CONVEX ANALYSIS OF METABOLIC NETWORK FOR OPTIMAL CELL DESIGN & FLUX VALIDATION BY GC-MS OR NMR

by

Tao Zhu

B.S. in Biological Science and Biotechnology, Tsinghua University, China, 1995

M.S. in Chemical Engineering, Tsinghua University, China, 1998

Submitted to the Graduate Faculty of

School of Engineering in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2003

UNIVERSITY OF PITTSBURGH

SCHOOL OF ENGINEERING

This dissertation was presented

by

Tao Zhu

It was defended on

December, 2003

and approved by

Karl Johnson, Associate Professor, Chemical and Petroleum Engineering Department

Saleem Khan, Professor, Molecular and Genetics and Biochemistry Department

Richard Koepsel, Associate Professor, Chemical and Petroleum Engineering Department

Robert Parker, Assistant Professor, Chemical and Petroleum Engineering Department

Michael M. Domach, Professor, Chemical Engineering Department, CMU

Dissertation Director: Mohammad M. Ataai, Professor, Chemical and Petroleum Engineering Department

ABSTRACT

CONVEX ANALYSIS OF METABOLIC NETWORK FOR OPTIMAL CELL DESIGN & FLUX VALIDATION BY GC-MS OR NMR

Tao Zhu, PhD

University of Pittsburgh, 2003

In this thesis an integrated software tool: MetaboLogic was developed. This software not only integrates most of the techniques developed in Metabolic Flux Analysis (MFA), but also two mathematical techniques for finding multiple optimal solution and designing tracer experiments. MetaboLogic is designed to allow users to construct arbitrary network model visually through a friendly graphic user interface.

The functions of the software can be classified into two groups. The first group is to compute flux distribution by using stoichiometric information. This part includes traditional linear programming techniques and the modified simplex algorithm we developed to find the alternative optimal solutions. The second group of functions deals with the mathematical methods to simulate NMR and GC/MS spectra based on flux distributions. The multiple solutions found using MILP or modified simplex method can be used to better design the tracer experiment in terms of choice of labeled substrates and signal molecules.

We applied MetaboLogic to find a potential metabolic engineering strategy for inhibiting the acid formation. Acid formation is a major problem in production of recombinant protein in both E. coli and B. subtilis, because it limits process stability and cell concentration and thus cell-based biotechnological processes. The inactivation of pyruvate kinase (PYK) was identified as one potential metabolic engineering strategy for eliminating acidic by-products. PYK mutants were constructed and characterized in terms of growth and acid formation. The experimental results confirmed that the predicted strategy is an effective way to reduce acid formation. This application is good demonstration of the MetaboLogic's capabilities.

Finally, the MetaboLogic was used for the design of genetic-based strategies for enhancing folic acid production in E. coli and B. subtilis. The genetic strategy that emerged for reduction of acid formation (pyk mutation) was found to be a very promising start point for increased folic acid production. The experimental data in E. coli confirmed that pyk mutation increased the folic acid production by 5-6 fold compared to the wild type.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Page

1.0 SPECIFIC AIMS AND OVERVIEW

Metabolic Flux Analysis (MFA) has become a widely used approach for studying the properties and capabilities of metabolic networks in microorganisms (Stephanopoulos et at. 1995). MFA may be used for the determination of steady state metabolic flux distributions, if measurement of uptake and/or excretion rates of a cell culture in steady state is available. Combined with tracer experiment data, MFA can also be used to determine the flux distribution even if there are not enough uptake/execration data available. These predicted distributions provide information on both product yield horizons as well as suggest how the regulation of existing and inserted pathways needs to be altered in order to attain the yield horizon.

Various mathematical and experimental techniques are developed for prediction of flux distributions that provide high product yield. These methods include linear programming (Majewski & Domach, 1991), pathway analysis (Schilling et at. 2000), elementary mode analysis (Schuster et at. 1999), NMR and GC/MS simulation (Wiechert et at. 1997). Software are developed which integrates most of the mathematical methods (E-Cell Tomita, 2001; Gepasi, Mendes 1997; Glaser, 1999; Metatool (Pfeiffer et al. 1999); Metabolic Explorer, http://www.simtec.mb.uni-siegen.de/Software/ 13CFlux/).

One issue that has arisen with metabolic engineering models and software (Schaff et al., 1997; Goryanin et al., 1999; NSF Workshop on Metabolic Engineering, Sept. 2000) is that new and powerful software and modeling developments are challenging to adopt by potential users. It can be problematic to unravel the underlying assumptions encoded in a program or adapt the model for use in another problem. Also, there is a lack of a software tool which integrates all these techniques in a comprehensive manner. The user has to access different software if they want to apply different analysis methods to the same problem. The challenges may compound when the non-expert user desires to use a model or software as an input for another analysis activity.

There is another issue in the metabolic network modeling work. Two important mathematical methods are not well implemented in any software yet. There is no software package of metabolic flux analysis that can find all alternative solutions for a degenerate LP problem, which commonly arises in metabolic flux analysis. Furthermore, there is no software can validate flux distribution found by NMR or GC-MS spectra in any arbitrary network model.

1.1 Specific Aims

The first aim in this research is to develop an integrated software tool: *MetaboLogic*. This software shall not only integrate most of the techniques developed in MFA, but also two mathematical techniques for finding multiple optimal solution and the design of radio labeled pattern for flux confirmation. *MetaboLogic* is designed to allow users to construct arbitrary network model visually through a friendly graphic user interface.

We adopted MATLAB ® (Mathworks, Inc; www.mathworks.com) as the computational engine, which offers the possibility to directly benefit from predefined functions of MATLAB. Visual C++ and Microsoft Foundation Class (MFC) library were used to develop the user interface. This is a convenient choice because many libraries exist in C++ which expedites the developmental process. Many advanced user interface features are designed to increase the user friendliness of software significantly.

The functions of the software can be classified into two groups. The first group is to compute flux distribution by using stoichiometric information. This part includes traditional linear programming techniques and the modified simplex algorithm we developed to find the alternative optimal solutions. There are several algorithms available for finding alternative optimal solution of a degenerate LP problem, such as the MILP approach that our group proposed in earlier work (Lee et al. 2000). The modified simplex method was developed instead of using MILP because MATLAB doesn't have MILP solver. Moreover, the modified simplex method is efficient in handling a LP problem with many equality constraints. In next chapter, this algorithm will be discussed in detail.

The second group of functions deals with the tracer experimental design and data analysis. The multiple solutions found using MILP or modified simplex method can be used to better design the tracer experiment in terms of choice of labeled substrates and signal molecules. Once the experiment is done, software can validate the flux distribution by matching the simulated NMR or GC-MS spectra with the actual data. NMR and GC-MS spectra simulations are both based on the isotopomer distribution calculations. The tracer experiment should maximize the difference among spectra simulated from each extreme solution. The determination of actual flux distribution is to find the combination coefficient of extreme solutions that closely resembles the observed spectra.

The second aim is to study the acid formation in *E. coli* and *B. subtilis* in a glucose minimal medium. Acid formation is a major problem in production of recombinant protein in both *E. coli* and *B. subtilis*, because it limits process stability and cell concentration and thus cell-based biotechnological processes (Ko, et al. 1993, Shiloach et al. 1996). We applied *MetaboLogic* to find a potential metabolic engineering strategy for inhibiting the acid formation. The inactivation of pyruvate kinase (PYK) was identified as one potential metabolic engineering strategy for eliminating acidic by-products. PYK mutants were constructed and characterized in terms of growth and acid formation. The experimental results confirmed that the predicted strategy is an effective way to reduce acid formation. This application is a good demonstration of the *MetaboLogic's* capabilities.

The third aim in this research is to design genetic-based strategies for enhancing folic acid production and to experimentally test some strategies. We applied *Metabologic* in the calculation of the folic acid production capacity of different strains. The genetic strategy that emerged for reduction of acid formation (pyk mutation) proved to be a very promising start point for increased folic acid production. The experimental data confirmed that pyk mutation increased the folic acid production by 5-6 fold compared to the wild type.

1.2 Thesis Outline

- Chapter 2 presents the mathematical background of convex analysis and the modified three-phase simplex algorithm to find the multiple solutions. A stoichiometric model of *E. coli* and *B. subtilis* central carbon metabolism is constructed. When applied with the convex analysis algorithm, the model predicts that PYK mutation is a good strategy for reducing acid formation.
- Chapter 3 explores NMR spectra simulation and searches for operative flux distribution in the convex polytope found by the modified simplex algorithm. This chapter also presents a strategy for the design of an effective NMR and GC-MS experiment.
- Chapter 4 presents the features of the user-friendly computer software, *Metabologic*, which integrated the convex analysis and the NMR spectra simulation.
- Chapter 5 discusses experiment results on acid formation in both *E. coli* and *B. subtilis.* The experimental data confirmed the model prediction that was made in chapter 2.
- Chapter 6 applies the convex analysis to optimize the theoretical folic acid production and evaluate folic acid production in *E. coli* batch fermentation.

In the thesis, there is no separate chapter about material and methods for the experiment, because most of experimental work is done in chapter 5 and 6 only. The experimental methods will be covered in those chapters. Computer codes for the convex analysis appear in Appendix A. The isotopomer mapping matrix method is reviewed in Appendix B. The source code of the software is not listed as appendix due to its length (about 70000 lines), but it has been archived on a compact disk.

2.0 MODIFIED SIMPLEX METHOD TO FIND ALTERNATIVE FLUX DISTRIBUTIONS

2.1 Introduction

Finding alternative yet feasible flux distributions in a metabolic network is the first task in designing a metabolic engineering strategy aimed at enhanced product yield. Many mathematical methods have been developed to solve this problem. Linear Programming has been one of the most important tools used in analysis of the productive capabilities of metabolic networks (Fell & Small, 1986; Majewski & Domach, 1990; Varma & Palsson, 1994a). Typically, flux balance equations are formulated and local and system-wide constraints are imposed. Examples of local and system-wide constraints are inclusion of an upper bound enzymatic capacity or that an ATP requirement for biosynthesis must be met through the aggregate activity of a subset of reactions. An objective is then posed such as maximization of a particular flux that leads to high product yield. The answer informs one of the yield horizons. Moreover, the values of the fluxes in the optimal solution indicate how to engineer the trafficking of metabolites by altering the available reaction paths and feedback loops.

While very useful, the system of equations and constraints is often underdetermined, which means more than one solution may exist that could satisfy the objective. These other solutions are not always immediately accessible to the analyst because classical solution methods such as the Simplex Algorithm typically stop after one optimal flux solution is found. One must inspect the details of the intermediate calculations to discover that multiple solutions exist and then the problem must be manually reformulated to ferret them out.

The identification of other solutions is of interest for several reasons. First, the different solutions may differ in how easy it is to implement them. For example, one solution may indicate more genetic manipulation is required than another that could yield the same value of the objective. Second, there is fundamental value in knowing how many alternatives exist. This provides some insights on how redundant and robust a metabolic system can be. Finally, knowing the portfolio of alternatives ahead of time can result in the development of a more inclusive and tighter patent strategy.

Recent work in metabolic engineering has drawn from the network analysis field to address the problem of enumerating alternate metabolite trafficking solutions. Lee et al (2000) and Phalakornkule et al (2001) developed an alternative approach to find all the alternate optima in LP models that entailed transforming the master problem into a mixed integer linear programming (MILP) problem. Phalakornkule et al (2001) used different solutions from LP analysis to design 13C NMR experiments such that different potential flux patterns in a strain can be more conclusively distinguished.

To find the alternative solutions of an arbitrary network using *MetaboLogic*, we have utilized some additional features of convex space analysis to find multiple solutions. In this chapter, we first describe the basic idea about how the linear programming problem can be converted to a convex hull searching problem. Then we will describe new algorithm modified from two phase simplex method to find the flux space. Then this algorithm is used to calculate flux distribution in *B. subtilis* and *E. coli* under different culture conditions. The comparison between the new algorithm and MILP algorithm will be given at the end of the chapter.

2.2 Background on Convex Analysis

A set X in E^n is called a convex set if given any two points x_1 and x_2 in X, then $\lambda x_1 + (1-\lambda) x_2 \in X$ for each $\lambda \in [0,1]$. A convex set can be bounded or unbounded.

Theorem 1: A set S of feasible solutions of an LP is a convex set.

The convex analysis of metabolic network entails finding the set of minimum basic vectors spanning the convex space that satisfy mass balance equations for all metabolites. The set (S) can be denoted as

$$S = \left\{ x : x = \sum_{i}^{k} a_i e_i, \quad a_i \geq 0 \right\} \tag{1}$$

where x is any possible vector that satisfies the constraints. In Equation (1), e_i (i=1, 2…k), k, and a_{i} refer to the minimum solution set that spans the solution space, the number of basic vectors, and a combinational coefficient.

When the solution space is bounded, the convex set is a polytope, otherwise it constitutes a polyhedron. Each flux in the polytope can expressed as a convex combination of base vectors by

$$S = \left\{ x : x = \sum_{i}^{k} a_i e_i, \quad a_i \geq 0, \quad \sum_{i}^{k} a_i = 1 \right\} \tag{2}$$

A linear programming problem then is equivalent to a problem to find such a set of e_i. A number of special purpose algorithms have been reported for finding all the extreme points of convex polytope (Matheiss & Rubin, 1980). Lee (2000) developed a recursive MILP method, which utilizes the MILP solver that is already implemented in a modeling language (e.g. GAMS, AMPL). Since we choose MATLAB as our computational engineering, which doesn't have a MILP solver, the MILP approach doesn't satisfy our requirement.

It is noticed that in a convex analysis problem of metabolic network, many of the constraints are equality constraints (mass balance constraints). Although the system has many variables (e.g. reaction rates), the total degree of freedom of the system is usually low (less than 10). We have developed an efficient algorithm that takes advantage of these properties. The algorithm is based on a two-phase simplex method (Hartley, 1985). Unlike the traditional simplex method, which stops at one optimal solution, the new proposed algorithm will continue to find all the other extreme points that have the same value of objective function. This algorithm can also be named as three-phase simplex. To search the neighboring vertices we adopted depth first search algorithm (Sedgewich, 2001), which is commonly used in graph search in computer science.

2.3 Algorithm

Simplex method starts from an arbitrary vertex, searching the adjacent vertices to find the one that optimizes objective function locally. It then moves to the new vertex and repeats the searching again. It stops when all the adjacent vertices will not improve the objective function. But there might be some adjacent vertices that have the same value of the objective function, if the LP problem is degenerate. We add the third phase (depth first search) to the commonly used two-phase simplex method to find all the alternative optimal vertices. A detailed description of algorithm appears in Appendix A.

The algorithm is implemented in MATLAB language (Appendix B). An example problem is built up to test the algorithm. In this problem, there are 69 inequality constraints and 28 equality constraints and 34 variables. The solutions found by our algorithm are identical to the MILP methods. To evaluate the speed of the algorithm, we deleted some of the equality constraints so that more alternative optimal solutions exist and record the time needed to find all these solutions. We found the quadratic dependency of time on the total number of solution (Figure 2-1). Since number of solution is exponentially depending on the system freedom number in the worst case, the computational time to find all solutions is then also exponentially depending on system freedom number in worst case, which is same to MILP approach.

Compared with MILP and other available algorithms, our algorithm has several features. First, the computation time of this algorithm is second order with respect to the number of optimal vertices, yet still exponentially to the system number of freedom in the worst case. Secondly, this algorithm can be implemented easily and it does not require a MILP solver.

Thirdly, it is efficient when the system has a large number of equality constraints, which is common for flux balance analysis. This is because the computational time is exponential to the system degree of freedom, which is usually small in metabolic flux analysis. The disadvantage of our algorithm is that the first step of algorithm may not succeed in all cases. Simplex methods may fail because of infinitely looping. One way to overcome this problem is to adopt another LP method to first find the optimum value of the objective function. Then the modified simplex method can be used to find all feasible solutions. In this approach, it will ensure that the searching can be finished in a finite time.

2.4 Application of Algorithm: Solution Space of *E. coli* and *B. subtilis*

2.4.1. Stoichiometric Model of Glucose Metabolism in Bacteria

The metabolic networks of *E. coli* and *B. subtilis* as well as candidate reversible reactions are shown in Figure 2-2. The networks are based on prior representations (Lee et al, 2000; Phalakornkule et al, 2001; Sauer et al, 1998). The only difference between the *B. subtilis* and *E. coli* pathways is that *B. subtilis* synthesizes oxaloacetate from pyruvate instead of PEP.

The 16 molar balances about intracellular metabolite are represented by equations (1a) - (1p), which have been previously described (Lee et al, 2000; Phalakornkule et al, 2001).

$$r_1 - r_2 - r_3 - r_{10} = 0 \tag{1a}$$

$$r_{10} - r_{12} - r_{11} + r_7 + r_8 = 0 \tag{1b}$$

$$r_{14} - r_{15} - r_{16} = 0 \tag{1c}$$

$$-r_1 + r_{16} - r_{18} - r_{17} - r_{31} = 0 \tag{1d}$$

$$r_1 + r_{18} - r_{21} - r_{19} - r_{20} + r_{32} = 0 \tag{1e}$$

$$-r_{31} + r_{24} + r_{30} - r_{33} = 0 \tag{1f}$$

$$r_{21} - r_{22} - r_{23} - r_{24} = 0 \tag{1g}$$

$$r_{24} - r_{25} = 0 \tag{1h}$$

$$r_{25} - r_{26} - r_{27} = 0 \tag{1i}$$

$$r_2 - r_4 - r_5 = 0 \tag{1j}$$

$$r_4 - r_6 - r_7 = 0 \tag{1k}$$

$$r_5 - r_7 - r_8 = 0 \tag{1l}$$

$$r_7 - r_8 - r_9 = 0 \tag{1m}$$

$$2r_{12} + r_8 - r_{14} - r_{13} = 0 \tag{1n}$$

$$r_{29} - r_{32} - r_{33} = 0 \tag{1o}$$

$$r_{27} - r_{28} - r_{29} = 0 \tag{1p}$$

Another 11 equality constraints equations (2a) – (2k) for the biosynthetic load were derived from cellular mass composition. Here, different cell mass compositions are used for *E. coli* (Mandelstam 1982; Ingraham 1983; & Lee et al., 1996) and *B. subtilis* (Sauer et al, 1997). Other constraints are derived from NADPH and minimum ATP requirements (Eq. 3a -3b, Goel et al., 1996). A reference specific growth rate equal to 0.4 h^{-1} is used to generate flux units that can be scaled, when other growth rates are considered.

$$r_3 = 0.205 \cdot m \ (E.\,coli) \quad or \quad r_3 = 0.154 \cdot m \ (B.\,subtilis) \tag{2a}$$

$$r_{11} = 0.0709 \cdot m \ (E.\,coli) \quad or \quad r_{11} = 0.190 \cdot m \ (B.\,subtilis) \tag{2b}$$

$$r_{13} = 0.129 \cdot m \ (E.\,coli) \quad or \quad r_{13} = 0.194 \cdot m \ (B.\,subtilis) \tag{2c}$$

$$r_{15} = 1.493 \cdot m \ (E.\,coli) \quad or \quad r_{15} = 1.395 \cdot m \ (B.\,subtilis) \tag{2d}$$

$$r_{17} = 0.7191 \cdot m \ (E.\,coli) \quad or \quad r_{17} = 0.711 \cdot m \ (B.\,subtilis) \tag{2e}$$

$$r_6 = 0.897 \cdot m\ (E.\, coli) \quad or \quad r_6 = 0.816 \cdot m\ (B.\, subtilis) \tag{2f}$$

$$r_9 = 0.361 \cdot m\ (E.\, coli) \quad or \quad r_9 = 0.308 \cdot m\ (B.\, subtilis) \tag{2g}$$

$$r_{20} = 2.833 \cdot m\ (E.\, coli) \quad or \quad r_{20} = 2.942 \cdot m\ (B.\, subtilis) \tag{2h}$$

$$r_{22} = 2.928 \cdot m\ (E.\, coli) \quad or \quad r_{22} = 2.132 \cdot m\ (B.\, subtilis) \tag{2i}$$

$$r_{26} = 1.078 \cdot m\ (E.\, coli) \quad or \quad r_{26} = 1.071 \cdot m\ (B.\, subtilis) \tag{2j}$$

$$r_{30} = 1.786 \cdot m\ (E.\, coli) \quad or \quad r_{30} = 1.923 \cdot m\ (B.\, subtilis) \tag{2k}$$

These constraints are obtained from the network structure and reaction stoichiometry. Dauner and Sauer (2001) found the cell composition has a linear relation with dilution rate. In our research, fixed dilution rate (growth rate) is used, and we neglected the change of composition. There is also one type of constraint coming from the aggregate activity of different subsets of metabolic reactions. Such constraints provide connectivity between individual reaction rates. These constraints involve NADPH and ATP production. NADPH is the molecule responsible for reducing chemistry and ATP hydrolysis provides the energy for biosynthesis (i.e. free energy of hydrolysis drives otherwise thermodynamically infeasible polymerization). The NADPH and minimum ATP requirement (Goel et al. 1996, Dauner and Sauer 2001) are represented by Eq (3a), and Eq (3b).

$$2r_2 + r_{25} + r_{32} = 18 \cdot m\ (E.\, coli) \quad or \quad 2r_2 + r_{25} + r_{32} = 16.7 \cdot m\ (B.\, subtilis) \tag{3a}$$

$$-r_{12} + 3r_{14} + r_{18} + r_{23} + 3r_{27} + 2r_{21} + r_{29} + 2r_{33} + r_{31} - 2r_{19} \geq 33.1 \cdot m\ (E.\, coli)$$

$$-r_{12} + 3r_{14} + r_{18} + r_{23} + 3r_{27} + 2r_{21} + r_{29} + 2r_{33} + r_{31} - 2r_{19} \geq 31.7 \cdot m\ (B.\, subtilis) \tag{3b}$$

In order to guarantee the convex set is bounded, all reaction's rate are constrained to have a lower bound of 0 mmol g^{-1} h^{-1} and an upper bound of 20 mmol g^{-1} h^{-1} (Lee, *et al*. 2000) except flux r_{10} and r_{33} (see Fig. 1), which has a lower bound of -20 mmol g^{-1} h^{-1}. These later two reactions are reported to have standard free energy changes nearly equal to 0; hence, they may be quite reversible.

The linear systems of wild type *B. subtilis* and *E. coli* are defined by the 28 equality and 69 inequality constraints. These equality constraints are equation 1a - 1p, 2a – 2k and 3a and the inequality constraints are 3b and the upper and lower boundary constraints of 34 reactions. When gene deletion or addition occurs, extra constraints can be introduced (i.e., $r_{18} = 0$ for pyruvate kinase deletion). If the network structure is changed due to reaction addition/deletion, a new set of constraints is redefined based on same approach, for example, when citrate is fed to the culture, there are one new variables and 2 new inequality constraints (the boundary of the variable) about the flux boundary.

