S.C. KLEENE'S famous proof on the equivalence of regular languages and finite state machines. The algorithm was generalized by MCNAUGHTON and YAMADA in 1960. The concept of an algebraic structure called *closed semiring* was formulated, which can be seen not only as the underlying principle for the KLEENE algorithm, but also for the widely known graph algorithms by WARSHALL (reflexive and transitive hull, 1960) and FLOYD (all-pair shortest path, 1960) [AHU74].

```
1  -- initialization of the adjacency matrix V^0, N = |V|
2  ∀i,j ∈ {1,...,N} : V^0_ij ← { P(V_k)   iff V_k = (i,j) ∈ E
                               { 0        otherwise
3  -- tracing of all paths / calculation of a transitive, reflexive hull
4  for k in 1..N loop
5      for i,j in 1..N loop
6          V^k_ij ← P ( V^{k-1}_ij + V^{k-1}_ik ( V^{k-1}_kk )* V^{k-1}_kj )
7      end loop
8      V^k_kk ← V^k_kk + 1
9  end loop
10 -- Evaluation of labeling fractions
11 for j ∈ (V − inputnodes) loop
12     P(j) ← 0
13     for i ∈ inputnodes loop
14         P(j) ← P(j) + P(i) · V^N_ij
15     end loop
16 end loop
```

The variant of the algorithm presented here uses an important modification for non-idempotent closed semirings[1], suggested in [Fl80]. This modification includes a simplified initialization as adjacency matrix in line 2 and the update step in line 8. Line 6 performs an *eager evaluation* of the probability value, so that there is no need to store the regular path expression itself. In lines 11–16 all unknown labeling fractions are evaluated by using the law of complete probability.

[1] i.e. closed semirings with non-idempotent addition operator; used for the *addition* of probability values of parallel paths in this context

4 Discussion and Future Work

A prototypical implementation of the modified KLEENE algorithm gives correct results for example networks of realistic size. Moreover a formal proof (not presented here) guarantees the convergence of the generated series expansions. The following points emphasize the properties of the new approach:

A "natural" approach Solving a CLS by path tracing is a very natural approach compared to the task of interpretation of the GAUSS algorithm's abstract results. The presented method describes the formation of labeling fractions by the superposition of regular paths between pools which corresponds better with the biologist's association.

Comparison of path tracing and Gauss algorithm The drawback of the usage of a conventional, matrix based solving algorithm is usually running time. In case of the GAUSS algorithm the running time is of order $O(n^3)$, where n is the number of unknown quantities. In real networks n can be > 5000. Clearly, the algorithm presented in the last section has an asymptotic lower bound of running time $\Omega(n^3 + n)$, which is slightly worse than a common implementation of the GAUSS algorithm using the *full pivoting* scheme: $\Omega(n^3/3 + 3n^2/2 + n/6)$. However, in the discussed application the KLEENE-algorithm does some superfluous work because it constructs regular path expressions between *all* pairs of nodes in the graph – also between inner nodes, and inner & output nodes – which are needed during the processing in lines 4–9 but not for the evaluation step in lines 11–16.

Exploiting network topology The application of the KLEENE algorithm can be considered as prototypical. In fact, we can expect that any algorithm that converts sets of paths into regular expressions can be used – as long as no redundant paths are added to the expressions. The computation steps performed by the KLEENE algorithm are independent from the underlying graph's topology. Metabolic networks are rather loosely connected and pools are usually fed by a small number of fluxes. This gives the hint that an algorithm that starts at the input pools, visiting pools in a breadth-first-search order should be much faster. It can easily be shown that cycle-free parts in networks, where each pool is fed by a small number other pools, can be processed in linear time – whereas network parts with the maximum number of cycles, where each pool is fed by every other pool, lead to at most $O(n^3)$ computation steps. $O(n^3)$ can be seen as a worst case upper bound and we are hopeful much more efficient algorithms can be formulated.