2.4.2. Choice of Objective Function

A linear objective function is usually chosen to maximize/minimize the product formation rate. Minimizing the glucose update rate is also used by assuming the cell utilizes the carbon source the most efficiently. Also a constant objective function may be used to find all feasible extreme points.

2.4.3. Results

We first calculated the convex polytope of *E. coli* and *B. subtilis* wild type cultured in glucose minimum medium. A constant objective function is used to find all feasible extreme points. The number of extreme point for *B. subtilis* and *E. coli* is 16 and 18, respectively. Selected fluxes from each scenario, carbon yields (Y_c), and ATP yields (Y_{ATP}), which are calculated from Equations (4) and (5), are reported in Table 2-1 and Table 2-2.

$$Y_c = \frac{1000 * C\% m}{72 r_1} \tag{4}$$

$$Y_{ATP} = \frac{1000 m}{r_{ATP}} \tag{5}$$

In Equations (4) and (5), C% and μ denote the carbon composition of cell mass (50% for E. coli and 45% for *B. subtilis*) and the specific growth rate, respectively.

Each solution reveals different ways for carbon to traffic through the *E. coli* and *B. subtilis* networks. The maximum biomass yield of *B. subtilis* is 0.847 g biomass C/g glucose C, which is corresponding to 121.9 g biomass/mol glucose. The solution is identical to the value found by Sauer et al (1998).

Another output that convex analysis produced is the minimal reaction set that can satisfy all the biomass synthetic and energetic requirements. This is done by checking the number of none zero flux in each flux scenarios. Convex analysis can also find if a reaction is redundant or not by checking f zero flux exists for the reaction. It is found that for the growth on glucose, pyruvate kinase is a required enzyme for *B. subtilis* (in all flux scenarios, it has a positive flux value, but it is not always required for *E. coli* (in some flux solution, it has zero flux value, these scenarios are marked in bold in Table 2).

Another use of convex analysis is the phenotype prediction. The prediction based on such analysis is rough and can only predict the range of phenotype. It is likely the precise position of an object in the space can not be predicted, but the smaller subspace that the object might fall in can be predicted. Such a prediction is very useful for formulating a metabolic engineering strategy and understanding a strain's metabolic capacity. Adding more constraints will make the sub space smaller, and allow for a closer look at the system. Phase plane projections are very useful to present the flux data obtained from convex analysis. It projects the solution space (a convex polytope) into a two dimensional plane. Because the phase plane can be used to represent the phenotype of a strain, it is also called phenotype phase plane. For example, in Figure 2-3, we project total acid production (acetate r_{19}, lactate r_{23}, succinate r_{28}) vs. glucose uptake rate phase plane. The wild type phonotype should be located anywhere in the biggest polygon. Adding more constraints, such as PYK flux = 0 ($r_{18} = 0$), or Malate flux = 0 ($r_{32} = 0$), the solution space is much reduced.

Pyruvate kinase mutation is a metabolic engineering strategy that proposed by Ataai' group (Goel et al. 1995) to reduce the acid by-production. The PYK mutation adds one more equality constraint ($r_{18} = 0$) than the wild type. In fact all the extreme solution of PYK mutants are in a subset of the extreme solutions for the wild type, because in these strains, there is an inequality constraints ($r_{18} \geq 0$). The subset of the PYK mutant in the Table 1, and Table 2 is marked with a different color. It is found that there is no solution in *B. subtilis* that has zero PYK flux.

The relationship between total acid production ($r_{19} + r_{23} + r_{28}$) and PYK flux is presented in the phase plane plots (Figure 2-4 and Figure 2-5). The plot is the projection of solutions space to the phase plane of total acid production vs. r_{18}. The results show that in both *E. coli* and *B. subtilis* strain, in order to minimize the acid production, PYK flux should be minimized. When PYK flux is greater than 2.5 mmol g^{-1} h^{-1} (*E coli*) and 3.3 mmol g^{-1} h^{-1} in (*B. subtilis*), the lower bound of acid by-production is greater than 0, which means the flux of glycolysis is in excess of TCA cycle.

For *E. coli*, even if PYK flux is constrained to be 0, it still may have high acid production capacity compare to *B. subtilis*. But in *B. subtilis*, when PYK flux is minimized to 0.3 mmol g^{-1} h^{-1}, the upper bound and lower bound of total acid production are both 0, which suggest pyk deletion must be effective in reducing acid by-production in *B. subtilis*, and it might be effective in *E. coli* also.

To validate the prediction of convex analysis, experiment works (acid production measurement) have been done on both *E. coli* (Zhu et al., 2001) and *B. subtilis* (Fry et al. 2000). The chapter 5 will discuss the experimental work done with the *E. coli* pyk mutant.

2.5 Chapter Summary

Convex analysis can be used to find the alternative optimal solutions which satisfy a given objective function. What is the biological significance of such a computation? What information are derived and represented to the biologist or metabolic engineer?

The identification of other solutions is of interest for several reasons. First, the different solutions may differ in how easy it is to implement them. For example, one solution may indicate more genetic manipulation than another for attaining the same value of the objective function. Secondly, there is fundamental value in knowing how many alternatives exist. This provides some insights on how redundant and robust a metabolic system can be. Finally, knowing the portfolio of alternatives ahead of time can result in the development of a more inclusive and tighter patent strategy.

In this chapter, a modified simplex method for finding the alternative solutions of an LP problem was presented. Algorithm is applied to calculate the solution space of acid production of *E. coli* and *B. subtilis*. Convex analysis not only finds all alternative solutions of a degenerate LP problem, but also provides more insight view of carbon metabolism.

Table 2-1 Alternate solutions of *B. subtilis* wild type on glucose minimum medium

Alternate Solutions	Select fluxes (mmol g-1 cell h-1)					
	r_1	r_2	r_{10}	r_{18}	r_{19}	R_{23}
1	3.281	2.924	0.2954	0.3746	0	0
2	3.3024	3.0524	0.1884	0.3532	0	0
3	3.39094	0.6152	2.71414	1.25414	0	0
4	3.9618	0.6152	3.285	1.825	0	0
5	4.0506	0.6152	3.3738	1.9138	0	2.3088
6	5.3334	0.6152	4.6566	3.1966	0	0
7	7.7706	0.6152	7.0938	5.6338	0	0
8	9.8226	0.6152	9.1458	7.6858	13.8528	0
9	10.9262	0.6152	10.2494	8.7894	16.06	0
10	10.9262	0.6152	10.2494	8.7894	0	16.06
11	10.9262	0.6152	10.2494	8.7894	0	6.3112
12	10.9262	0.6152	10.2494	8.7894	6.3112	0
13	10.9262	0.6152	10.2494	8.7894	11.1856	0
14	10.9262	0.6152	10.2494	8.7894	0	11.1856
15	11.3324	3.0524	8.2184	8.3832	16.06	0
16	11.3324	3.0524	8.2184	8.3832	0	16.06
continued						
Alternate Solutions	r_{27}	r_{32}	Yc(mol/mol)	Y_{ATP}	$Y_{acid\ total}$	Y_c (g/mol)
1	0	0.2568	0.846625	31.84713	0	121.9141
2	0	0	0.841139	30.0066	0	121.124
3	0.98949	3.88491	0.819176	31.84703	0	117.9614
4	1.0656	3.8088	0.70114	31.84713	0	100.9642
5	0	4.8744	0.685769	31.84713	0.56999	98.7508
6	4.8744	0	0.520827	5.974715	0	74.99906
7	4.8744	0	0.357473	6.44388	0	51.47608
8	0	4.8744	0.282795	31.84713	1.410299	40.72242
9	0	4.8744	0.254231	27.08706	1.469861	36.60925
10	0	4.8744	0.254231	4.20755	1.469861	36.60925
11	4.8744	0	0.254231	4.002337	0.577621	36.60925
12	4.8744	0	0.254231	5.849185	0.577621	36.60925
13	4.8744	0	0.254231	5.119384	1.023741	36.60925
14	4.8744	0	0.254231	2.983685	1.023741	36.60925
15	0	0	0.245118	13.60989	1.417176	35.29702
16	0	0	0.245118	3.646627	1.417176	35.29702

Table 2-2 Alternate Solutions of *E. coli* wild type on glucose minimum medium

Alternate Solutions	Select fluxes (mmol g-1 cell h-1)											
	r_1	r_2	r_{10}	r_{18}	r_{19}	r_{23}	r_{27}	r_{32}	Y_c(mol/mol)	Y_{ATP}	$Y_{acid\ total}$	Y_c (g/mol)
1	**4.00917**	**2.74762**	**1.17955**	**0**	**0**	**0**	**1.27357**	**0**	**0.692856**	**13.35016**	**0**	**110.857**
2	**4.22143**	**3.3844**	**0.75503**	**0**	**1.48583**	**0**	**0**	**0**	**0.658018**	**22.93742**	**0.351973**	**105.2829**
3	**4.22143**	**3.3844**	**0.75503**	**0**	**0**	**1.48583**	**0**	**0**	**0.658018**	**16.08499**	**0.351973**	**105.2829**
4	**4.94251**	**0.6476**	**4.21291**	**0**	**0**	**0**	**3.84025**	**1.63335**	**0.562018**	**7.282129**	**0**	**89.92282**
5	5.75918	0.6476	5.02958	2.45002	0	0	5.4736	0	0.482322	5.251927	0	77.17148
6	**8.2092**	**0.6476**	**7.4796**	**0**	**0**	**0**	**5.4736**	**0**	**0.338374**	**5.251927**	**0**	**54.1398**
7	**8.78276**	**0.6476**	**8.05316**	**0**	**11.52076**	**0**	**0**	**5.4736**	**0.316276**	**24.20358**	**1.311747**	**50.60419**
8	**8.78276**	**0.6476**	**8.05316**	**0**	**0**	**11.52076**	**0**	**5.4736**	**0.316276**	**5.395906**	**1.311747**	**50.60419**
9	**8.78276**	**0.6476**	**8.05316**	**0**	**0**	**0.57356**	**5.4736**	**0**	**0.316276**	**5.024881**	**0.065305**	**50.60419**
10	**8.78276**	**0.6476**	**8.05316**	**0**	**0.57356**	**0**	**5.4736**	**0**	**0.316276**	**5.212672**	**0.065305**	**50.60419**
11	10.6394	0.6476	9.90976	1.8566	15.23396	0	0	5.4736	0.261085	19.76316	1.431849	41.77361
12	10.6394	0.6476	9.90976	1.8566	0	15.23396	0	5.4736	0.261085	4.14897	1.431849	41.77361
13	10.6394	0.6476	9.90976	1.8566	0	4.28676	5.4736	0	0.261085	3.926069	0.402915	41.77361
14	10.6394	0.6476	9.90976	1.8566	4.28676	0	5.4736	0	0.261085	4.972077	0.402915	41.77361
15	10.6394	0.6476	9.90976	7.3302	9.76036	0	5.4736	0	0.261085	4.655337	0.917382	41.77361
16	10.6394	0.6476	9.90976	7.3302	0	9.76036	5.4736	0	0.261085	2.969018	0.917382	41.77361
17	11.0955	3.3844	7.62909	6.87407	15.23396	0	0	0	0.250352	12.82591	1.372987	40.05632
18	11.0955	3.3844	7.62909	6.87407	0	15.23396	0	0	0.250352	3.725898	1.372987	40.05632

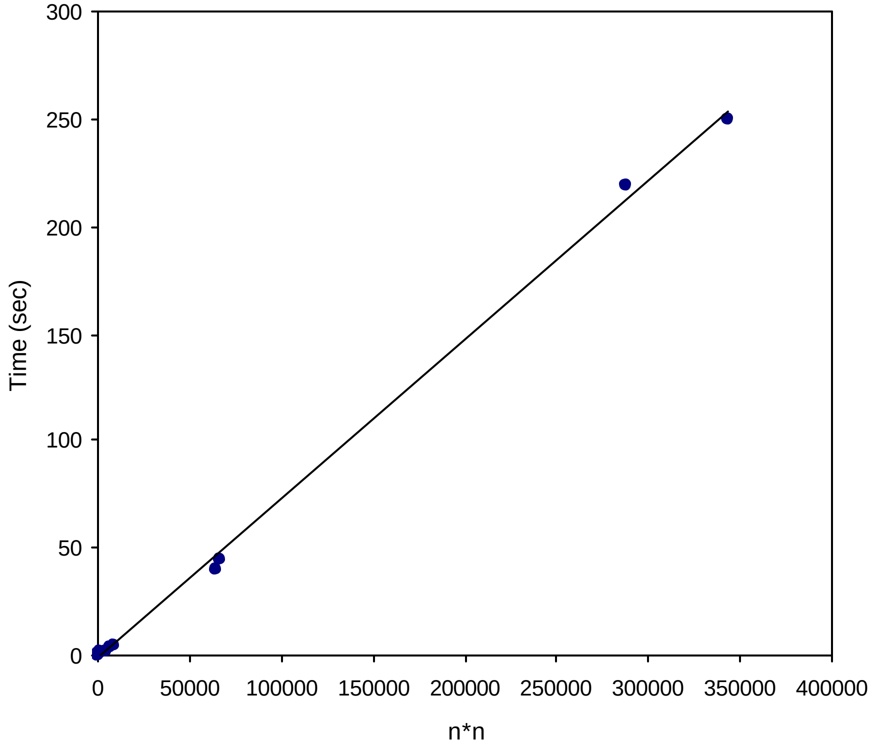


Figure 2-1 Relationship between the computation time and number of alternative solutions.

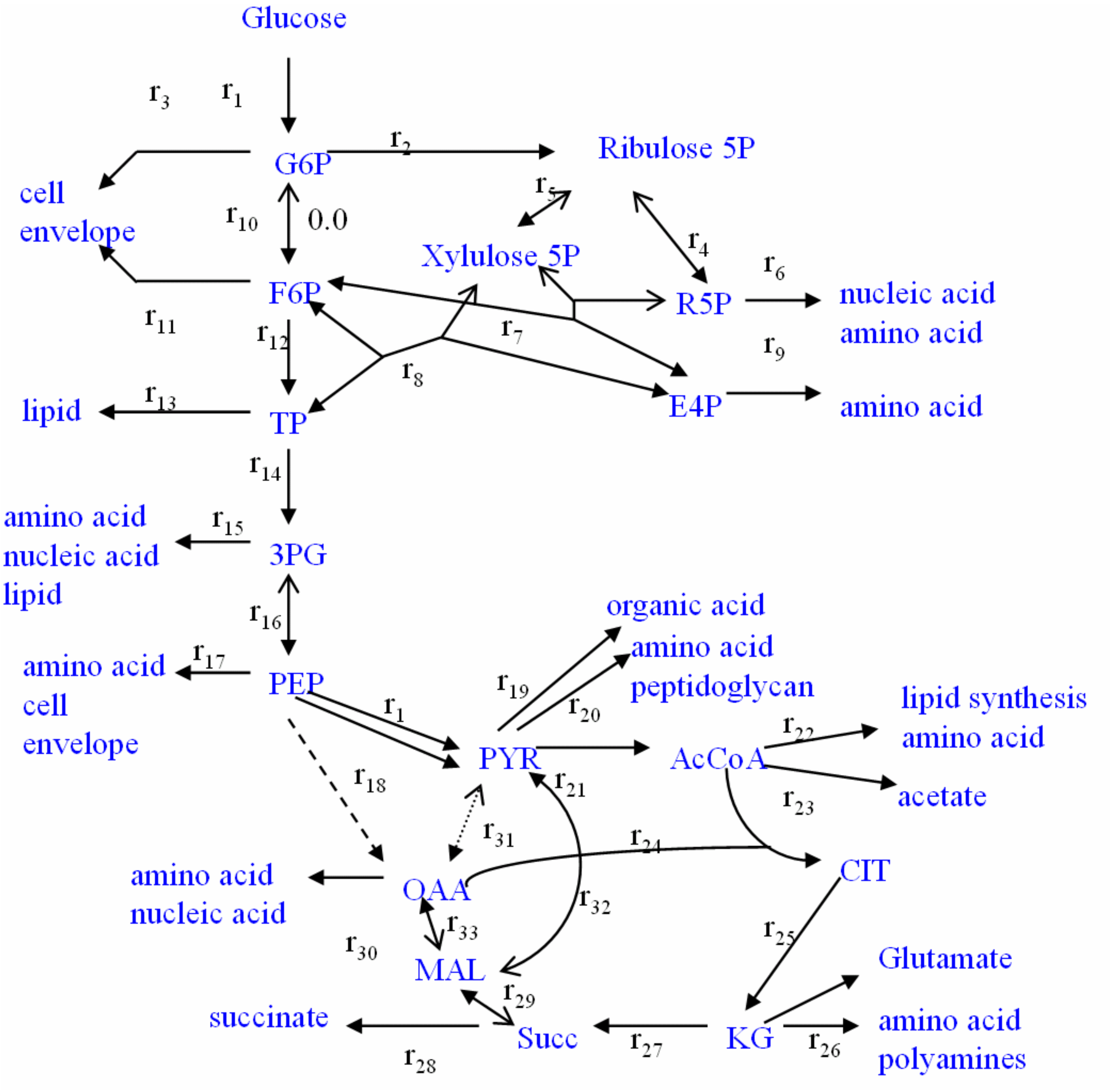


Figure 2-2 Metabolic networks of *B. subtilis* and *E. coli*. The two-way arrows represent reversible reaction with the bold heads showing the forward reaction. The dash arrow from PEP to OAA is only for *E. coli* and the dot arrow from PYR to OAA is only for *B. subtilis*.

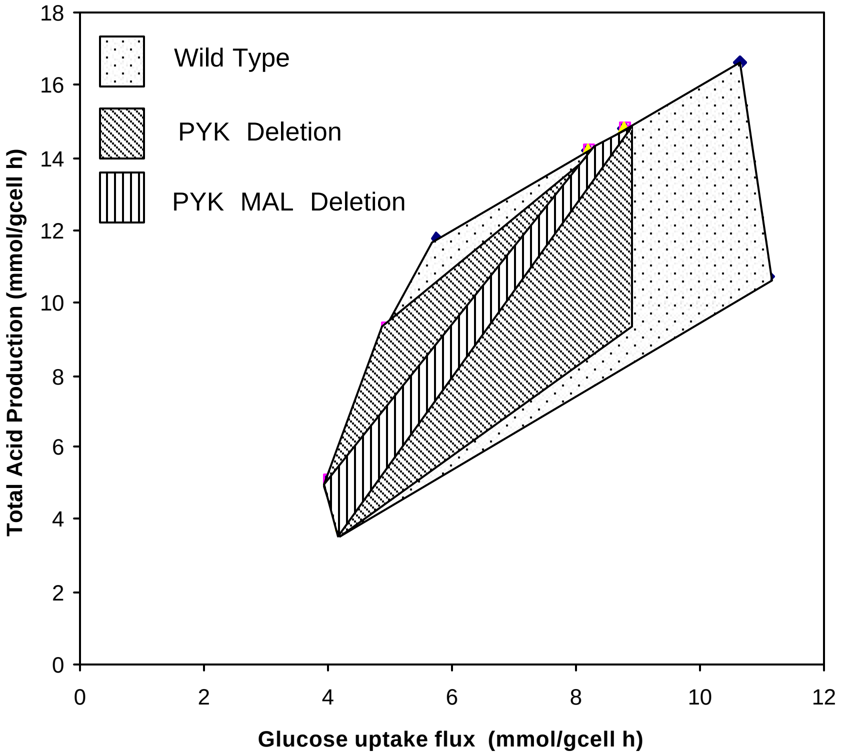


Figure 2-3 Phenotype phase place of *E. coli*: total acid production vs. glucose uptake rate.

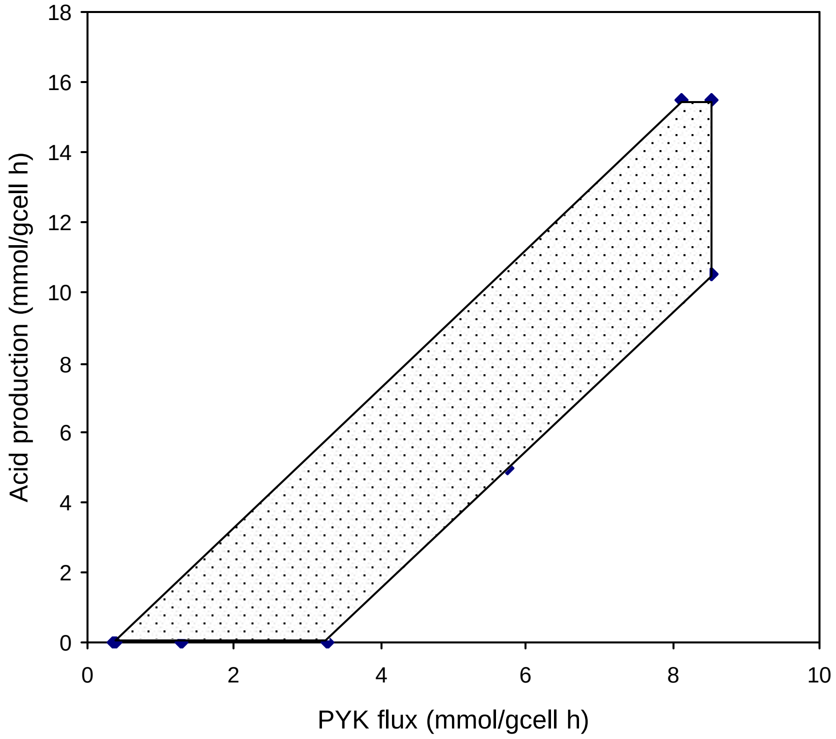


Figure 2-4 Phase plane of B. subtilis wild type (acid production vs. PYK flux)

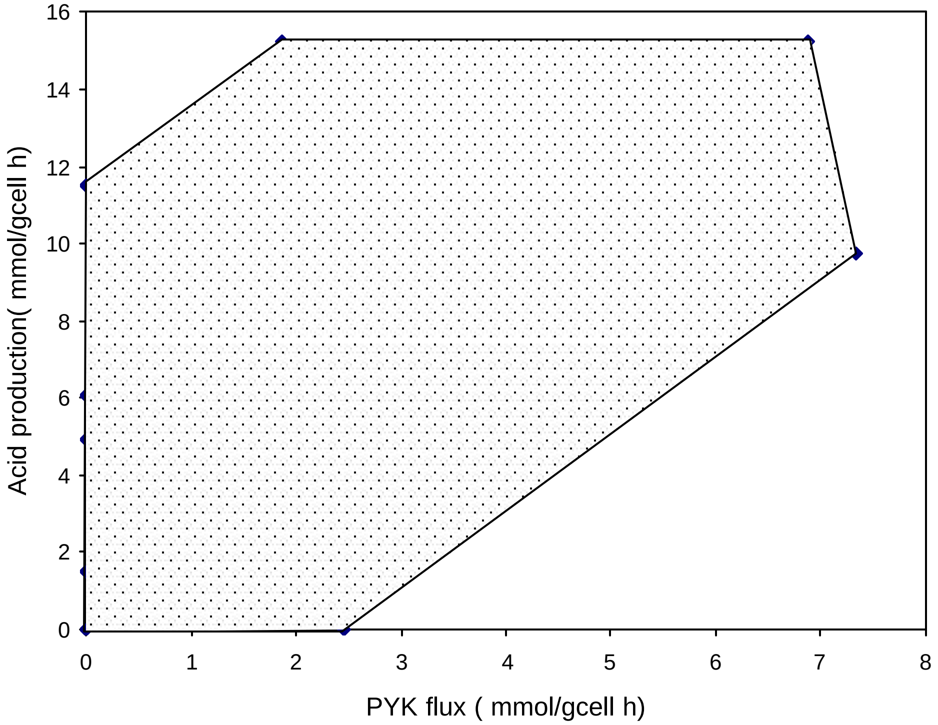


Figure 2-5 Phase plane of E. coli wild type (Acids production vs. PYK flux)

3.0 PRECISELY DETERMINE THE FLUX USING NMR / GC-MS & DESIGN OF TRACER EXPERIMENT

3.1 Introduction

Due to the lack of constraints on the metabolic network, multiple solutions arise from convex analysis. Any one or a combination may constitute the actual flux distribution. For the *E. coli* example in last chapter, minimizing PYK flux yields nine different solutions. In order to determine the flux distribution precisely, more constraints must be added to the system. Isotopomer tracer measurements provide an effective route for introducing more information that can be used to discriminate between alternative flux scenarios.

Recently, several elegant approaches have been proposed for deriving flux distribution from isotopomer data (Schmidt et al., 1998; Wiechert et al., 1997). In this chapter, I will present a different approach to find the flux distribution that integrates the NMR/MS simulation with the convex analysis. The algorithms proposed in this chapter are all implemented in the software *MetaboLogic*.

3.2 Isotopomer Distribution Calculation at Steady State

An isotopomer is a metabolite with a specific ^{13}C labeling pattern. A metabolite with *n* carbons thus has 2^n isotopomers (Schmidt et al., 1997). For each metabolite, at steady state the composition of different isotopomer is a constant. Hence, for a metabolic network, the isotopomer distribution for each metabolite can be decided by the isotopomer distribution of the substrate fed to the system, flux distribution, and reaction reversibility.

To solve the isotopomer distribution, the relationship between isotopomers in a single reaction should be found first. Atom Mapping Matrices (AMM) (Zupke 1997) and Isotopomer Mapping Matrices (IMM) (Schmidt 1997) are the matrices that describe the switching of carbon atoms. The detailed information about these methods is given as Appendix B and C. In our approach, Schmidt's IMM approach is used. Appendix D gives an algorithm to construct IMMs based on AMMs.

Based on conservation law of mass, we have the following equation for a given metabolic network.

$$G^T v = 0 \qquad \text{(Eq. 4 -1)}$$

G is the stoichiometric coefficients matrix and *v* is the flux distribution vector. Each row of G^T corresponds to the mass balance of one intracellular metabolite. For simplicity, first assume that all reactions are irreversible. Consider the i^{th} intracellular metabolite mass balance. We have

$$\sum_{j=1}^{m} g_{ij} v_j = 0, \quad \text{for all } \mathrm{i} \in [1, \ldots \mathrm{g}], \mathrm{i} \in [1, \ldots \mathrm{n}] \qquad \text{(Eq. 4-2)}$$

where g_{ij} is the stoichiometric coefficient of i^{th} metabolite in j^{th} reaction. If $g_{ij} > 0$, for the j^{th} reaction, metabolite i is a product. For each reaction j, we can calculate the i^{th} metabolite's isotopomer's production (denoted as $U_{i,j}(\boldsymbol{I})$) using the following relation.

$$U_{i,j}(\boldsymbol{I}) = vj*(IMM_{k1 \rightarrow i} * I_{k1} \otimes IMM_{K2 \rightarrow j} * I_{k2}) \qquad \text{(Eq. 4-3)}$$

k_1 and k_2 denote the substrates in the j^{th} reaction that contribute carbon atoms to the metabolite i, and v_j is the flux of the j^{th} reaction. I is the distribution vector of different metabolites. $\otimes$ is the pair-wise multiplication of two vectors with same length.

$U_{i,j}$ is not a convex function of ***I*** due to the pair-wise multiplication. This property makes it difficult to search the flux space to match the experimental data. The form of $U_{i,j}$ will be simpler in the case that only one substrate is involved carbon switching, such as A → B + C or A → C.

$$Ic = IMM_{A \to C} * I_A$$

$$\text{And } U_{i,j}(\boldsymbol{I}) = vj*(IMM_{k1 \to i} * I_{k1}) \qquad \text{(eq. 4-4)}$$

The consumption rate of isotopomer of A, and B in the reaction (A + B -> C + D) is identical to the rate of the reaction.

Rearranging equation 4-2 by moving all consumption term to one side of the equation yields

$$\sum_{j=1, g_{ij}>0}^{m} g_{ij} v_j = \sum_{j=1, g_{ij}<0}^{m} - g_{ij} v_j \qquad \text{(Eq. 4-5)}$$

Based on the conservation law for each isotopomer, we have a corresponding equation

$$\sum_{j=1, g_{ij}>0}^{m} U_{i,j}(I) = \sum_{j=1, g_{ij}<0}^{m} I_i v_j = I_i \left(\sum_{j=1, g_{ij}<0}^{m} v_j \right) \qquad \text{(Eq. 4-6)}$$

The left term is formation rate of isotopomers for metabolite *i*, and the right term is the consumption rate of isotopomers. By rearranging the equation, we obtain

$$I_i = \frac{\sum_{j=1, g_{ij}>0}^{m} U_{i,j}(I)}{\sum_{j=1, g_{ij}<0}^{m} v_j} \qquad \text{(Eq. 4-7)}$$

Notice for a metabolic network, we can write down a list of Equations 4-7. The number of such an equation is *g* (number of intracellular metabolites), which is less than n (the total number of metabolite). Based on this equation, a steady state isotopomer distribution (*I*) can be solved using an iteration algorithm.

3.2.1 Reaction Reversibility Consideration

The scrambling due to reversibility can be accounted by exchange coefficient (e), which is defined as the relative magnitudes of the net flux and reverse flux.

$$e = V_R /(V_R + V_N) = V_R / V_F$$

$$V_N = V_F - V_R$$

The choice of *e* is usually based on the thermodynamics or those assignments made by other works (Zupke and Stephanopoulos, 1994; Schmidts et al. 1997 and Klapa et al., 1998). After introducing the exchange coefficient, a reversible reaction can be replaced by two reactions, which have different directions but positive fluxes. The original stoichiometrics matrix G should be expanded. If there are *k* reversible reactions, the number of column in matrix G is now $k + m$. The equation 47 holds true, after replacing each reversible reaction with two irreversible reactions.

3.2.2 Algorithm to find IDV

If the substrate's IDV is known, a simple iteration method can be used to find the steady state IDV distribution. The algorithm is described by following:

Start with $I_i^0 = (1\ 0\ \ 0\ \ 0\ \dots\ 0\,)^{\mathrm{T}}$ $\mathrm{i} \in [1, \dots \mathrm{g}]$, and I_{sub}.

Calculate $I_i^k = \dfrac{\sum\limits_{j=1, g_{ij}>0}^{m} U_{i,j}\left(I^{k-1}\right)}{\sum\limits_{j=1, g_{ij}<0}^{m} v_j}$, k=0, 1, 2 , …

if $\left\| I_i^k - I_i^{k-1} \right\| \le e$, or k < $\mathrm{K}_{\mathrm{Max}}$ stop the iteration

There is no mathematical proof for the convergence of the algorithm, but in the application of different example network, it converges when *k* (iteration number) is less than 100. And the initial value of I_i doesn't have significant effect on the convergence of algorithm.

3.3 NMR or MS Simulation Based on IDV

If the IDV of target molecule is found, the NMR and MS spectra can be simulated. MS spectra can be represented by mass distribution vector (MDV). A matrix (M) can be constructed to convert an IDV into MDV.

$$MDV = M*IDV$$

where M is a matrix decided by the number of carbons in the metabolite only.

NMR spectra simulation is more complex compared to the MS simulation, and it depends on the chemistry of different metabolites. In the current work, only the NMR spectrum of glutamate was computed. There are three reasons glutamate was chosen as the signal molecule. First, glutamate is the most abundant metabolite in *E. coli* (Tempest et al., 1970); hence, the signal-to-noise ratio is large. Second, the signals from the three interior carbons of glutamate (C2, C3, and C4) are not confounded by effects such as partial saturation or different nuclear Overhauser enhancements. Third, glutamate is located deep enough within metabolism to have "recorded" the entire flux pattern.

3.4 Operative Flux Distribution Search

Once GC-MS and NMR spectra experiment data are collected, the flux distribution can be found by matching the simulation data with the experiment data. In order to quantify the difference between to spectra data, a difference function should be defined first.

3.5 Tracer Experiment Design

The choice of labeled substrate (e.g. 1-^{13}C, 2 -^{13}C, U-^{13}C, etc) will alter the difference between experimental and simulated spectra. We can iterate all the different substrate labeling patterns to choose the labeled substrate that makes the difference between these spectra most significant.

Phalakornkule et al. (2000) proposed the alternative choice for labeled glucose to be screened by assessing the relative intensities of the singlet and satellite peaks that would appear in the NMR spectra. In the example research on the glucose metabolism of *B. subtilis*, the ratio of C3 singlet and C4 singlet is useful characteristic of a spectrum. The ratio of different peaks in the spectra then is used to characterize the spectra.

In the case that many peaks exist in a spectrum, we proposed a formal method to evaluate the difference between spectra. If there are only two possible scenarios exist, the choice of the labeled substrate is aimed to minimize the value of δ (in Eq. 4-8). If there are n different scenarios, then there are *(n*(n*-1*))/*2 values of δ need to be minimized, and these δ's can be list into a matrix form.

$$\Delta = \begin{bmatrix} 0 & d_{12} & d_{13} & \cdots & d_{1n} \\ & 0 & d_{23} & \cdots & d_{2n} \\ & & \cdots & \cdots & \cdots \\ & & & 0 & d_{(n-1)n} \\ & & & & 0 \end{bmatrix}_{n*n} \qquad \text{(Eq 4-9)}$$

where d_{ij} is the difference between spectra simulated from flux scenario *i* and *j*.

There are many choices for defining an objective function *f(D)* that can character the overall difference between each spectra. One possible form of *f(D)* could be the function shown as equation 4-10.

$$F(L) = f(\Delta) = \sum_i \sum_j d_{ij} \qquad \text{(Eq. 4-10)}$$

where *L* denotes different labeled substrate. A labeled substrate that maximize such of *f(D)* would be a good choice.

It is noted that the elements in vector S is chosen to be the intensity at the major peaks in the spectra. Generally, it could be the intensity at any resonant frequency, but the former approach has at least two advantages. First, less calculation is needed since the dimensions of vector S is much decreased. Secondly, the noise that appears in the base line of experimental spectra won't affect the comparison of spectra. The disadvantage of this approach is the computer/man time required to recognize the peak in the spectra.

Exchange coefficient *e* significantly affects the simulated spectra. In the label design phase, the exchange coefficients are set to 0.5 (Phalakornkule, *et al*. 2001). A robust labeling substrate can be tested by varying the value of *e*, and to check if the difference of each scenario remains significant. The task was to confirm that the chosen labeled glucose could differentiate two flux scenarios over a large range of reaction reversibilities.

3.6 Chapter Summary

Using the convex analysis in combination with IMM-based spectra simulation allows for the prediction of the NMR spectra associated with a particular flux distribution and labeled precursor compound. These predicted spectra, in turn, can enable the design of 13C NMR experiments. Based on the spectra, the labeled glucose can be chosen such that the spectra associated with different flux distribution candidates will differ most significantly. From the practical standpoint, these algorithms and strategies may facilitate the performance of rapid-screening NMR experiments and minimize the use of expensive 13C-labeled precursor compounds. The result of these methods will be presented and discussed in the next chapter. Although glutamate was chosen as the target NMR analyte based on several desirable properties, the same experimental design strategy can be used for other analytes because the isotopomer distributions are readily extracted from the IMM calculation.

4.0 METABOLOGIC PROGRAM

4.1 Introduction

In this chapter, we will describe the user interface of the MS-Windows program, *MetaboLogic*, which integrates the modified simplex method and NMR spectra simulation presented in previous chapters. The interface enables one to build and edit a network model by drawing/modifying a picture on a computer screen as opposed to programming with a symbolic language or working at the source code level. Once a metabolic model has been "drawn," the program's computational component can (1) enumerate the flux distributions that satisfy the constraints and optimize an objective function and (2) design ^{13}C NMR and/or GC/MS experiments that can help validate that is desired flux distribution is actually used.

The first facet acknowledges that biological systems have built in redundancy (e.g. more than one ATP generating reaction exists; induction/repression adds new paths). In a large network it is not always straightforward to intuit what flux distribution scenarios are feasible; hence, a computerized aid can be of assistance. The second aspect pertains to designing and interpreting experiments that will ultimately help to validate a metabolic engineering strategy with intracellular information. Ultimately finding a match between the predicted and observed spectra would support process validation and patent applications; i.e. both the extracellular and intracellular measurements indicate that a strategy's intended outcome occurs and the claimed mechanisms are actually responsible for the outcome.

The program's interface and requirements are summarized first. Thereafter, the sequence of how a user works with the program and the nature of the interfaces is illustrated with an example problem. We conclude with a discussion of the limitations and how they may be addressed.

4.2 Overview of Model Construction, Visualization, Computational Components & System Requirement

The model construction process mimics the process of using a software application (e.g. PowerPoint) to draw a network on a computer screen. To build a network, the user first "clicks" reactant nodes and reaction links to draw a figure. Each node is assigned a label (e.g. metabolite name) when it is defined. At this point it is also possible to access and edit properties such as the exchange coefficient and isotopomer mapping matrix. The size of model is only limited by the available memory on the computer. For a large network, multiple pages (i.e. computer screens) are provided to group some sub-networks. Also different graphical objects can be added to the model "picture" for customization and presentation purposes.

Once the network is drawn on the computer screen by the user, the software automatically generates the mole balance equations for each metabolite. These equations are accessible to the user for checking or modification. They are presented in tabular form in a way that resembles how one would write them on a sheet of paper. Additionally, physiological-based constraints (Bonarius et al., 1997) can be conveniently inputted and integrated to the model. An example of such a constraint is the aggregate activity of a subset of reactions must provide a minimum or certain amount of ATP. This feature allows a user to check model assumptions, develop case studies, or perform sensitivity analyses.

There are two computational engines: (1) prediction of all flux distributions that satisfy all constraints and optimize an objective and (2) NMR/GC-MS experiment simulation and design. The second engine uses the output of the first engine as an input. In one version mixed integer linear programming is used to find alternative flux distributions by wedding the program to GAMS software; the algorithm has been described elsewhere (Lee et al., 2000). The current version employs the Depth First Search Algorithm to implement convex analysis (Zhu et al., 2002; Sedgewick, 2001), which simplifies the integration of different software platforms. More details on the Depth First Search Algorithm are provided in Appendix 1.

The alternate and feasible "flow sheets" found for a particular objective and set of constraints are provided to the user in the form of a list. "Clicking" an entry in the list will update the window to display the network and flux values. Clicking another list entry will provide a new figure; flux values that changed from the prior figure are shown in a different color to flag values that differ between solutions.

Once the convex analysis is complete, each solution can be examined to find the minimum gene set (zero flux means that gene is not required) that satisfies all the constraints. Moreover, by checking the rate of a specific reaction in all scenarios, whether a reaction is redundant or not can be reported. This search of the solution space provides information akin to Burgard et al (2001 a, b) investigation of the minimal reaction sets for *E. coli* metabolism under different growth requirements and uptake environments.

The prediction of the NMR/GC-MS spectra associated with a particular flux scenario is based on the isotopomer mapping matrix method (Schmidt et al., 1997). One can either choose a particular labeled compound (e.g. 1-^{13}C glucose) or allow the program to identify the labeled glucose that will maximally distinguish one flux distribution from another by having significantly different spectral features. The approach used to enable NMR experiment design has been described elsewhere (Phalakornkule et al., 2001). In the version described here, the selection of the best labeled compound to use has been automated as opposed to making this decision by manually comparing the predicted spectra.

Automation of label selection for MS/GC or NMR experiments was accomplished by performing two steps. First, the quantitative comparison of NMR spectra is accomplished by vectorizing the spectra. Each dimension corresponds to the intensity of a feature (e.g. singlet intensity of carbon 3 in glutamate), and the resultant magnitude and angle are computed. The best labeled glucose to use can be identified by maximizing the differences between the vectorized spectra.

Secondly, a function that quantifies the difference between the spectra vectors is defined. For MS/GC experiment, the "difference" function is defined as the distance between the spectra vectors because MS spectra data are typically normalized. For an NMR experiment, the difference function is based on the angle between spectra vectors. The substrate label pattern that maximizes the average difference between different flux scenarios is found by iterating through all different label patterns that are available for use. In the labeling design phase, all exchange coefficients are fixed to a reasonable value ($\varepsilon = 0.5$). The user can change these values of the exchange coefficients to assess how the labeling pattern works when reversibilities are higher or have a wider range, which allows for the most robust label pattern to be identified.

4.3 Example Problem

The elimination of pyruvate kinase activity has been suggested to be a strategy for minimizing glucose uptake and eliminating acid production by some bacteria (Goel et al., 1995). The theory is the mutation leads to the reduced formation of the acid precursor, pyruvate, from glucose. Thus, glycolysis and the TCA cycle are more tightly coupled where the latter is fed by the pyruvate formed by the activity of the phosphotransferase system. Glucose-6-phosphate, however, is still free to flow through the hexose monophosphate pathway thereby producing precursors and NADPH. The efficacy of the mutation strategy has been recently been supported by extracellular measurements (Fry et al., 2000; Zhu et al., 2001).

Here, we shall show how the program can be used to quickly develop a model, explore *in silico* the potential effects of the suggested mutation in *E. coli*, and then generate a ^{13}C NMR experimental design. In general, looking at the results generated by using different labels ahead of time would be useful for eliminating inconclusive or unnecessary experiments. Moreover, the effect of assumptions (e.g. assigned fluxes, cellular composition) can be assessed ahead of time to gain a sense of how uncertainties may affect the interpretation of the NMR experiments.

4.4 Application of Software to the Example Problem

Model Construction. The model-picture construction process has two components. The first step entails reactant node placement on the computer screen and formatting/definition. The second step involves drawing a reaction link and identifying its properties. These two steps are shown in Figure 4-1, where the reaction involves the phosphorylation of glucose. The user types the name of a reactant in a text box (see Fig. 4-1a). Clicking the text box opens the text window shown in Figure 4-1b. In this box, format (e.g. border around species), the number of carbons, and other details can be addressed. Drawing an arrow between two species using the tool in the menu links reactants to product species. Clicking the arrow opens another text box (see Fig. 4-1d) that allows the user to identify the reaction as reversible, input values of exchange coefficients, and set lower and/or upper bounds of rate if desired. The default stoichiometry is 1:1 because this value is common for glycolytic and other elemental reactions in the central metabolic pathways. However, the displayed values can be changed by the user. It took about 30 minutes to build the full "model-picture" shown Figure 4-2. The network's layout can be customized by the user as indicated by the different colors used for the different metabolites.

The balance equations for the metabolites are automatically produced (see Figure 4-3). The equations are identical to those manually derived by Phalakornkule et al. (2000). All the user has to do is input the "biosynthetic loads" based on cellular composition; these values can thus be varied to permit sensitivity analyses.

Flux Trafficking Solutions. Once the user has built a model "picture" and inputted assumptions, the different analysis tools are accessed through the software menu or toolbar. For this example, to calculate all possible metabolite trafficking solutions, the user can use the "**lp**" toolbox. By click the "**linear programming**" menu, a dialog will pop up to accept the objective function (see Figure 4-4) and other options for the linear optimizations. If a constant linear objective function (i.e. "0") is used, the software will find all the solutions that satisfy the mass balances and inequality constraints. Otherwise, a specific objective can be inputted (e.g. minimize pyruvate kinase flux) and an optimal subset will be reported (see Figure 4-4).

The number of different flux solutions is eighteen for the *E. coli* wild type. When the objective is to minimize pyruvate kinase flux, which is the aim of the aforementioned mutation strategy, nine flux distributions are feasible. The nine solutions are actually a subset of the eighteen solution found for the wild-type strain. Overall, pyruvate kinase deletion can generate a number of outcomes; a subset of the outcomes is consistent with the goal of acid elimination and reduced glucose uptake. The computational time required to find the possible flux solutions using a PIII-900 MHz personal computer with 128 RAM was less than 1 second. The computation time increases with the number of different solutions in a second-order fashion.

Once the search for the optimal flux distribution scenarios is done, the user can choose **Show Flux** in **View Menu** to view the flux values as a conventional biochemical pathway diagram. A window will be displayed that asks the user to select the scenario of interest. The user then chooses a number between 1 and the maximum numbers of flux scenarios found by the LP solver. The user can also view the different scenarios in sequence by a toolbox on the toolbar. These values can also be exported to Excel for further analysis or an email attachment for a collaborator. An example of the output is shown in Figure 4-5. This solution has minimal pyruvate kinase and acid fluxes.

NMR Experimental Design. The tool under the **Tool Menu**, **Simulate Spectrum**, enables the design of a ^{13}C NMR experiment or if preferred, a mass spectrometry experiment. Here, we restrict the demonstration to NMR experimentation using glutamate as signal molecule.

Via the **"NMR Simulation"** window shown in Figure 4-6, the user can select one or multiple scenarios of interest. The labeling pattern and degree of ^{13}C enrichment of glucose can also be varied by the user. Enrichments between 0 and 100 per cent can be chosen. Additionally, the seven forms of labeled glucose that are now commercially available can be chosen in the **"Substrate Labeling Pattern"** box.

Once the user presses the "**OK"** button, the program produces several MATLAB code files (m file) and executes these files to perform the NMR simulation. It first finds the isotopomer distributor vector (IDV) for each reactant in the network. Then, the software simulates the NMR spectra based on the glutamate's IDV. The program will automatically generate a plotted spectrum on the screen. For the flux scenario shown in Figure 5, using the choices of exchange coefficients (default values of 0.5 used, see Figure 4-2.d), and uniformly labeled glucose (see Figure 4-6) results in the NMR spectrum shown in Figure 4-7. The intensities of singlets and satellites vs. ppm arising from the central glutamate carbons (carbons 2, 3, and 4 only) are shown. To more easily view the fine structure, the user can magnify a section by using the zoom in/out function denoted by the magnifying glasses in the tool bar.