Code Generation An interesting application of the KLEENE algorithm is code generation: the above algorithm can be used to generate sequential code which can be compiled on-the-fly. The compiled code can be used as a plug-in for the parameter fitting algorithm. The resulting speed-up should be significant, since the compiled code is free of branches and consists solely of floating point instructions. This method takes advantage from modern superscalar processor architecture and speculative execution.

Numerical robustness Solving a CLS using the GAUSS algorithm involves a large number of row transformation steps to obtain a triangular matrix suitable for back-substitution. Especially for large systems this procedure leads to an accumulation of numerical errors in the lower-right matrix elements. In the following back-substitution steps the accumulated errors are distributed over the results. The KLEENE algorithm does not involve

any pre-structuring of the system. Thus, it prevents the primary source of numerical errors of the GAUSS algorithm. Solutions are obtained by successive insertion of known quantities coming from topological predecessors. First experimental results confirm this intuition but detailed analysis has to done on this topic.

References

[AHU74] Aho, A. V., Hopcroft, J. E., and Ullman, J. D.: *The Design and Analysis of Computer Algorithms*. Comp. Sci. and Inf. Proc. Addison-Wesley Publishing. 1974.

[Fl80] Fletcher, J. G.: A more general algorithm for computing closed semiring costs between vertices of a directed graph. *Communications of the ACM*. 23(6). 1980.

[IW03] Isermann, N. and Wiechert, W.: Metabolic isotopomer labeling systems. Part II: structural flux identifiability analysis. *Mathematical Biosciences*. 183:175–214. 2003.

[KPSS99] Klapa, M., Park, S., Sinskey, A., and Stephanopoulos, G.: Metabolite and isotopomer balancing in the analysis of metabolic cycles: I theory. *Biotechnology and Bioengineering*. 62:375–391. 1999.

[LW04] von Lieres, E. and Wiechert, W.: Bayesian statistics and markov chain monte carlo simulation: An alternative method for parameter identification and error estimation. In: *EUROSIM 2004, Paris*. 6.–10. September 2004.

[SCNV97] Schmidt, K., Carlsen, M., Nielsen, J., and Villadsen, J.: Modelling isotopomer distribution in biochemical networks using isotopomer mapping matrices. *Biotechnology and Bioengineering*. 55:831–840. 1997.

[Sz98] Szyperski, T.: ^{13}C-NMR, MS and metabolic flux balancing in biotechnology research. *Quart. Rev. of Biophysics*. 31:41–106. 1998.

[W02] van Winden, W.: *^{13}C labeling technique for metabolic network and flux analysis*. PhD thesis. Kluver Laboratory for Biotechnology: TU Delft. 2002.

[Wi01] Wiechert, W.: ^{13}C metabolite flux analysis. *Metab. Eng.* 3:195–206. 2001.

[WMI+99] Wiechert, W., Möllney, M., Isermann, N., Wurzel, M., and de Graaf, A. A.: Bidirectional reaction steps in metabolic networks. Part III: Explicit solution and analysis of isotopomer labeling systems. *Biotechnology and Bioengineering*. 1999.

[WMPG01] Wiechert, W., Möllney, M., Petersen, S., and de Graaf, A.: A universal framework for ^{13}C metabolic flux analysis. *Metabolic Engineering*. 3:265–283. 2001.

[WSGM97] Wiechert, W., Siefke, C., de Graaf, A. A., and Marx, A.: Bidirectional reaction steps in metabolic networks. Part II: Flux estimation and statistical analysis. *Biotechnology and Bioengineering*. 55(1):118–135. 1997.

[WW01] Wiechert, W. and Wurzel, M.: Metabolic isotopomer labeling systems. Part I: Global dynamic behaviour. *Mathematical Biosciences*. 169:173–205. 2001.

[YWH04] Yang, T. H., Wittmann, C., and Heinzle, E. E.: Metabolic network simulation using logical loop algorithm and jacobian matrix. *Metabolic Engineering*. 7, in press. 2004.