Before an experiment is done, it is important to check if a particular labeled substrate can provide an NMR spectrum that will allow one to conclusively distinguish between the different flux pattern possibilities. Reaction reversibility and other facets may blur the distinctiveness of the spectra. To address this problem, a **Substrate Labeling Pattern Design Tool** can be employed. This tool calculates the average difference between the NMR spectra associated with the different flux scenarios and possible combinations of exchange coefficients.

As we discussed in chapter 3, the best labeled glucose to use can be identified by maximizing the differences between the vectorized spectra, which is defined as the angle between different spectra vectors. To illustrate, the spectra arising from the nine flux alternatives will maximally differ in their features when 2-^{13}C glucose is used (see Table 4-1). Glucose labeled in the 1, 5, and 6 positions also yield spectra that can distinguish between the nine scenarios, but the contrast is lower than when 2-^{13}C glucose is used. Uniformly labeled glucose or labeling in positions 4 and 3 may not prove to discriminate between the spectra that can arise from the nine flux distribution scenarios.

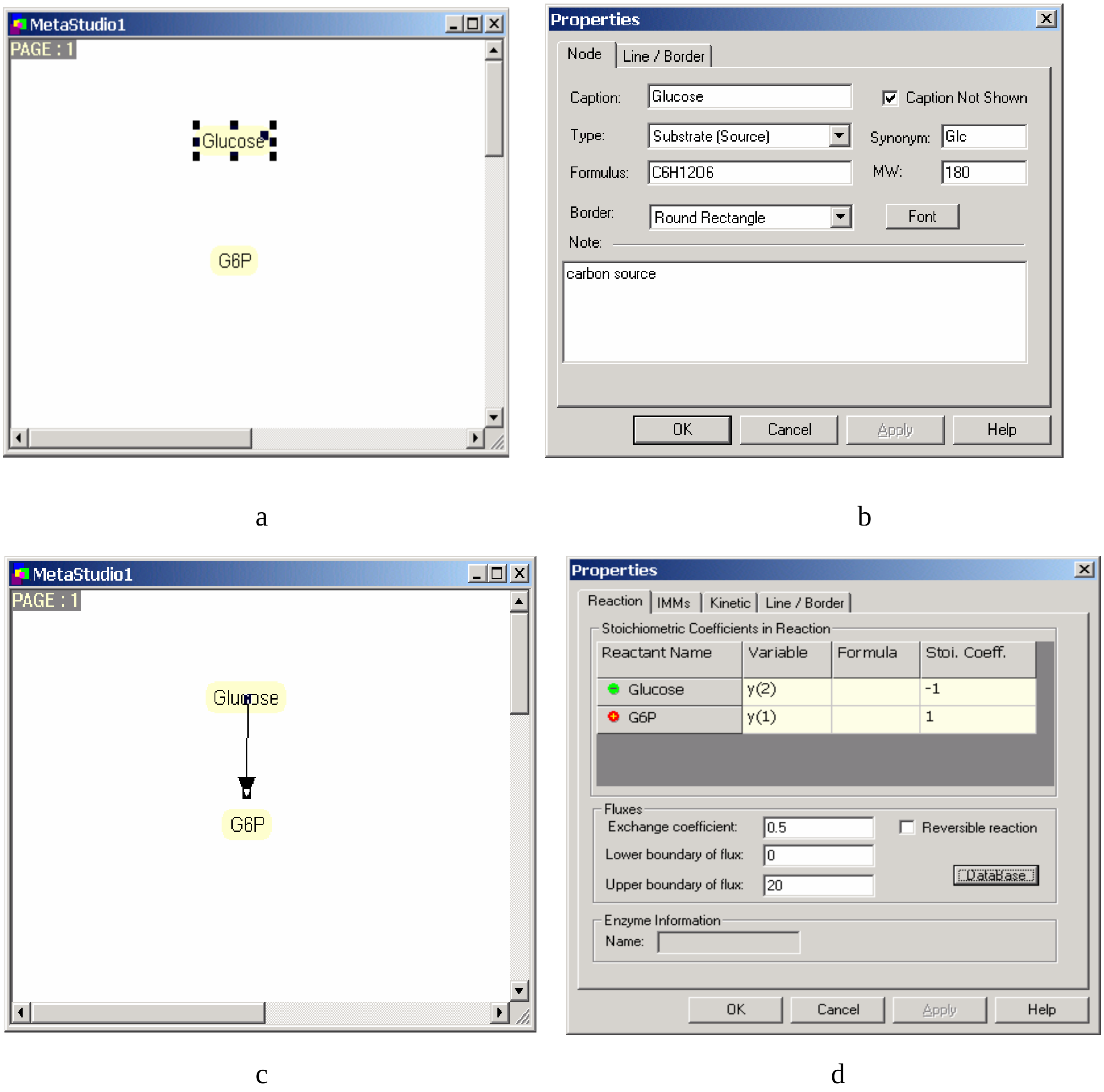
Once an NMR experiment has been performed, the software can also be used to further interpret the data in terms of what fluxes and reaction reversibility could generate the observed spectra. This tool is accessed by choosing '**Spectra Search"** from the '**Tools"** menu. A dialog box (see Fig. 8) will appear to request the NMR data file and other information. The user can also define the upper bound and lower bound of exchange coefficients for each reversible reaction. This feature is useful because two different flux distributions could conceivably yield similar NMR spectra if the reaction reversibility was different. After the searching is finished, the software will report the most likely flux distribution and exchange coefficients, as well as the difference between the simulated spectra and the experimental one.

4.5 Chapter Summary

A Windows program for metabolic engineering analysis and experimental design has been developed. A graphical user interface is intended to enable model construction, modification, and communication. Such an interface eliminates the need for programming at the source code level when one wishes to investigate other's computational results or to modify a problem. The example shown can be run on a 900 MHz lap top computer. The different tasks are executed in seconds to several minutes.

Two limitations exist which are the subject of ongoing work. First, the NMR spectra prediction focuses on the analyte, glutamate. The rationale is glutamate is an informative and often used analyte. Moreover, the demands and cost of performing a glutamate-based NMR experiment are less that, for example, acquiring the spectra of hydrolyzed protein. Thus, a layered approach, where one first ascertains if the easier NMR experiment can do the job would seem to be a logical and cost effective first step. If one finds through the simulation and design functions that the glutamate NMR spectra cannot provide conclusive information, then a strong justification for a including more analytes in a more labor- and cost-intensive experiment is provided before that experiment is performed. Including additional analytes in the simulation and design function is straight forward as long as they are network components, or the mapping from a network component to a desired analyte is known. Thus, we shall add a menu that allows for the NMR spectra of different analytes to be predicted.

The second limitation is the trafficking solutions provided are not fully narrowed down by the regulation (e.g. feedback inhibition) actually present. Currently, upper and lower bounds for individual fluxes, reaction directionality, and aggregate constraints capture thermodynamic feasibility and some regulatory details, which in turn, limit the number of alternative solutions. In practice, however, separated network elements can "communicate" by feedback loop cascades and/or mutual induction and repression behavior, and these features are not fully captured by capacity or other constraints. This limitation can be addressed by adding integer or other constraints, which will be the subject of future work. However, from another standpoint, it may be still worthwhile to examine a larger solution set because they imply what regulation needs to be present *and* also absent. Both types of information are of interest and can be the goals of cell engineering at the molecular level.



Caption: Glucose

☑ Caption Not Shown

Type: Substrate (Source)

Synonym: Glc

Formulus: C6H1206

MW: 180

Border: Round Rectangle

Note:

carbon source

Stoichiometric Coefficients in Reaction

Reactant Name	Variable	Formula	Stoi. Coeff.
● Glucose	y(2)		-1
● G6P	y(1)		1

Fluxes

Exchange coefficient: 0.5

□ Reversible reaction

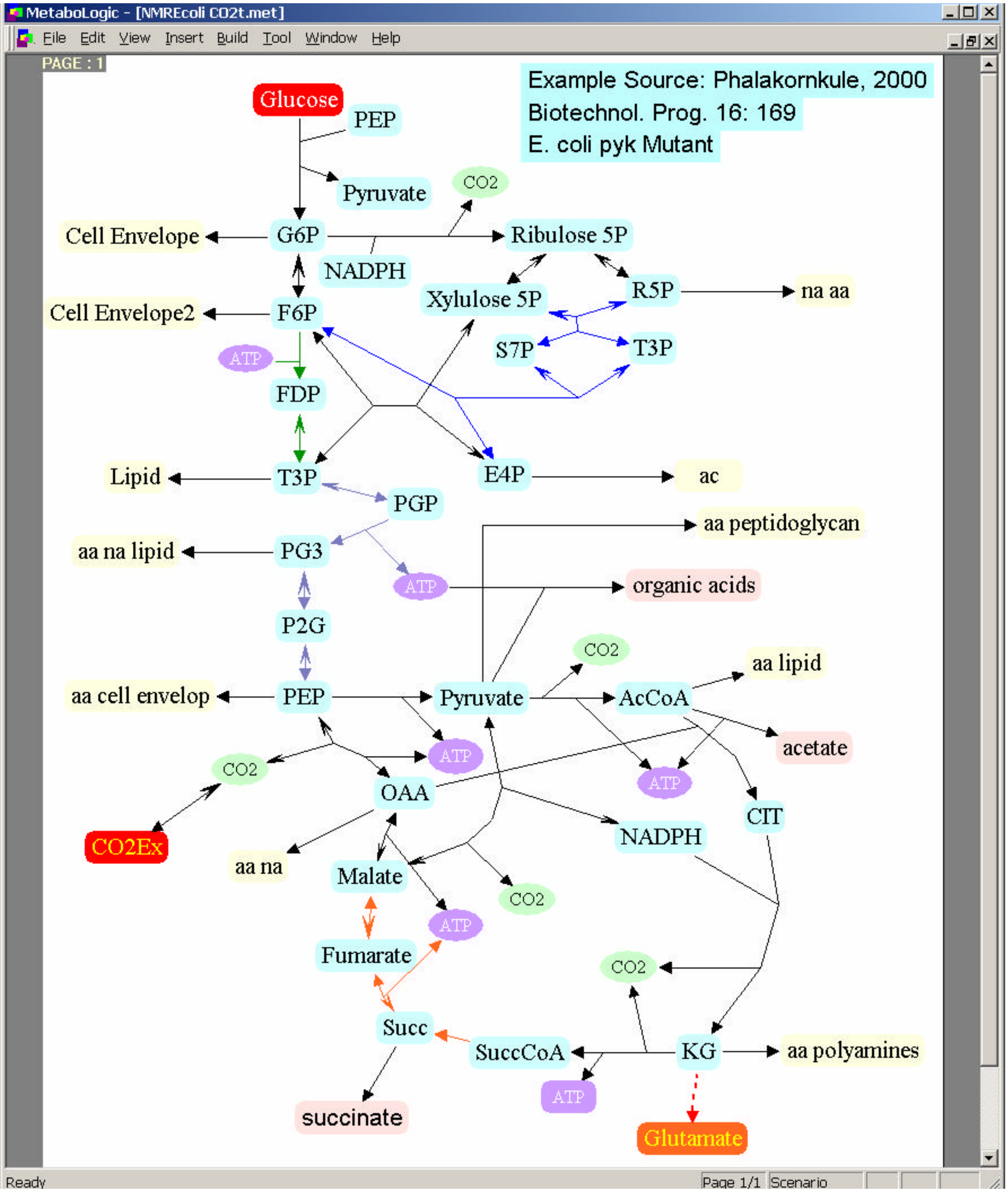
Lower boundary of flux: 0

Upper boundary of flux: 20

Enzyme Information

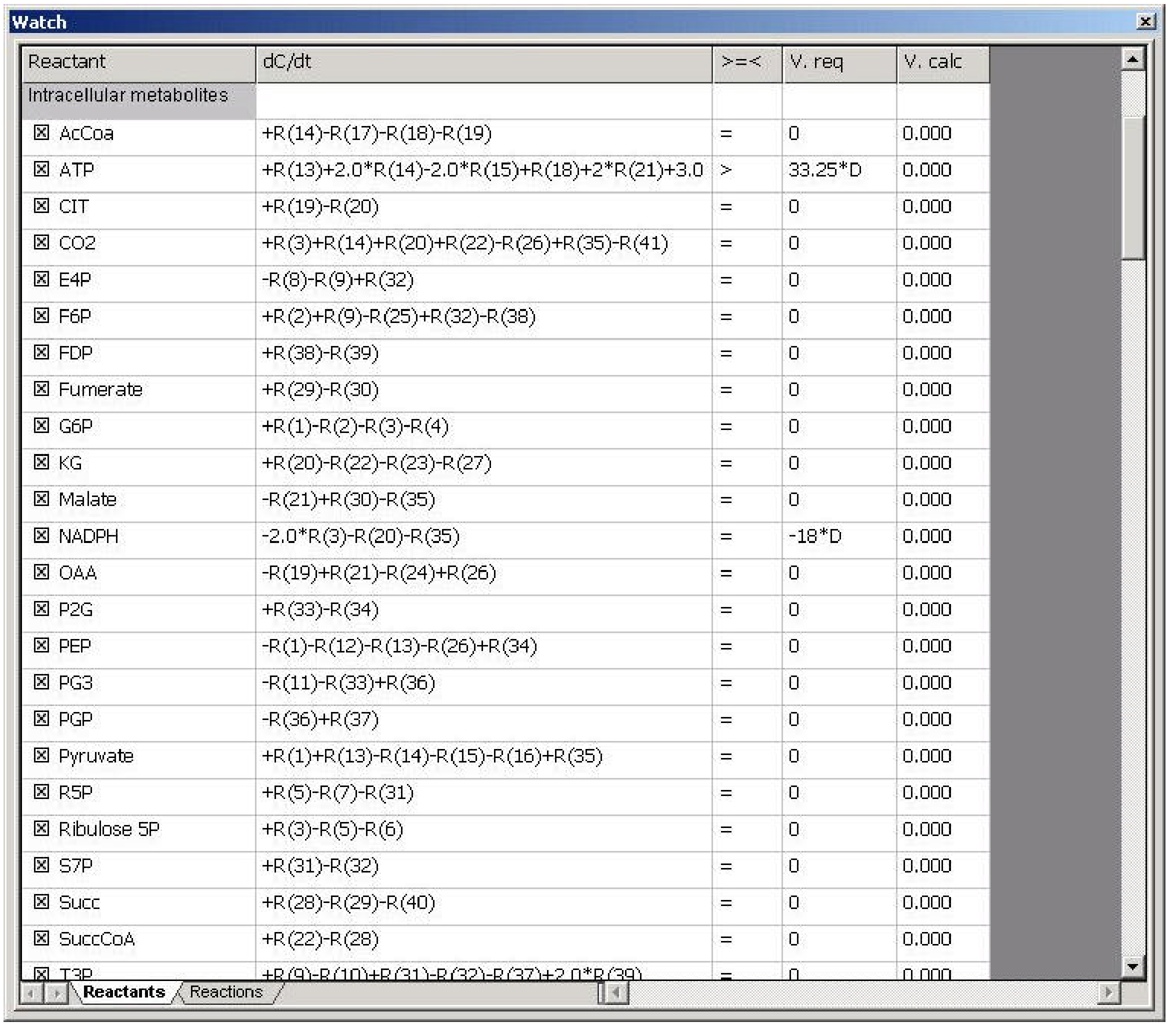
Name:

Figure 4-1 Initial steps in model construction. The user can choose reactant or reaction from toolbar and add them into the network by typing a name in a text box. Glucose and glucose-6 phosphate are the reactant and products shown. By double clicking reactant (a) and reactions (c), property dialogs will pop up for user to edit properties and to customize information (b, d).



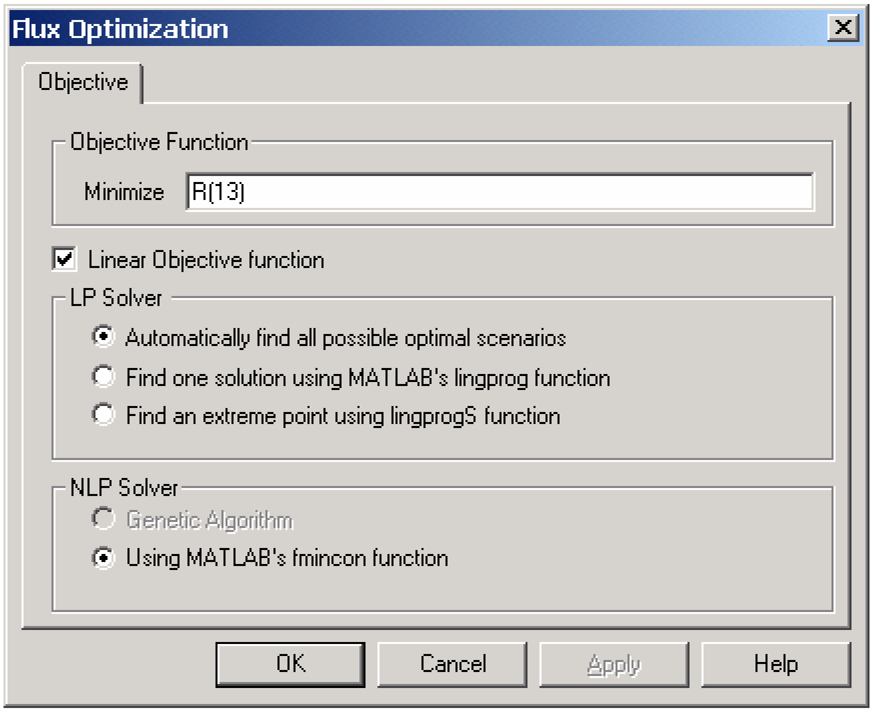
Example Source: Phalakornkule, 2000
Biotechnol. Prog. 16: 169
E. coli pyk Mutant

Figure 4-2 The window of MetaboLogic that displays the model network. This *E. coli* network is the basis for the example problem of finding flux distributions and NMR spectra.



Reactant	dC/dt	>=<	V. req	V. calc
Intracellular metabolites				
☒ AcCoa	+R(14)-R(17)-R(18)-R(19)	=	0	0.000
☒ ATP	+R(13)+2.0*R(14)-2.0*R(15)+R(18)+2*R(21)+3.0	>	33.25*D	0.000
☒ CIT	+R(19)-R(20)	=	0	0.000
☒ CO2	+R(3)+R(14)+R(20)+R(22)-R(26)+R(35)-R(41)	=	0	0.000
☒ E4P	-R(8)-R(9)+R(32)	=	0	0.000
☒ F6P	+R(2)+R(9)-R(25)+R(32)-R(38)	=	0	0.000
☒ FDP	+R(38)-R(39)	=	0	0.000
☒ Fumerate	+R(29)-R(30)	=	0	0.000
☒ G6P	+R(1)-R(2)-R(3)-R(4)	=	0	0.000
☒ KG	+R(20)-R(22)-R(23)-R(27)	=	0	0.000
☒ Malate	-R(21)+R(30)-R(35)	=	0	0.000
☒ NADPH	-2.0*R(3)-R(20)-R(35)	=	-18*D	0.000
☒ OAA	-R(19)+R(21)-R(24)+R(26)	=	0	0.000
☒ P2G	+R(33)-R(34)	=	0	0.000
☒ PEP	-R(1)-R(12)-R(13)-R(26)+R(34)	=	0	0.000
☒ PG3	-R(11)-R(33)+R(36)	=	0	0.000
☒ PGP	-R(36)+R(37)	=	0	0.000
☒ Pyruvate	+R(1)+R(13)-R(14)-R(15)-R(16)+R(35)	=	0	0.000
☒ R5P	+R(5)-R(7)-R(31)	=	0	0.000
☒ Ribulose 5P	+R(3)-R(5)-R(6)	=	0	0.000
☒ S7P	+R(31)-R(32)	=	0	0.000
☒ Succ	+R(28)-R(29)-R(40)	=	0	0.000
☒ SuccCoA	+R(22)-R(28)	=	0	0.000
☒ T3P	+R(9)-R(10)+R(31)-R(32)-R(37)+2.0*R(39)	=	0	0.000

Figure 4-3 Equations automatically produced by software based on the model “ picture” and information inputted during model construction (see Fig. 1). The first column is the metabolite. The second, third and fourth columns present the balance equation for the metabolite. The “ >=<” column denotes the constraint type, where “ >” and “ <” denote “=” and “=” , respectively.



Flux Optimization

Objective

Objective Function

Minimize R(13)

☑ Linear Objective function

LP Solver

● Automatically find all possible optimal scenarios

○ Find one solution using MATLAB's lingprog function

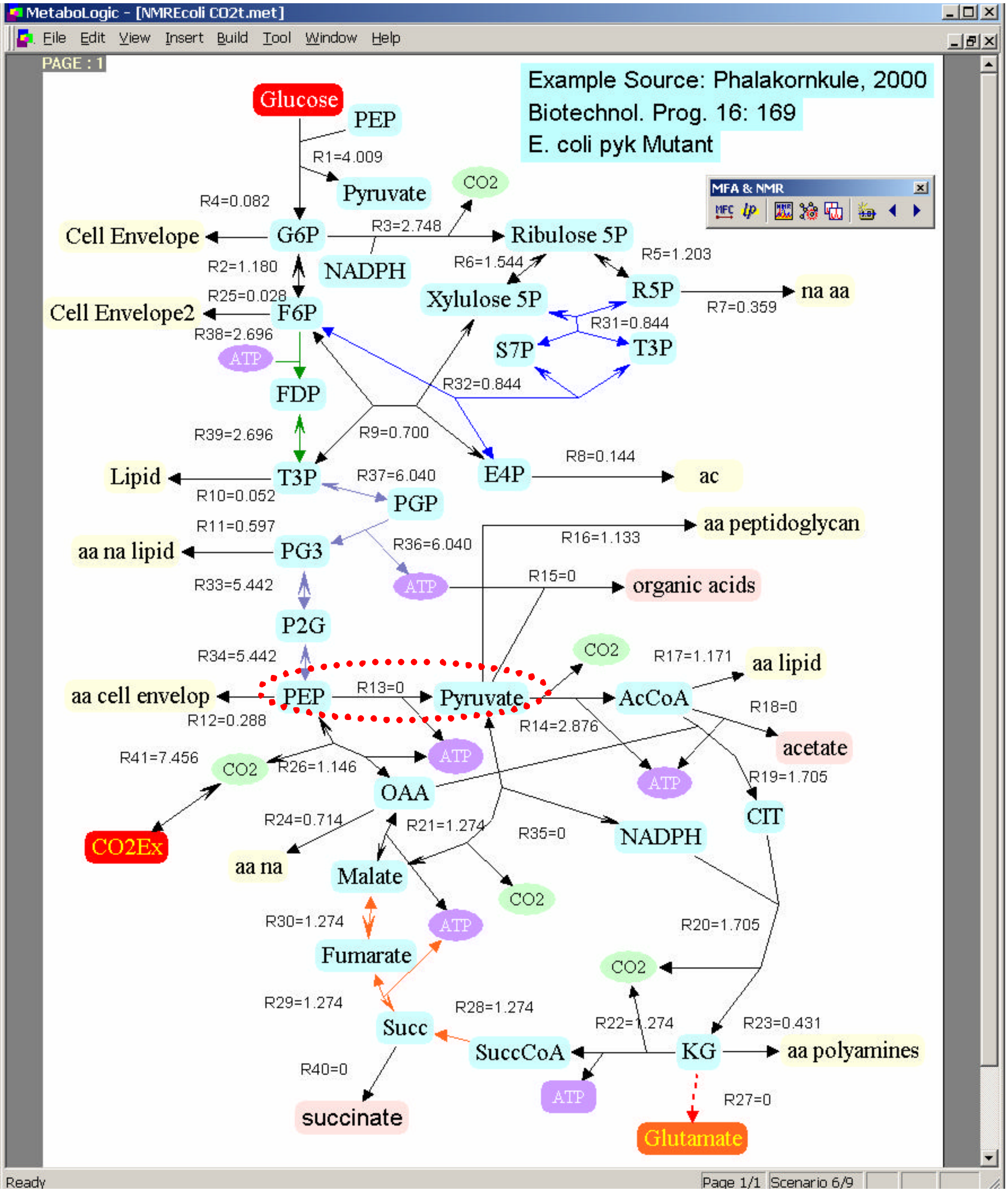
○ Find an extreme point using lingprogS function

NLP Solver

○ Genetic Algorithm

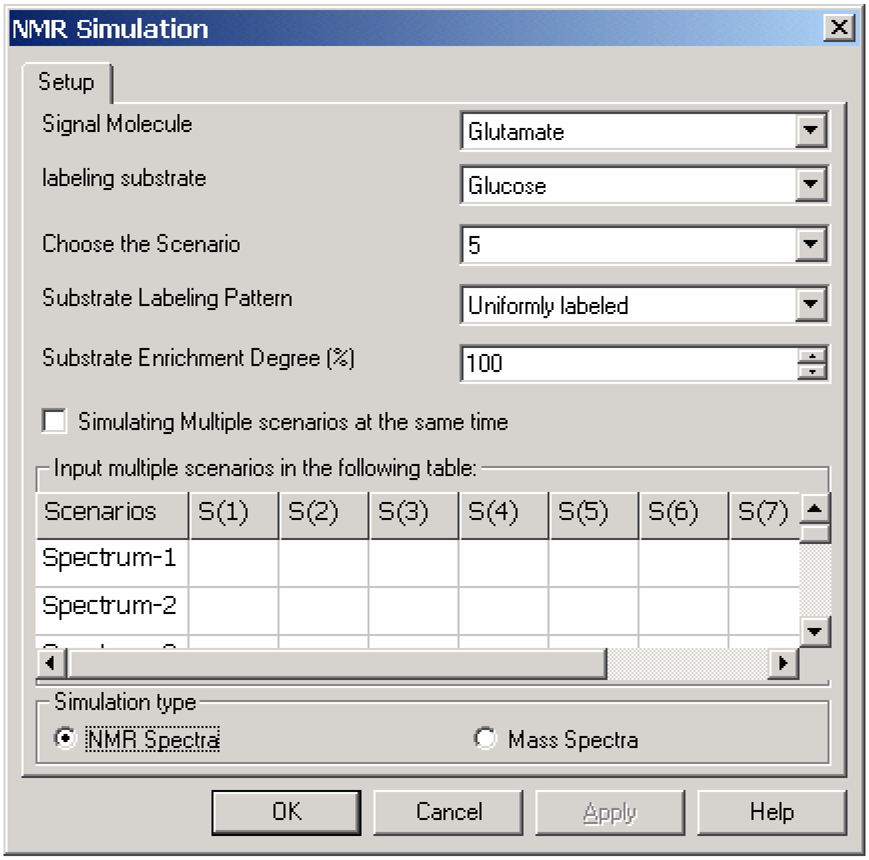
● Using MATLAB's fmincon function

Figure 4-4 Dialog box used to engage flux balance analysis (linear programming or convex analysis). All flux solutions can be found by inputting an integer into the objective function (top) or a specific objective can be sought (bottom).



Example Source: Phalakornkule, 2000
Biotechnol. Prog. 16: 169
E. coli pyk Mutant

Figure 4-5 Flux distribution of one scenario that satisfies the balance equations, inequality constraints, and the objective of minimizing pyruvate kinase-catalyzed flux.



Signal Molecule: Glutamate

labeling substrate: Glucose

Choose the Scenario: 5

Substrate Labeling Pattern: Uniformly labeled

Substrate Enrichment Degree (%): 100

□ Simulating Multiple scenarios at the same time

Input multiple scenarios in the following table:

Scenarios	S(1)	S(2)	S(3)	S(4)	S(5)	S(6)	S(7)
Spectrum-1							
Spectrum-2							

Simulation type

● NMR Spectra ○ Mass Spectra

Figure 4-6 The dialog for setting up the NMR simulation. Two operative modes are possible. The spectrum associated with one flux distribution and a particular labeling of glucose can be computed. Alternately, a subset of all feasible flux distributions can be explored. By vectorizing the spectra and looking at different combinations of exchange coefficients, the labeled glucose will be determined that produces spectra that maximally differ from each other.

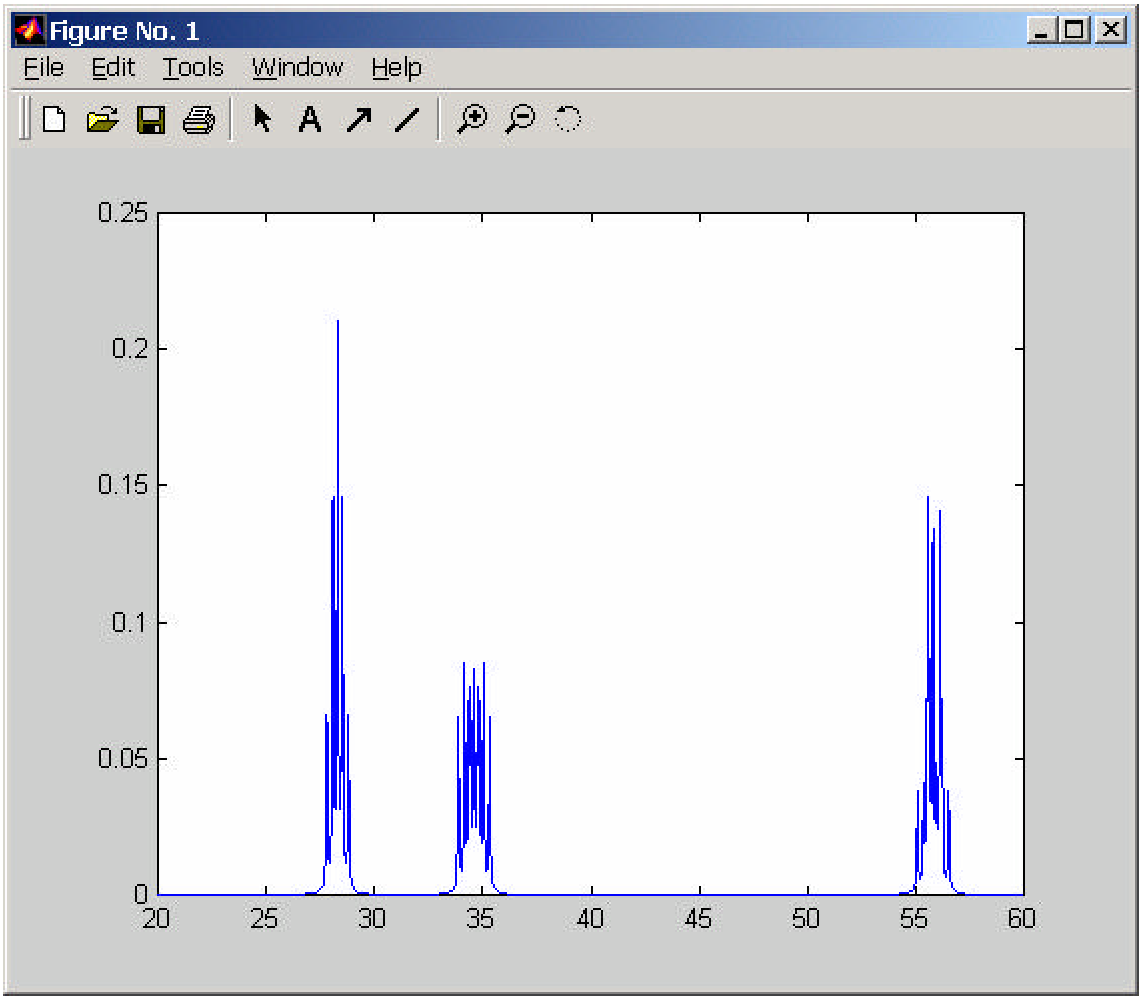
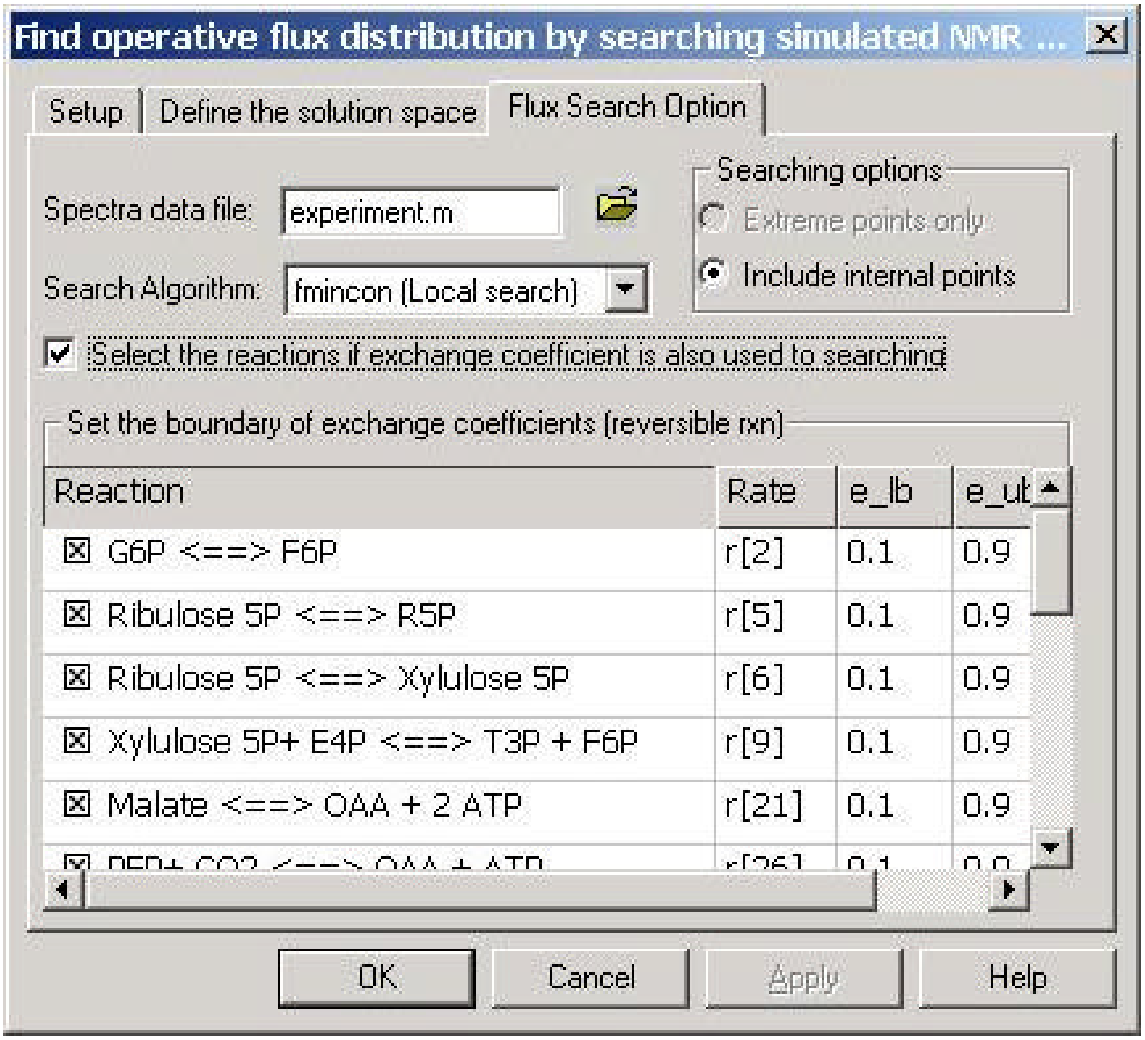


Figure 4-7 The simulated spectra of the central glutamate carbons. From left to right, the multiplets result from C3, C4, and C2 of glutamate (C2 bears the amino group).


Find operative flux distribution by searching simulated NMR ...
Setup
Define the solution space
Flux Search Option
OK
Cancel
Apply
Help

Spectra data file: experiment.m

Search Algorithm: fmincon (Local search)

Searching options
○ Extreme points only
● Include internal points

☑ Select the reactions if exchange coefficient is also used to searching

Set the boundary of exchange coefficients (reversible rxn)

Reaction	Rate	e_lb	e_ub
☒ G6P <==> F6P	r[2]	0.1	0.9
☒ Ribulose 5P <==> R5P	r[5]	0.1	0.9
☒ Ribulose 5P <==> Xylulose 5P	r[6]	0.1	0.9
☒ Xylulose 5P+ E4P <==> T3P + F6P	r[9]	0.1	0.9
☒ Malate <==> OAA + 2 ATP	r[21]	0.1	0.9
☒ PEP+ CO2 <==> OAA + ATP	r[26]	0.1	0.9

Figure 4-8 The dialog window for the flux searching based on inputted NMR data. All combinations of exchange coefficients are searched based on discretizing the bounded domain established by the user. This function allows one to establish what feasible flux and exchange coefficients combinations can account for the experimental data.

Table 4-1 Performance of Different Labeled Glucose Substrates in Distinguishing Flux Scenarios by ^{13}C NMR Analysis

Glucose Label Pattern	Average Difference between NMR spectra*
2- C ^{13}	21.3
1- C ^{13}	11.1
5- C ^{13}	8.12
6- C ^{13}	6.14
4- C ^{13}, 3- C ^{13}, U- C ^{13}	<0.16

* The average difference between NMR spectra simulated from different extremes point of the convex polytope is calculated based on the angle between the vectors describing the NMR spectra features.

5.0 ELIMINATATION OF ACID FORMATION

5.1 Introduction

It has been suggested that *E. coli* and *B. subtilis* have excess glycolytic capacity relative to the Krebs cycle (Majewski and Domach, 1990, Jensen and Pederson, 1990, Marr, 1991) when grown in glucose containing growth medium. In chapter 2, we found pyk deletion might be a good genetic engineering strategy to reduce acid by- production.

In this chapter, the acid formation in an *E. coli* pyk and the *B. subtilis* pyk mutant were compared. The results demonstrate that acid formation is significantly reduced in the PYK mutant of *E. coli* as well. This substantial reduction in acids is accompanied by an increase in CO_2 production. Moreover, the growth rate of the *E. coli* pyk mutant is only slightly lower than the wild-type. This is in contrast to *B. subtilis* pyk mutant, which exhibited a significant reduction in growth rate (Fry *et al*., 2000). This difference might reflect a dramatic increase in PEP pool of the *B. subtilis* pyk mutant. The *E. coli* pyk mutant did not show a substantial increase in the PEP pool.

5.2 Materials and Methods

5.2.1 Cells and Growth Medium.

Construction of *B. subtilis* pyk mutant was described in Fry *et al.* (2000). The *E. coli* mutant strain (PB25) lacking activities of both PYKI and PYKII was used in these experiments. The strain was generously provided to us by Dr. Fernando Vale (Ponce *et al.*, 1995). The medium was M9 (Maniatias *et al.* 1982). The concentrations of glucose used in each experiment are noted in the text.

5.2.2 Cultivation & Measurement of Cell Mass, Glucose, & Extracellular Acid..

A fermentation vessel with 2 liter capacity (Applikon, Austin, TX) was used. An on-line data acquisition system collected data for CO_2 evolution rate from a Dycor mass spectrometer (Ametek, PA) using a standard gas mixture for calibration. The air flow rate was set at 2.5 l/min. Optical density was measured off-line using a Lambda 6 Perkin Elmer spectrophotometer (Perkin Elmer, Norwalk, CT), (1 OD_{660} = 0.36 g cell dry weight/l). The glucose concentration was measured enzymatically. The concentrations of organic acids (lactate, acetate, acetoin, formate, fumarate, malate, pyruvate, succinate, and citrate) were measured using a HPLC gradient Module (Bio-Rad, CA) equipped with a UV/VIS detector (Goel *et al.*, 1993).

5.2.3 Intracellular Enzyme & Metabolite Assays

For PEP assays, cell extracts were prepared using the formic acid extraction method. A sample was withdrawn via a syringe and quickly filtered (5-10 s) followed by immersion of the filter into liquid nitrogen (20 s). The metabolites were extracted from the filtered cells by immersing the filter in formic acid as detailed by Fisher & Magasanik (1984) and assayed as described in Goel *et al.* (1999).

5.3 Results and Discussion

Growth experiments were first conducted in shake flasks. The volume of the growth medium in the flasks was 10% of total volume. Our initial experiments with a low glucose concentration (2.2 g/l) in the growth medium showed very similar growth rates and final cell densities for the wild-type and the mutant cultures (Figure 5-1). Additionally, very low levels of acids were found in both cultures of wild-type and the mutant.

Since acid formation is substantially more pronounced in high glucose cultures, an experiment was conducted with a higher initial glucose concentration. Figure 5-2 is the growth curve of the wild-type and the mutant in batch cultures with an initial glucose concentration of 9 g/l. This figure shows that the growth rates of the wild-type and the mutant are similar in the shake flasks. but higher cell densities are attained for the mutant than the wild-type culture. The major difference between the wild-type and the mutant was, in the levels of acids found in their cultures. Figure 5-3 demonstrates that the pyk mutant produces significantly lower levels of acetic acid. The increase in cell density (1 OD 660 = 0.36 g/l of cells based on our calibration) of the mutant culture can not compensate for the dramatic reduction in acetic acid. Thus, it is likely that CO_2 formation may be substantially higher in the mutant cultures than the wild-type.

In order to assess how glucose carbon was distributed between cells and metabolic by products, an experiment was conducted in a 2 liter fermentor (800 ml working volume). In this experiment, the CO_2 evolution rate was monitored. Figures 4a and 4b compare the cells and CO_2 produced by the wild-type and *E. coli* mutant, respectively. The initial glucose concentration in this batch experiment was 10 g/l. Data shown in Figure 4a depict that the growth rate of the mutant was about 15% lower than the wild-type initially but the mutant reaches a higher cell density. For the wild-type culture an exponential increase in cell density occurs for the first 200 minutes of the growth. This is consistent with the rapid rise in CO_2 evolution rate during the same period (Figure 5-4b). For the time period after the first 200 minutes the rate of increase in cell density of wild type culture decelerates. During this later period acid formation is significant (see Figure 4c) and CO_2 evolution rate is decelerated.

The cell density of the mutant also increases exponentially for the first 200 minutes but it is followed by an extended linear increase. This trend is also observed in CO_2 evolution rate of mutant culture (Figure 5-4a). The CO_2 evolution rate (Figure 5-4b) shows an initial exponential increase followed by an extended period of relatively high but constant CO_2 evolution rate. Figure 4b also shows a substantially higher CO_2 production in the mutant than the wild-type cells. While CO_2 production is higher in the mutant, acetate and pyruvate are present at a drastically lower level in the mutant than the wild-type cultures (Figure 5-4c). Figure 5-4d indicates that in contrast to the wild-type culture, glucose was completely utilized in the mutant culture. The drop in acid concentration following the exhaustion of glucose may be due to the consumption of acids towards the end of growth phase.

The carbon balances were performed for the fermentor experiment at two points during fermentation. The results are summarized in Table 1. Overall, the results indicate a lower rate of acid formation in the mutant (Figure 5-4c) and higher cell mass (Figure 5-4a) and CO_2 (Figure 5-4b) production. These results confirm our original hypothesis that acid formation should be substantially lower in the pyk mutants (Goel *et al.*, 1995). The results of Table 1 also demonstrate that a substantially higher fraction of the glucose consumed evolved as CO_2 in the mutant culture.

This higher CO_2 yield for the mutant appears to be in contrast with the results reported by Ponce *et al.* (1998) who found equal CO_2 yields (mole CO_2/mole glucose) for the wild-type and the PB25 strain (lacking activities of both PYKI and PYKII). Ponce *et al.* (1998), did not measure acid for the wild-type or the mutant in their experiments. Our results (Figure 5-4 and Table 5-1) are for a medium glucose concentration of 10 g/l versus 2 g/l in the experiments performed by Ponce et al. (1998). In experiments with 2.2 g/l glucose in the medium (Figure 5-1), we have found very little or no acids in both wild type and mutant cultures. Furthermore, given that both cultures attained essentially similar cell density, the CO_2 evolution rate may not be significantly different in the two cultures at low glucose concentrations. Thus, our results at low glucose concentration are in agreement with the results of Ponce et al. (1998). However, the results of Figure 5-4 and Table 5-1 indicate that major differences between the wild-type and the mutant become discernable at high glucose concentrations.

The pyk mutation has the advantage of allowing the cell's strategy of phosphorylating metabolites to prevent leakage to be harnessed. This advantage contrasts other acetate flux-reducing approaches. Diaz-Ricci *et al.* (1991) abolished the activities of acetate kinase and acetyl phosphotransferase, the enzymes responsible for acetate production in *E. coli.* The deletion eliminated acetate production but resulted in significant formation of pyruvate. Another interesting strategy entails diverting excess glucose flux to products that are less toxic than

In NMR spectra data, the relative signal intensity of different singlets and multiplets is the information that can be used to compare between two spectra. If a vector collects the signal intensity to represent a NMR spectrum, then the angle between such two vectors will be a good indication of the difference of spectra.

$$d_{ij} = \cos q_{ij} = \frac{S_i \cdot S_j}{\|S_i\| \cdot \| S_j \|} \qquad \text{(Eq. 4-8)}$$

$$S = \begin{bmatrix} f_1 & f_2 & \dots & f_n \end{bmatrix}$$

where f_i is the intensity of ith peak in the spectra. || S_i || denotes the second norm of vector S_i. Such an angle function performs better than the distance between spectra, S_i and S_j, because the absolute signal intensity also depends on the sample concentration and instrument sensitivity.

A mass distribution vector (MDV) will present the mass spectrum. Since the summation of all components of MDV is one, the distance between two MDV is a good indication of the difference. Because of natural abundance of 13C, the raw data of MS should be adjusted before using to search the flux space.

The experiment data of NMR spectra can also be present as such a vector. Finding the operative flux distribution can be solved by searching the convex polytope to find a matching between simulated NMR spectra and experiment one. Such a problem is a constrained global nonlinear optimization problem. We adopt GAOT (generic algorithm optimization toolbox) (Houck et al. 1995) to solve the optimization problem. GAOT can be downloaded on http://www.ie.ncsu.edu/mirage/GAToolBox/gaot/. We also use the "fmincon" function from the MATLLAB optimization toolbox to solve the problem. Although, fmincon is a local optimization tool, it converges much faster than GAOT (more than several orders). Additionally, fmincon can find the most of the solution accurately, and a good initial guess will significantly reduce the computation time. Further work on formulating a better objective function and a better initial may help increase the efficiency of searching.

acetate. Dhedia *et al.* (1991) engineered *E. coli* to overproduce glycogen, as a way of diverting the excess glucose flux. Although glycogen levels increased by several-fold, the acetate production remained essentially unaltered. Chou *et al.* (1994) mutated *ptsG*, the gene encoding an enzyme in the glucose phosphotransferase system. The *ptsG* mutation reduced glucose uptake by about 20%. This decreased glucose uptake led to a 20-40% decrease in acetate yield, 15% lower growth rate, and an overall 50% increase in productivity of β-galactosidase (as a model recombinant protein). Finally, Farmer and Liao (1997) were able to achieve acetate reduction by increasing the activity of two anaplerotic pathways. *E. coli* strains which overexpressed the PEP carboxylase gene and depressed the glycoxylate bypass were shown to reduce acetate production four fold with respect to the wild type strain. However, there was no increase in cell density indicating the possibility of glucose conversion to other metabolic by-products.

Finally, we measured the intracellular PEP concentration for the mutant and the wild-type of *E. coli* (Figure 5-5). The intracellular concentration of PEP is typically very low in bacteria. The PEP concentration for both wild-type and the mutant remained below the detection limit of about 0.06 mM. This was in sharp contrast to our recent finding with the *B. subtilis* pyk mutant (Fry *et al.*, 2000). The *B. subtilis* pyk mutant exhibited a very high intracellular PEP concentration of about 30-fold higher than the wild-type level.

The difference in PEP concentration between *E. coli* and *B. subtilis* pyk mutants is very interesting. This difference may be due to the differences in the path of formation of oxaloacetate in these bacteria. In *E. coli*, PEP is converted to oxaloacetate by the enzyme PEP carboxylase. Excess oxaloacetate may produce malate, which in turn can yield pyruvate via malic enzyme activity. This two step path may result in a "trickle" around the pyk mutation that could have just enough mass action potential to be thermodynamically feasible. Thus, since *E. coli* can convert PEP to oxaloacetate or pyruvate, it may not accumulate high levels of PEP. In *B. subtilis* oxaloacetate is formed by pyruvate. Hence, in a *B. subtilis* pyk mutant, there are no routes for converting the PEP formed by glucose transport to TCA cycle metabolites or pyruvate. This difference in the oxaloacetate-forming reactions may explain the build up of PEP in *B. subtilis* pyk mutant but not in *E. coli* pyk mutant.

5.4 Chapter Summary

In summary, while the residual acid production in the mutant (even with relatively high initial glucose concentration) is low and we are pleased with the results, some further improvements may result from blocking a possible small flux that bypasses the PYK bottleneck through the activities of the PEP carboxylase and the malic enzyme. It is likely that incorporation of a malic enzyme deletion into the pyk double mutant of *E. coli* might further reduce acid formation.

Table 5-1 Carbon balance at different time in the batch growth of wild type and mutant

Strain (culture time)	$\frac{\text{C in cells}^{(2)}}{\text{Total C used}^{(1)}}$	$\frac{\text{C in acids}^{(3)}}{\text{Total C used}}$	$\frac{\text{C in CO}_2{}^{(4)}}{\text{Total C used}}$
Wild Type (320 min)	29.3	29.5	33.6
Wild Type (600 min)	23.7	29.3	37.3
Mutant (320 min)	37.0	9.2	52.8
Mutant (600 min)	27.7	8.1	52.0

Total Carbon used was calculated using the difference between initial glucose concentration (zero time) and glucose concentration at the time indicated in the table.

The carbon content of cell was estimated based on (1 OD_{660} = 0.36 g cell/l) and cell carbon content of 0.45.

Carbon in acids (acetate, pyruvate) was calculated based on data of Figure 4c.

Carbon in CO_2 was calculated by the measurement of the area under the curve in Figure 4b.

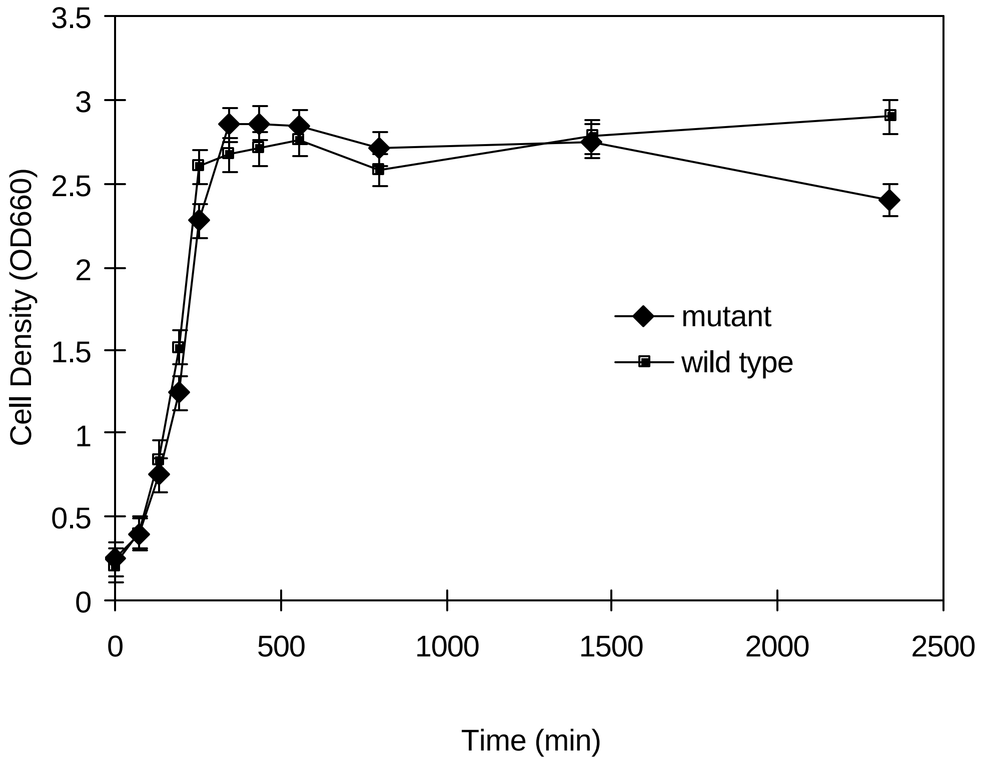


Figure 5-1 The growth of wild type and pyk mutant of E. coli with feed glucose concentration of 2.2 g/l. The experiment was repeated 3 times. The error bars represent the average variation in optical density measurements

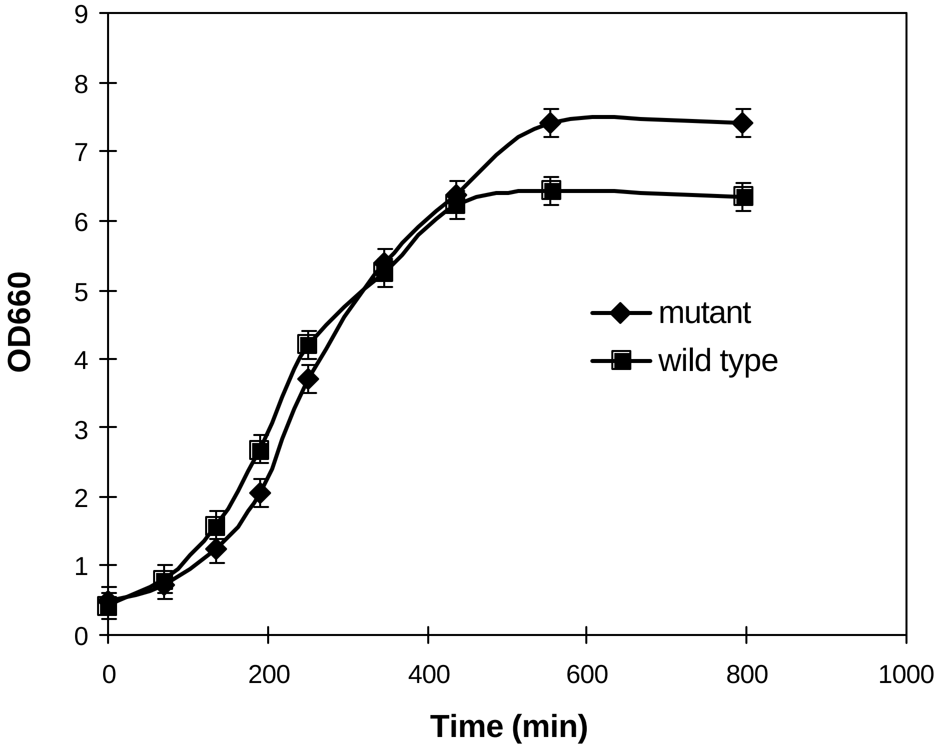


Figure 5-2 Similar to Figure 1.1 except that the glucose concentration is 9 g/l

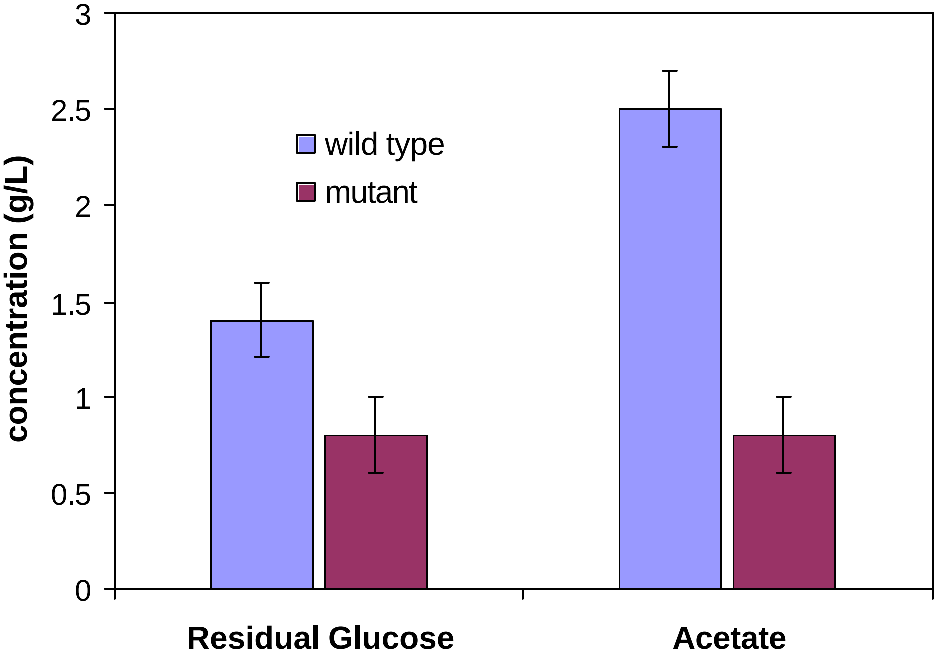


Figure 5-3 Residual glucose and acetate concentration for the shake flask with initial glucose concentration of 9 g/l (Figure 2). The error bars indicate the variation in measurements of residual glucose and acetate, which were preformed three times.

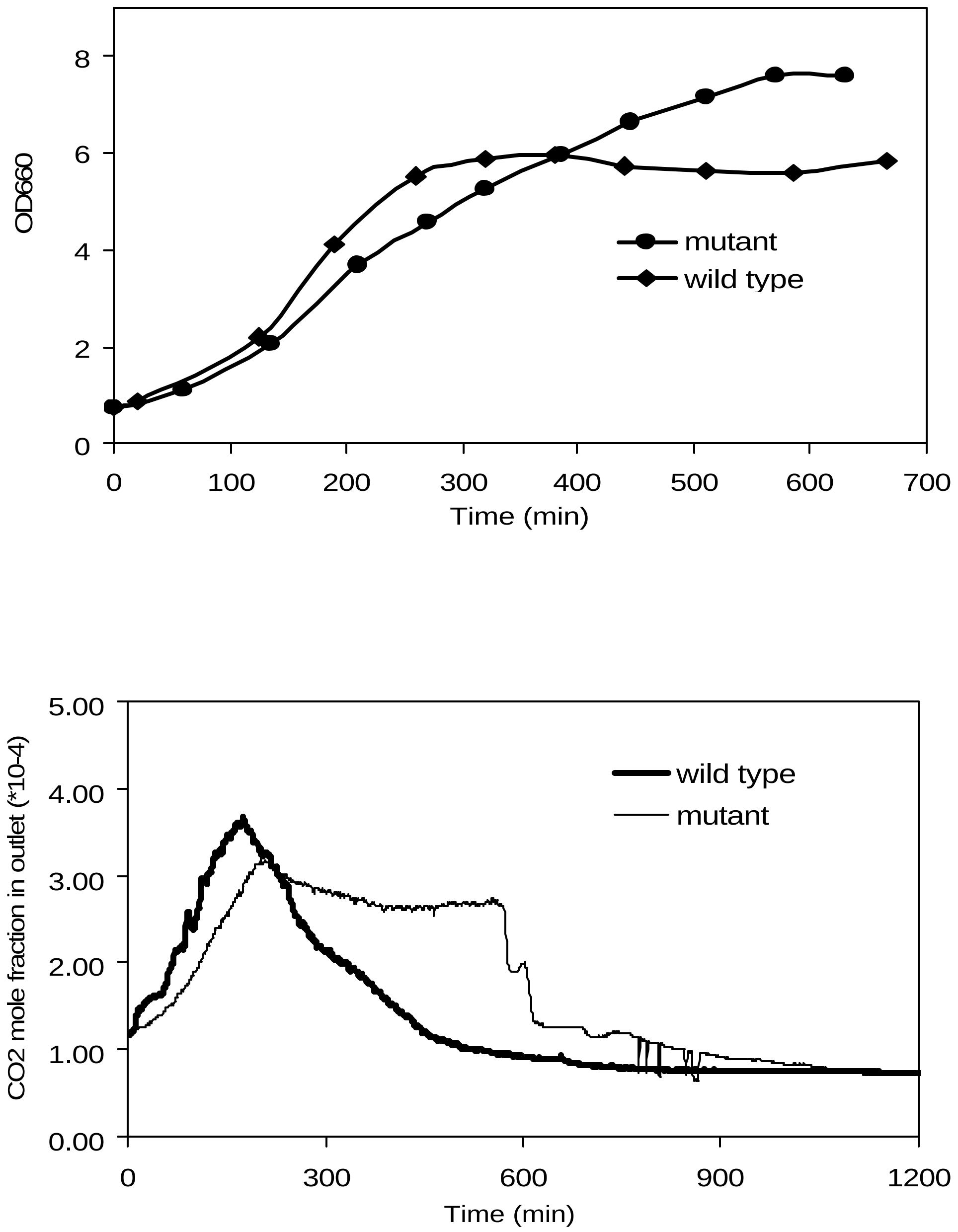


Figure 5-4 Growth of the wild type and pyk mutant of *E. coli* in the fermentor with initial glucose concentration of 10 g/l. (a) optical density, (b) CO$_2$ evolution rate.

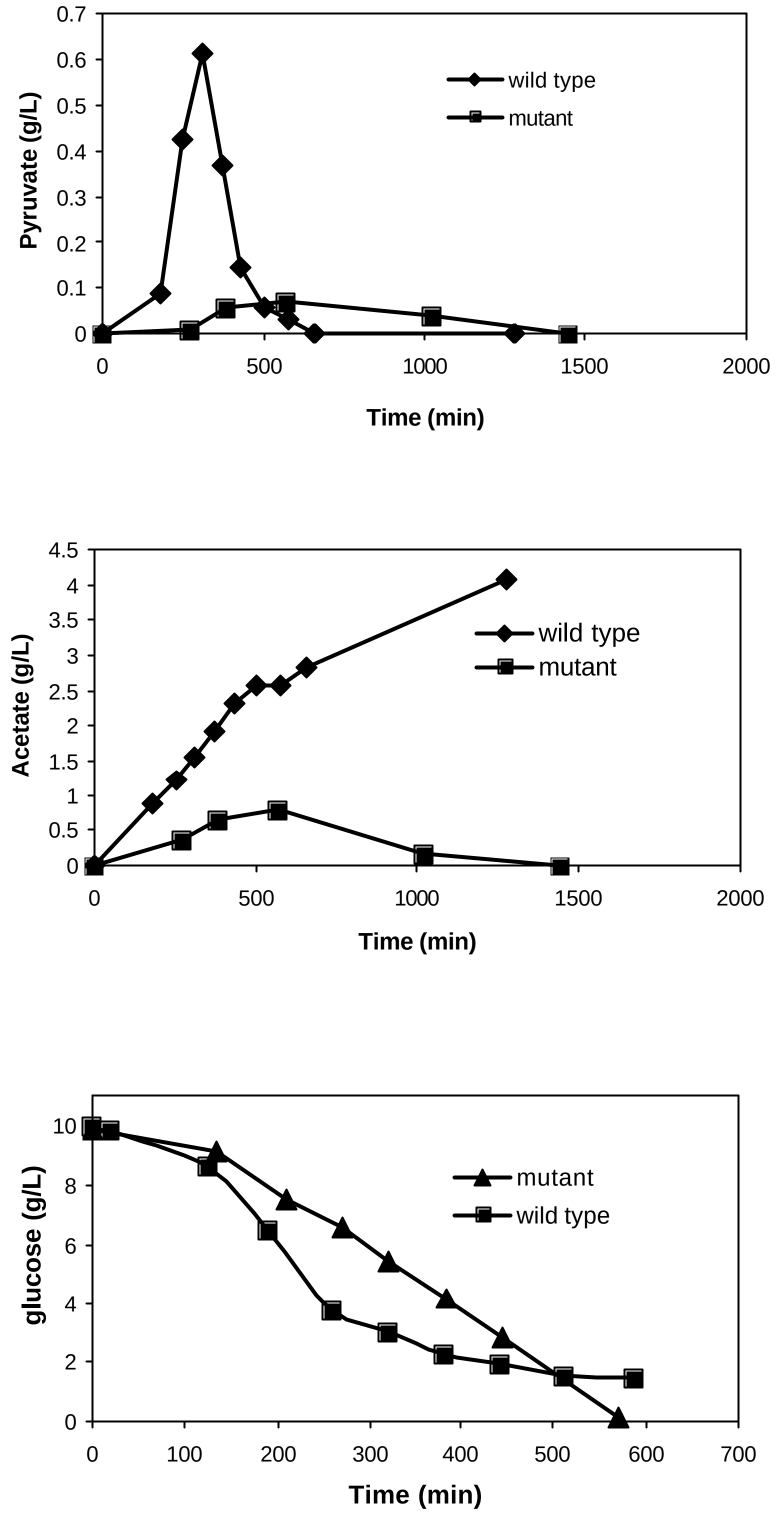


Figure 5-4 Growth of the wild type and pyk mutant of *E. coli* in the fermentor with initial glucose concentration of 10 g/l. (c) acid formation and (d) residual glucose.

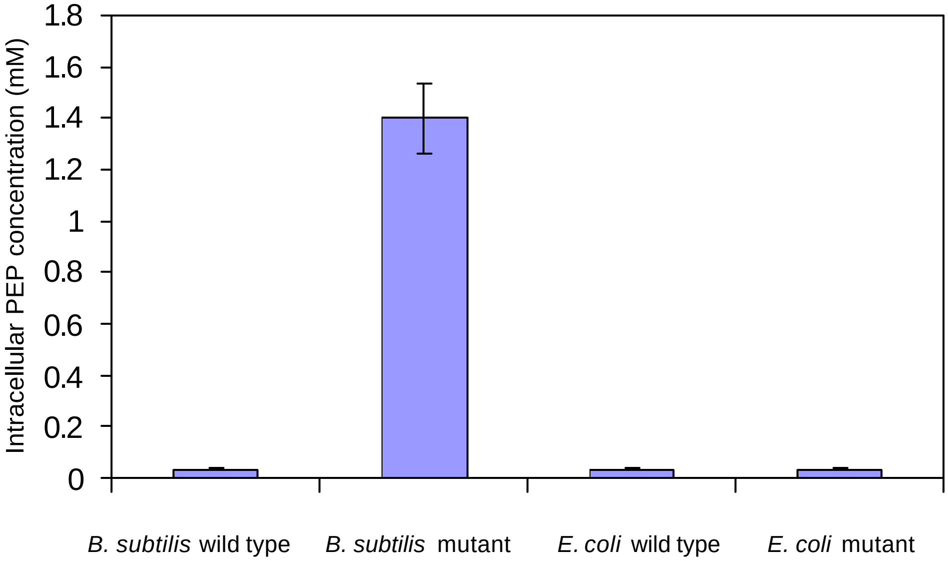


Figure 5-5 The intracellular PEP concentration for the wild type and pyk mutants of *E. coli* and *B. subtilis.* Error bars indicate the variations of two measurements.

6.0 ENHANCEMENT OF FOLIC ACID PRODUCTION

6.1 Introduction

Metabolic engineering has focused on increasing the yield of both low quantity/high value products and commodities. One product that has received less attention in the United States metabolic engineering community is folic acid, a member of the B class of vitamins. Folic acid is not produced by birds and mammals so it is an important dietary requirement. Food supplements and microbial production in the digestive track are the main sources of folic acid for mammals and birds. Production by intestinal bacteria is often insufficient and can be impaired by antibiotic intake. A number of recent developments, such as studies that show increased folic acid intake can lessen the incidence of birth defect have raised the demand for folic acid. To meet current demand, the bulk of commercially sold folic acid is produced by chemical synthesis. This synthetic source is relatively expensive based on raw material costs and the low process yield, hence, interest has emerged on increasing the yields associated with potential biological routes.

In the first part of this chapter, convex analysis method embedded in *MetaboLogic* will be used to explore metabolite trafficking options for enhanced folic acid production. The, phase planes will be used to represent the flux distribution of several potentially high yielding metabolic mutants. Finally, the most viable genetic engineering strategy predicted for enhanced folic acid will be discussed.

The second part of this chapter will focus on experimental verification of some of the model prediction. Specially, we will show that in according to the model prediction, PYK mutation is an efficient metabolic engineering strategy for enhanced folic acid production.

6.2 Stoichiometric Model and Methods

6.2.1 Model and Analysis Methods

The biochemical reactions in central carbon metabolism of *E. coli* as well as candidate reversible reactions are shown in Figure 6-1. The networks are based on our prior work (*4*).

The metabolite balance equations and constraints have been previously described (*4*). Cell mass compositions data for *E. coli* are from previous work (*6, 7*). Other constraints are derived from NADPH and minimum ATP requirements (*8*). A reference specific growth rate equal to 0.4 h^{-1} is used to generate flux units that can be scaled when other growth rates are considered.

The overall stoichiometry for folic acid production can be described as:

$$2PEP + E4P + 1.5\ 3GP + KG + 11\ ATP + R5P + 4\ NADPH \rightarrow$$
$$FA + Pyr + GLdh + 1.5CO2$$

where PEP (phosphoenolpyruvate), E4P (erythrose-4-phosphate), 3GP (3-phosphoglyceric acid), KG (oxoglutarate), R5P (ribose-5-phosphate), Pyr (pyruvate) and Gycolaldehyde (GLdh) are intermediate metabolites present in the central carbon metabolic pathway. FA (folic acid) and CO_2 are cellular products. This lumped reaction is added to the model, assuming the leakage of intracellular metabolites in this pathway is negligible when the strain is optimized for folic acid production.

Using the *MetaboLogic* model construction tool, all the reactants, reactions and constraints are inputted in less than an hour. The objective function, maximization of folic acid production, is used to find all the flux scenarios that lead to high folic acid production. Computation takes less than one minute on a PIII-900 Hz computer.

6.2.2 Experimental Methods

The *E. coli* wild-type strain (JM101) and mutant strain (PB25) lacking activities of both PYKI and PYKII was used in these experiments. The strain was generously provided to us by Dr. Fernando Vale (*9*). The medium was M9 (*10*). An initial glucose concentration of 4 g/L was used. Optical density was measured off-line using a Lambda 6 Perkin-Elmer spectrophotometer (Perkin-Elmer, Norwalk, CT), (1 OD660 = 0.36 g cell dry weight/l).

250 mL Shake flasks were used in the cultivation experiments. Folic acid concentration is measured using microbial method (Difco Manual 11th). Samples were collected at different culture time for *E. coli* wild type and mutant culture. These samples are filtered thought 0.2 µm filter, and diluted 40 to 200 times based the estimated folic acid concentration. Stock cultures of the test organism, *L. casei subsp. Rhamnosus* ATCC® 7469 were prepared by stabbing inoculation into prepared tubes of Lactobacilli Agar AOAC. The cultures were incubated at 35-37°C for 18-24 hours. The cultures were stored in the refrigerator at 2-8°C. Transfers were made monthly. The inoculums for assay were prepared by subculturing from a stock culture of *L. casei subsp. rhamnosus* into a tube containing 10 ml Micro Inoculum Broth. These inoculums were incubated at 35-37°C for 16-18 hours. Under aseptic conditions, the tubes were centrifuged to sediment the cells and the supernatant is decanted. Cells were washed 3 times in sterile single-strength Folic Acid Casei Medium. After the third washing, the cells were resuspended in 10 ml sterile single-strength medium and were diluted 100 times. 20 µL of this suspension was used to inoculate each of the assay tubes. In each test tube, 2.5 mL double-strength Folic Acid Casei Medium, 0.5 mL of sample and 2 mL distilled water were added to the test tube to make the total volume 5 mL. The growth response of the assay tubes was read turbidimetrically after 18-24 hours incubation at 35-37°C. It is essential that a standard curve be constructed for each separate assay. Autoclave and incubation conditions can influence the standard curve readings and cannot always be duplicated. The standard curve may be obtained by using folic acid at levels of 0.0, 0.1, 0.2, 0.3, 0.4 and 0.5 ng per assay tube (5 ml).

6.3 Results and Discussion

6.3.1 Computation Results

To scout the complete yield horizon, we first found all the solutions that can satisfy the metabolic constraints for the growth of *E. coli* in glucose minimum medium *while* maximizing folic acid flux. We found 32 different flux distributions. The highest carbon yield predicted for folic acid [mol C in folic acid/mol C from glucose] is 0.087. Certain aspects of the flux solutions will be highlighted further here because of their potential implementation via metabolic engineering and a tractable relationship exists to what is known about folic acid synthesis. One aspect is the role of PYK activity (i.e. Figure 1; r_{18} flux). As noted earlier, reducing the activity of PYK could elevate at least one folic acid precursor, PEP, as well as divert carbon normally lost to acids (*11, 12*) into other more useful metabolic products.

One phase plane for folic acid production by wild-type *E. coli* is shown in Figure 6-2, where PYK-catalyzed flux is the independent variable. Folic acid synthesis increases linearly as the PYK-catalyzed flux decreases to zero. This limiting result suggests that a mutant deficient in PYK activity may exhibit elevated folic acid production.

Because PEP is an important folic acid precursor, another potential strategy for elevating folic acid synthesis is to replace the phospho-transferase system (PTS) with a glucose permease. In *E. coli*, one mole of PEP is converted to pyruvate when one mole of glucose is transported into the cell via PTS. Replacing the PTS by glucose permease, which transports one mole of glucose at cost of one mole of ATP, could thus reduce PEP use and increase intracellular concentration.

A comparison of the PTS and PYK activity-modulating strategies and abbreviated pathways are shown in Figure 3. Based on the wild-type phase plane (Figure 6-2) and the stoichiometric potential of a PTS-deficient mutant, models of PYK- and PTS-deficient strains were constructed. The phase planes for the PYK- and PTS-deficient strains are shown and compared to the wild-type in Table 1.

The phase plane for the PYK-deficient strain (Table 6-1) is simply a subspace of the wild-type because one additional constraint was added (i.e. Figure 1, $r_{18} = 0$). Therefore, the maximum production potential of folic acid for the wild-type cannot be surpassed by a PYK mutant. However, the solution shows a contracted glucose uptake window within which, folic acid synthesis potential is high. Moreover, the phase plane for the single PYK mutation provides a bench mark for comparison to the folic acid production that may be achieved by the PTS deletion.

The PTS-deficient mutant is predicted (Table 6-1) to have greater folic acid production potential compared to the wild-type (and the PYK-deficient mutant). The maximum production rate is predicted to equal 1.81 mmol g^{-1} h^{-1}. Moreover, this high potential production rate could also be achieved with a lower to comparable glucose uptake rate (9 mmol g^{-1} h^{-1}), which indicates that the potential molar and carbon yields are higher than that associated with the wild-type or PYK-deficient strains. Indeed, based on the fluxes (not shown), the molar maximal potential yield for the PTS-deficient strain is 0.211 mol folic acid/mol glucose, which is more than double the wild-type's or PYK-deficient mutant's yield potential.

Although the results show that PTS mutation might be an effective strategy for increasing folic acid production, stoichiometric calculations need to be combined with kinetic and other considerations in order to fully assess different metabolic engineering strategies. Indeed, a recent study (*13*) reported that PTS-deficient strains can exhibit very slow growth rates. Slow growth could potentially limit the utility of the PTS-deletion strategy for folic acid production especially if the product was labile and regulation is such that product synthesis is growth-related.

6.3.2 Experimental Results and Discussion

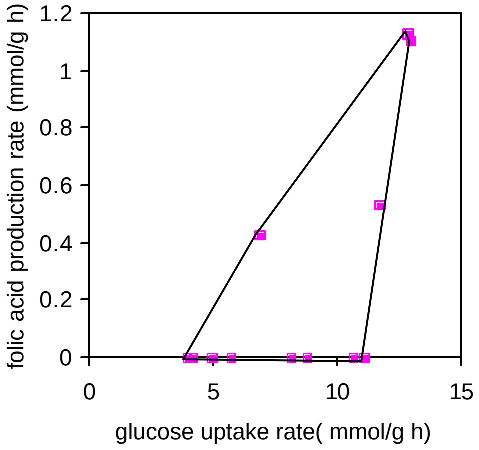
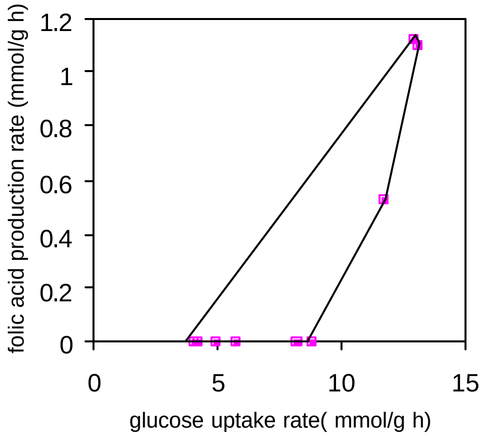
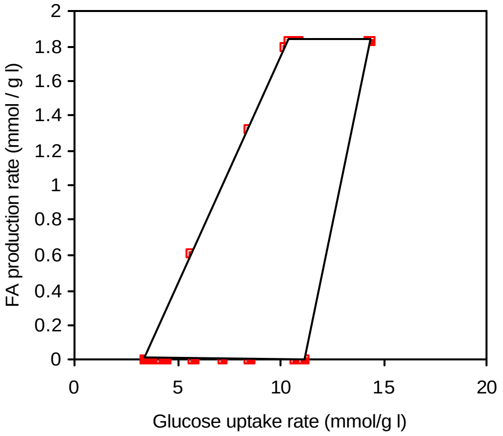
Based on desirable growth properties (*12*) and stoichiometric potential, we have first focused on examining the utility of using a PYK-deficient mutant of *E. coli* for producing folic acid. Figure 6-4 shows that the growth rates of the two strains were similar. Additionally, folic acid accumulation in the medium appears to parallel growth. However, the folic acid production by the PYK-deficient mutant (0.27mg/L at the end of growth) was significantly higher as compared to the wild-type strain (0.05mg/L at the end of growth). Thus, based on theoretical yield potentials and experiments, PYK-deficiency shows promise as a starting point for enhancing the folic acid production by bacteria.

Our other initial work has also suggested (not shown) that after accumulating in the medium, folic acid is consumed by nutrient-starved cultures presumably for the glutamate content. Future work will be devoted to mapping out the product formation and degradation kinetics in more detail in order to further maximize the product yield. Additionally, using what has been learned in *E. coli* to metabolically engineer *Bacillus subtilis* to produce folic acid with high yield is envisioned. In contrast to *E. coli, B. subtilis* is generally regarded as "safe;" hence, *B. subtilis* may be prove to be a more commercially acceptable platform for nutraceutical production.

6.4 Chapter Summary

Computer-aided metabolic flux analysis has suggested how to divert cell raw materials to elevate the production of folic acid in *E. coli.* One of the strategies, pyruvate kinase (PYK) deletion, is predicted to divert resources to increase folic acid production. This flux redirecting strategy is also consistent with elevating the precursors, phosphoenolpyruvate and erthyrose-4-phosphate, which provides the mass action potential to increase the product formation rate. Experimental measurement of folic acid released in the culture medium shows that deletion of pyruvate kinase activity from *E. coli* significantly increases folic acid production. These calculations and experimental results suggest that PYK deletion may be a good initial starting point for further enhancing folic acid production via metabolic engineering..

Table 6-1 Phase plane of different *E. coli* strains

Strain **(Metabolic engineering strategy)**	**Phase Plane of Folic Acid Production vs. Glucose Consumption**
Wild Type	
PYK – Deficient Mutant	
PTS- Deficient Mutant	

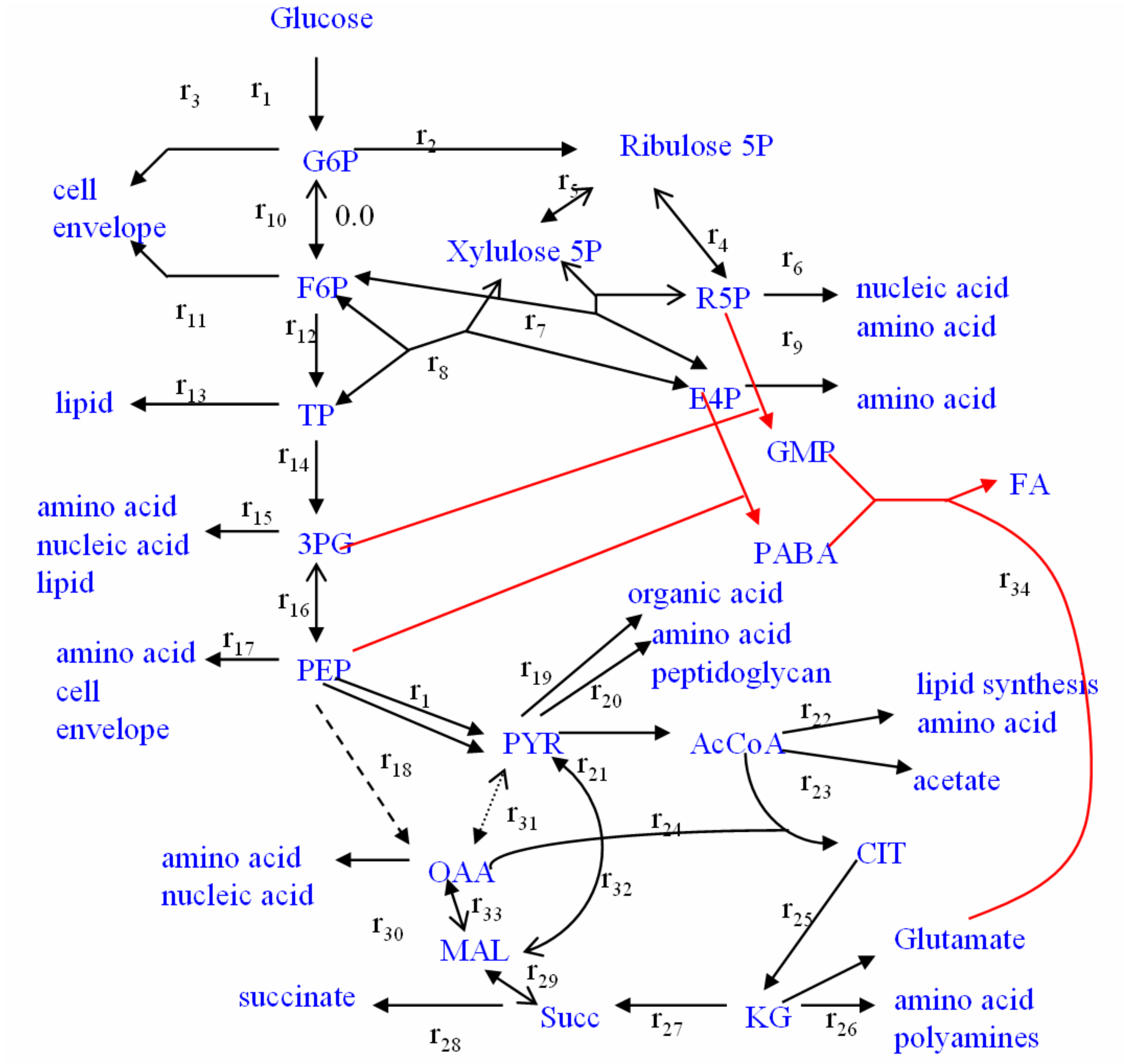


Figure 6-1 The metabolic network of *E. coli*. The two-way arrows represent reversible fluxes with the bold heads showing the constrained net directions

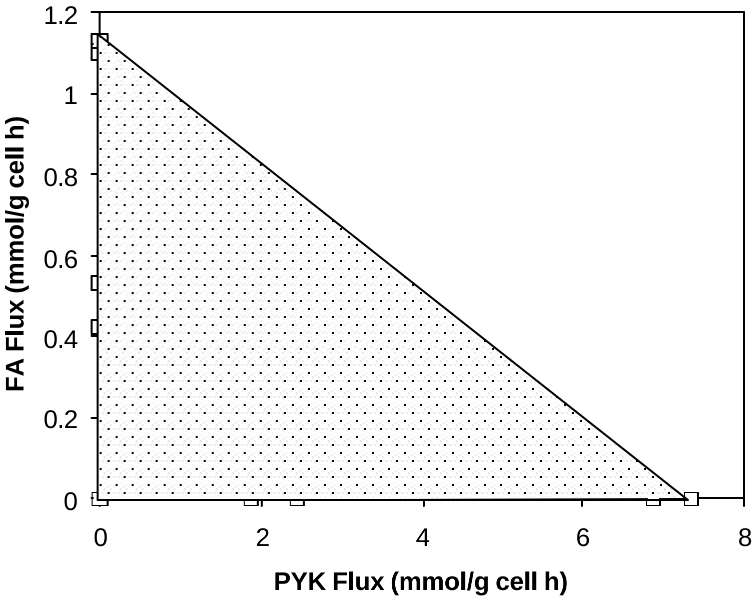


Figure 6-2 Phenotype plane of *E. coli* wild type. Pyk flux is constrained from 0 to 20 mmol g^{-1} h^{-1}.

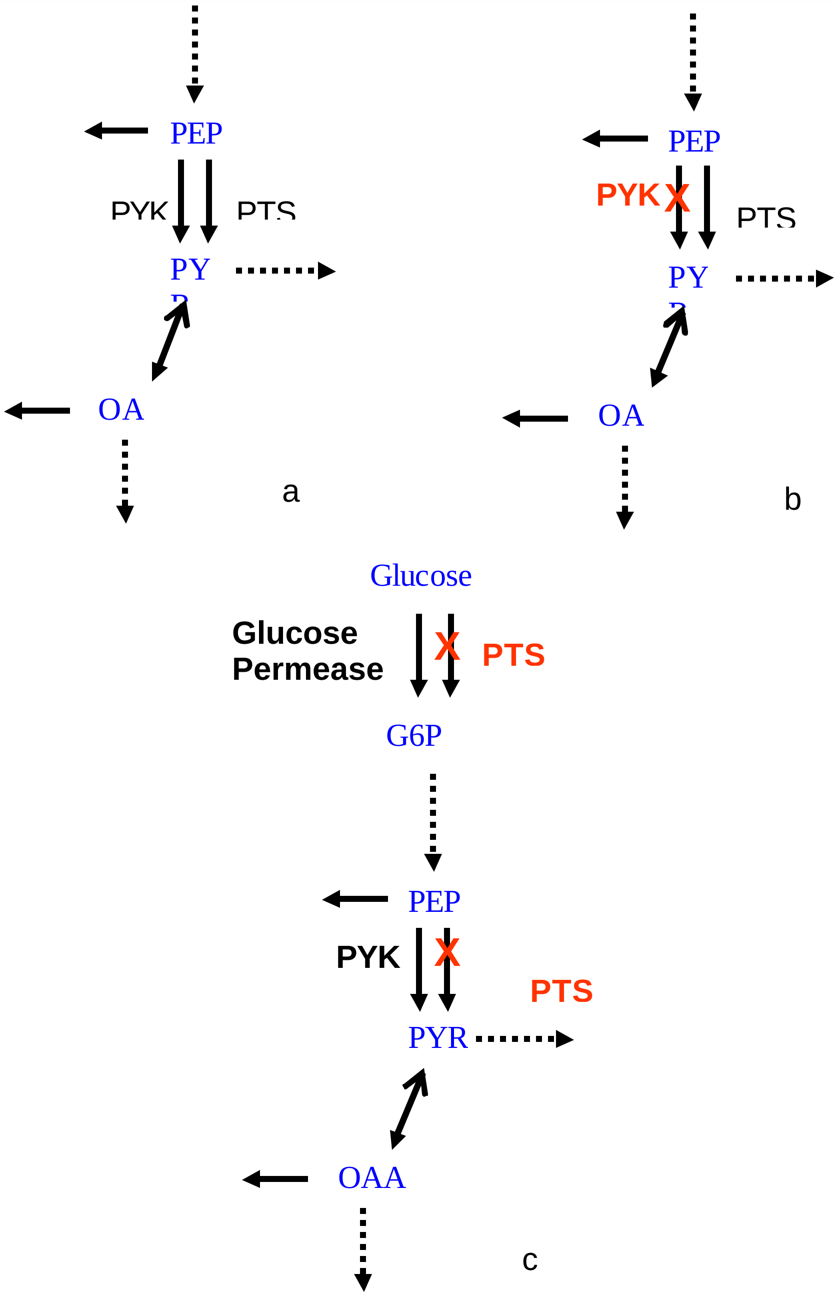


Figure 6-3 Schematic presentation of three metabolic engineering strategies to increase folic acid production and representative phenotypephase planes. The scenarios are (a) *E. coli* wild-type, (b) a mutation that deletes PYK activity, and (c) a mutation that replaces PTS with a glucose permerase..

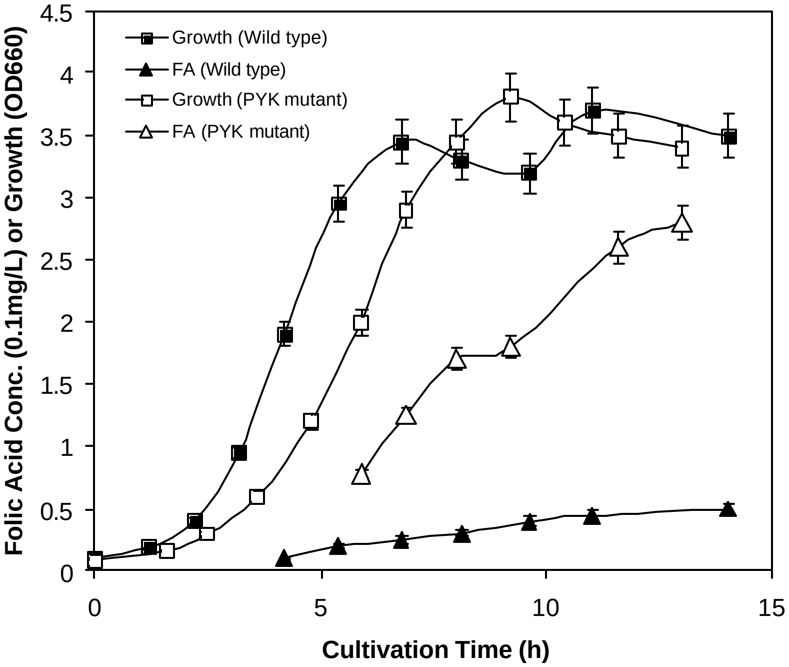


Figure 6-4 Folic acid production in E coli mutant (PB25) and wild type (JM101).

APPENDICES

Appendix A Depth First Search Algorithm

A simple problem (Steuer, 1986) can illustrate the computation procedure.

Max $z = 2x_1 + 2x_2$

s.t. $x_1 + x_2$ £ 4

$x_1 + 2x_3$ £ 8

and x_1, x_2, x_3 ³ $0.$

A graph of the problem is shown in Figure A1. There are six “corner points,” which are more typically termed, “extreme points.” In a linear programming problem where the feasible domain is convex, optimal solutions will lie at one or more corner point (or along an edge). In this case, the objective function is a family of parallel lines where x_2 versus x_1 has a slope equal to –1. Different values of x_2 intercepts and x_3 are possible. By inspection, the points, A, B, C, and D are the extreme points that minimize z.

Reformulating the problem by introducing slack variables to replace inequality constraints with equalities results in

Max $z = 2x_1 + 2x_2$

s.t. $x_1 + x_2 + x_4 = 4$

$x_1 + 2x_3 + x_5 = 8$

and x_1, x_2, x_3 ³ $0.$

The Depth First Search Algorithm (DFS; also known as the last-in-first-out (LIFO) algorithm) involves starting with a node and visiting each of the nodes adjacent to the current node, calling the algorithm on each that has not been visited. Each node corresponds to an extreme point; hence, DFS is a strategic way to identify and search the extreme points for the occurrence of equally optimal values of an objective function. An organized search becomes necessary when the dimensionality of a problem defies graphical solution. The Simplex Method is employed for determining which variables enter and depart the basis set and thus how the nodes connect and branch.

When choosing which edge to explore next, the algorithm always chooses to go ‘deeper’ into the graph. That is, it will pick the next adjacent unvisited vertex until reaching a vertex that has no unvisited adjacent vertices. (For this purpose, the algorithm simply initializes a set of markers, so, we can tell which vertices are visited. If a vertex has several neighbors, it would be equally correct to go through them in any order.) The algorithm will then backtrack to the previous vertex and continue along as-yet unexplored edges from that vertex. After DFS has visited all the reachable vertices from a particular source vertex, it chooses one of the remaining undiscovered vertices and continues the search. This process creates a set of depth-first trees, which together form the depth-first forest.

The search for optimal solutions can be tracked by Tableaus and Figure A1. The Tableau calculations identify extreme points and which basis variable to replace with a nonbasic variable.

Table I

	Z	X_1	X_2	X_3	X_4	X_5	RHS
Z	1	-2	-2	0	0	0	0
X_4	0	1	1	0	1	0	4
X_5	0	1	0	2	0	1	8

The reduced ratio test indicates that x_1 comes in and x_4 departs the basis set.

Table II

	Z	X_1	X_2	X_3	X_4	X_5	RHS
Z	1	0	0	0	2	0	8
X_1	0	1	1	0	1	0	4
X_5	0	0	-1	2	-1	1	4

This is an optimal solution **{4,0,0}** which corresponds to the point **A** in Fig.A1.

From here, it is possible to branch into 2 nodes :

x_2 comes into the basis.

x_3 comes into the basis.

Now examine the branch (1).

Tableau III

	Z	X_1	X_2	X_3	X_4	X_5	RHS
Z	1	0	0	0	2	0	8
X_2	0	1	1	0	1	0	4
X_5	0	1	0	2	0	1	8

Another optimal solution **{0,4,0}** (**D** in Fig. A1).

Further branching can be done as follows:

x_1 comes into the basis – here, the new basis is {x_1, x_5}, which has been already examined in Tableau II – so, this node is fathomed.

x_3 comes into the basis – this is now examined.

Table IV:

	Z	X1	X2	X3	X4	X5	RHS
Z	1	0	0	0	2	0	8
X2	0	1	1	0	1	0	4
X3	0	½	0	1	0	½	4

So, the optimal solution is **{0,4,4}** - this is **C** in Fig A1.

Further branching is as follows:

x_5 comes into the basis – here, the new basis is {x_2, x_5}, which has been already examined in Tableau III – so, this node is fathomed.

x_1 comes into the basis - this is now examined.

Table V:

	Z	X_1	X_2	X_3	X_4	X_5	RHS
Z	1	0	0	0	2	0	8

X_1	0	1	1	0	1	0	4
X_3	0	0	-½	1	-½	½	2

The optimal solution in this case is **B{4,0,2}**

Branching occurs again in the following manner :

x_2 comes into the basis – here, the new basis is {x_2, x_3}, which has been already examined in Tableau IV – so, this node is fathomed.

x_5 comes into the basis – here, the new basis is {x_1, x_5}, which has been already examined in Tableau II – so, this node is fathomed.

Both nodes have been fathomed so the search returns to Node 2 (after Tableau II). Here, the new basis is {x_1, x_3}, but this has already been examined in Tableau V. Thus, no further branching is possible and we have obtained all the possible solutions of the LP. The Depth-First Search (Branch & Bound Scheme) presented in the Tableaus is also summarized in Figure A.2.

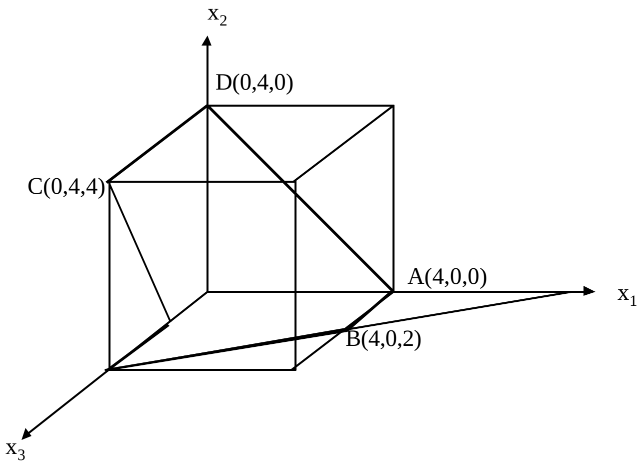


Fig A1 – Graphical Representation of the given LP
(the solutions are indicated)

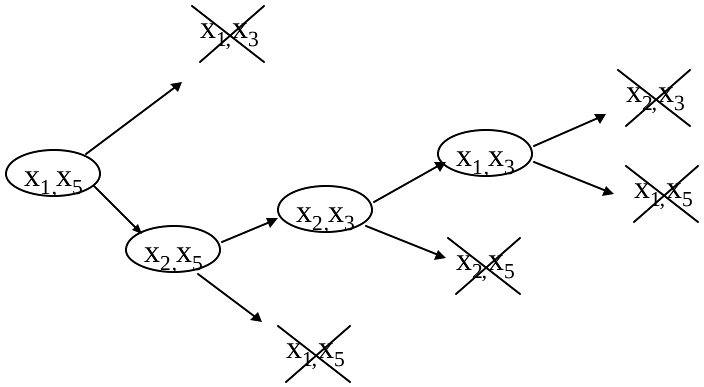


Figure A2. Branch and bound scheme used to solve the example problem. The current basis variables are shown in the circles. ‘X’ indicates that a node has been fathomed.

Algorithm

The algorithm is a typical depth first search algorithm. The distinction of this algorithm is that the graph is built as long as the searching goes on. In contrast, a common depth first search takes a built graph as input.

There are 3 arrays used in the searching, E [], finish [], and dad []. E collected alternative optimal vertices. Finish [i] is a Boolean variable to tell us if all the adjacent points to E [i] have been searched or not. Dad[i] save the index of the dad point, from which E[i] was searched, and for the starting point, the dad is set to be –1. The indices of all arrays start with 0.

1. Use simplex method to find starting optimal extreme point S. Put S in the array E, here S is the first element in the array (E [0]). And dad [0] = -1 and finish [0] = false.
2. Check the extreme points connecting with a E [i] (i =0,…). If one point has the same value of objective function and it is not in the Array E. Add this point to the end of array E, which is now the nth point in array E. Set finish [n] = false and dad [n] = i. Then start to search the neighboring point of E[n].
3. If for point E [i], there are no point found to be added to array, set finish [i] to true, The point in E array indexed by dad [i] is chosen to search the neighborhood. If

finish [dad [i]] is already true (already finished searching), use dad's dad, until back to E [0].

4. If every finish [0] is true, the algorithm finishes and all the optimal vertices are saved in array E.

Five m-files are used for convex analysis. LinprogT, lpex, PHASEI, PHASEII, and PHASEIII. The PHASEI and PHASEII are adopted from Jeff Stuart, Department of Mathematics at University of Southern Mississippi. These two pieces of code implements a two phase simplex method. Only PHASEIII.m implements the searching algorithm, which is given as followed.

Table A-1 phaseiii.m

```
function [z,xbasicTable,ibasicTable,ienter,iter,PCOL,OPTEST,CYCTEST] = phaseiii (A, b, c, ibasic );
%PHASEIII searchs other optimal extreme points
%The first optimal point is found by PHASEII with ibasic as the selected base,
%and xbasic as the values of x.
%
%See also PHASEI and PHASEII.
%Written for Matlab version 5.0.
%
%Written by Tao Zhu, Department of Chemical Engineering,
%University of Pittsburgh, PA 15219. March, 2001.
%tzzt@yahoo.com
%                    Written for MATLAB version 5.3
%                    Author:             Tao Zhu
%                    Email:              tzzt@yahoo.com
%                    date:                    March.03.2001
%                    Department of Chemical Engineering,
%                    University of Pittsubrgh, Pittsburgh, PA 15219.
%                    Copyright to Tao Zhu.   All right reserved.

[m,n]=size(A);
PCOL=[];
ienter=[];
iter=0;
cycle=0;
CYCTEST=0;
X=zeros(1,n);
J=X;
tol=0.0000000000001;
ztol=0.0000000000001;
J(ibasic)=ones(1,m);
K=[1:n];
inon=K(~J);
B=A(:,ibasic);
xbasic=B\b;
%save the basis in the table
xbasicTable=xbasic'; %xbasic is column vector, BUT xbasicTable is row oriented.
ibasicTable=ibasic;  %ibasic is row vector, AND ibasicTable is row oriented.
iSolution=1;
bFinished=0;
dads = 0; %the first basis has no dad

z=c(ibasic)*xbasic;
if m<n;
  X(ibasic)=xbasic;
  Cred=c(inon) - (c(ibasic)/B)*A(:,inon);
  OPTEST=1;
  loop =1;
  dad = -1;
  while loop ==1;
    if max(Cred) >= 0; %means other optimal solution is found
      %must find new basic, for Cred=0 and has not been selected into the table.
      pp=logical(find(Cred>=0));
     %       inonSelect = ZT(inon,pp);
      inonSelect = ZT(inon,pp);
      [row,col]=size(inonSelect);
                 %for each of possible new basis, found the one may switch with it.
      newBasisfound=0;
      for i=1:row;
           ienter=inonSelect(i);
           PCOL=B\A(:,ienter);
```

Table A-1 (continued)

```
    if PCOL <= 0 , OPTEST = 0;
    loop = 0;
else
  testJ = J;
      testJ(ienter)=1;
    TESTROWS=find(PCOL > ztol);
  TESTCOL=PCOL(TESTROWS);
  %another loop to pick the one that should be switched out from the original bases
    [minrat,j]=min(xbasic(TESTROWS)./TESTCOL);
    if minrat <=0, cycle = cycle+1;
           if cycle > m;
           disp('Algorithm terminated due to excessive cycling.')
           disp('Restart algorithm from phase II using a perturbed')
           disp(' RHS vector b and the current basis.')
           disp(ibasic)
           CYCTEST=1;
           break
           end
    else
           cycle = 0;
    end
  iexit=ibasic(TESTROWS(j));
                testJ(iexit)=0;
  testJ=logical(testJ); %don't change the state of J and ibasic at this time
  testBasic = K(testJ);
  %Check if the ibasic is already in the table
  alreadyHas=0;
  for p=1:iSolution;
     if (all(testBasic ==ibasicTable(p,:)))
       alreadyHas=1; %already in the ibasicTable
       break;
     end
  end
  if (alreadyHas==0); %new basis found ! good
           xbasic=xbasic - minrat*PCOL;
           X(ibasic)=xbasic;
    X(ienter)=minrat;
           X(iexit)=0;
    %z=z + Maxcost*minrat;
     J = testJ; %since it is good J
           ibasic=K(J);
           inon=K(~J);
           B=A(:,ibasic);
    xbasic=X(ibasic)';

     test = c(ibasic)/B;
     if test~=inf; %good test! B maybe not full ranked
            newBasisfound=1;
           Cred=c(inon) - (c(ibasic)/B)*A(:,inon);
         %add new basis to the table
           xbasicTable=[xbasicTable;xbasic'];
           ibasicTable=[ibasicTable;ibasic];
         iSolution=1+iSolution;
       bFinished= [bFinished; 0];
       if dad==-1;
          dads=[dads; iSolution-1];
       else
          dads=[dads; dad];
       end
```

Table A-1 (continued)

```
        break;
      end
     end
    end % end if
   end % end of for loop
   loop = newBasisfound;
   if (newBasisfound==0); %current starting basis finished searching
    %Get the dad
    if (dad==-1); %means the last one in the table is finished search
     bFinished(iSolution)=1;
     dad=dads(iSolution);
    else
     bFinished(dad)=1; %current basis (dad) finish searching
     dad=dads(dad); %check current basis' dad now
    end
    %if dad==0, means the last basis under check is the root basis.
    if dad~=0; % there are still some basis in the table can be used to search
     loop = 1;
     %make the next test;
     looping= 1;
     while looping;
      if dad==0;
       loop = 0 % already searched every basis stored in the ibasicTable
       break;
      end
      if bFinished(dad)==1 %current basic is already searched, get its parent;
       dad = dads(dad);
         looping = 1;
      else
         ibasic = ibasicTable(dad,:);       %row vector
         xbasic = xbasicTable(dad,:)';      %column vector
         X=zeros(1,n);
         J=X;
         J(ibasic)=ones(1,m);
         K=[1:n];
         inon=K(~J);
         X(ibasic)=xbasic;
         B=A(:,ibasic);
       test =  c(ibasic)/B;
         if test == inf;
        looping = 1;
        dad = dads(dad);
       else
        Cred=c(inon) - (c(ibasic)/B)*A(:,inon);
        looping = 0;
       end
      end % end of while
     end % end of if dad~=0;
    else % search is finished when " finish ==1 "
     loop =0;
    end
   else
    dad = -1;
   end
  else
   loop=0;
  end
 end %loop
end % i
xbasicTable=xbasicTable';
```

Appendix B Atom Mapping Matrices Method (Zupke, 1994)

Specific activity vector of n carbon molecule has n elements. The nth element of a metabolite contains the fractional isotope enrichment of the nth carbon. Examples are:

$$GLC = \begin{bmatrix} GLC(1) \\ GLC(2) \\ GLC(3) \\ GLC(4) \\ GLC(5) \\ GLC(6) \end{bmatrix} \qquad PYR = \begin{bmatrix} PYR(1) \\ PYR(2) \\ PYR(3) \end{bmatrix}$$

For example, glucose whose C1 is 88% 13C- enriched and C2 is 30% enriched can be presented by:

$$GLC = \begin{bmatrix} 0.88 \\ 0.3 \\ 0.0 \\ 0.0 \\ 0.0 \\ 0.0 \end{bmatrix}$$

Atom Mapping Matrix describes the transfer of atoms from reactants to products. They are constructed such that multiplication of the reactant specific activity vector by the AMM specifies the contribution to the product specific activity vector. The number of carbons in the reactant is the number of columns, while the number of carbons in the product decides the number of rows. The element in the ith row and the jth column (A_{ji}) specifies the amount of the ith carbon of the product that is derived from the jth carbon of the reactant. If it is 1, means all the ith carbon in the product are derived from the jth carbon. Typically, these numbers are 1 or 0. However, fractional elements are possible. For example, due to the symmetric structure, the mapping matrix of fumarate → malate is

$$[FUM->MAL]=\begin{bmatrix} 0.5 & 0 & 0 & 0.5 \\ 0 & 0.5 & 0.5 & 0 \\ 0 & 0.5 & 0.5 & 0 \\ 0.5 & 0 & 0 & 0.5 \end{bmatrix}$$

It is easy to see, $\sum_i A_{ij} = 1 \quad \sum_j A_{ij} = 1$, for all i, j.

The biochemistry of carbon switch could be easily represented by in string switching. For example, the pattern of carbon switching in reaction (Xylulose-5-P + R5P → F6P + E4P) could be expressed in: ABCDE + abcde → abcCDE + ABde. A c++ function is developed for producing AMMs that takes the four string patterns as the only input.

Table B-1 code to produce AMM based on pattern strings

```
/** Produce AMM based pattern strings
 * @param subAtoms A string specifies the pattern of substrate. i.e. ABCDE
 * @param proAtoms A string specifies the pattern of product, i.e. ABcdef
 * @return A string describes AMM ( MatLab syntax). Ie. “AMM = [ 1 0 \n 0 1;]”
 * @author Tao Zhu
*/

CString ProduceAMM (CString subAtoms, CString proAtoms)
{
        if (subAtom=="" || proAtom=="") {
                //There is no carbon switching between them at all.
                return "AMM = [ ]";
        }

        //build AMMs. len1 is column number, len2 is row number
        int len1 = subAtoms.GetLength();
        int len2 = proAtoms.GetLength();

        int* AMM = new int [len1 * len2];

        for (int I = 0; i < len1 * len2;  i++)
                AMM[i]=0;

        for (i=0 ; i < len2; i++){
                // AMM(i, j) specify the amount of the ith carbon of product that is developed
                // from the jth carbon of the reatant. It is either 1 or 0.
                char carbonInPro = proAtom.GetAt(i); // get a symbol, such as 'B', 'c'
                int carbonInSub = subAtom.Find(carbonInPro);
                if (carbonInSub != -1) //find it at the carbonInSub position, which means they are same carbon
                        AMM[i*len1 + carbonInSub] = 1;
        }

       CString t,strAmm;
        strAmm = "AMM = [";

        for (i=0;i<len2;i++){
                for (int j=0;j<len1;j++){
                        t.Format("%d ",AMM[i*len1+j]);
                        strAmm += t;
                }
                if (i!=len2-1)
                        strAmm +="\r\n";
        }
        strAmm +="];\r\n";

        delete [] AMM;
        return strAmm;
}
```

Appendix C Isotopomer Mapping Matrices Method (Schmidt et al. 1997, 1998)

Isotopomer Distribution Vectors (IDVs) of n-carbon metabolite has 2^n element, which represents the mole fraction of the metabolite's individual isotopomer. Since a carbon can only be labeled or non-labeled (^{12}C or ^{13}C), labeling pattern of metabolites can be coded as binary code.

As an example, the IDV of glucose has 2^6 or 64 elements. The first element of this vector is indexed by 000000_2, or 0 in octal code. The pattern 000011, which means the 5th and 6th are labeled, is the fourth elements in the IDV. The complete glucose IDV is:

$$Iglc = \begin{bmatrix} Iglc(000000_{bin}) \\ Iglc(000001_{bin}) \\ Iglc(000010_{bin}) \\ \vdots \\ Iglc(111110_{bin}) \\ Iglc(111111_{bin}) \end{bmatrix} = \begin{bmatrix} Iglc(0) \\ Iglc(1) \\ Iglc(2) \\ \vdots \\ Iglc(62) \\ Iglc(63) \end{bmatrix} \qquad (1)$$

The summation of all components equals to 1.

Isotopomer Mapping Matrices for a reaction is constructed based on the stoichiometric coefficient and information about reaction biochemistry. For each pair of substrate and product (which has carbon) in the reaction, there is isotopomer mapping matrix.

$$IMM_{MMsub->product} * IDV_{sub} = IDV_{product}$$

The individual row is associated with the product isotopomer labeling pattern, and the number of rows equals the number of elements in the product IDV. The i th row's multiplying to the IDV_{sub} gives the i th component in the $IDV_{prodcut}$, which means given a distribution of isotopomer in substrate, if the reaction's biochemistry is known, the distribution of each isotopomer in product can be solved by matrix multiplication. IMMs are matrices filled with 1 or 0 usually. The position of the nonzero elements can be found by considering all possible combinations of the reactant isotopomers, and by calculating the resulting labeling pattern of the product molecule using the AMM concept.

As an example, consider the construction of IMM for isotopomer transitions from pyruvate to oxaloacetate in pyruvate carboxylation ($IMM_{pyr \rightarrow oaa}$). The reaction is

Pyruvate + CO2 = OAA

Assume PYR is not labeled. It has pattern of [0 0 0], and CO2 is labeled, which has pattern of [1], since in the reaction, the carbon in CO2 will add into pyruvate in the following way.

$$\begin{array}{c} \mathrm{o} \\ \mathrm{o} \\ \mathrm{o} \end{array} + \bullet = \begin{array}{c} \mathrm{o} \\ \mathrm{o} \\ \mathrm{o} \\ \bullet \end{array}$$

or in the form matrix:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} [1] = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \qquad (2)$$

where $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$ and $\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$ are AMM $_{\mathrm{PYA}\rightarrow\mathrm{OAA}}$ and AMM $_{\mathrm{CO}_2\rightarrow\mathrm{OAA}}$,

In the same way, consider all the combination of any pattern of substrate, For example, if CO2 is no labeled too, the equation is

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} [0] = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \qquad (3)$$

The pattern of result OAA is [0 0 0 0]', which is first element in the $\mathrm{IDV}_{\mathrm{OAA}}$. EQ 2 and EQ 3 tell us that for a non labeled pyruvate ($\mathrm{IDV}_{\mathrm{pyr}}$ = [1 0 0 0 0 0 0 0]), the product OAA may have pattern [0 0 0 1] (last carbon is labeled) or [0 0 0 0] (none of the carbon is labeled). Hence, the first and second column of IMM $\mathrm{IMM}_{\mathrm{PYK}\rightarrow\mathrm{OAA}}$ should be

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ & & & & \mathrm{M} & & & \end{bmatrix} \qquad (4)$$

If all combination of isotopomer patters of pyruvate and CO2 is considered in the same manner, the $\mathrm{IMM}_{\mathrm{PYK} \rightarrow \mathrm{OAA}}$ is found to be:

$$IMM_{PYK \rightarrow OAA} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Appendix D Algorithm to Construct IMM from AMM

The above procedure is tedious. Based this idea, the algorithm is easy to be implement by MatLab language. For the following reaction:

$$A + B \rightarrow C + D$$

Assuming A has n carbons and B has m carbons, while C has p carbons and D has q carbons. The dimension of labeling patter of A, B, C and D are n, m, q and p respectively and the IDV for these four reactants are 2^n, 2^m, 2^p and 2^q.

The following procedure is used to find IMM from A to C.

There are 2^n columns in $IMM_{A \rightarrow C}$. We will find IMM column by column. Iterate pattern A from $[0\ 0\ ..0]_n$ until $[1\ 1\ ..1]_n$. Each pattern corresponds to a column in the IMM.

For the first pattern of A, which is $[0\ ..0]_n$, iterate pattern B from $[0\ 0\ ..0]_m$ until $[1\ 1\ ...1]_m$ to find all possible pattern C that corresponds to this pattern A. Then convert all pattern C to octocal code (x_1, x_2, … x_i..). The first column is found by set the all x_i elements as 1 and left element 0.

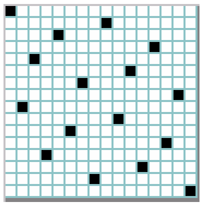
By the same procedure, all columns in IMM can be produced. This algorithm can be finished in O (n^2) time.

Usually, the element in the IMM is 1 or 0. But value of 0.5 may appear if we consider the scrambling effect of symmetrical molecules. As an example, consider the reaction between succinate-CoA and succinate. Because succinate is symmetric, a succinate-CoA labeled as (1 0 0 0) could be converted to a succinate molecule labeled as (1 0 0 0) or (0 0 01). If succinate was not symmetricl, the IMM (succ-CoA→ succ) is simply unit matrix (*I*). And if succinate was not symmetrical and all atoms sequenced oppositely after reaction, the IMM is found using the described algorithm:

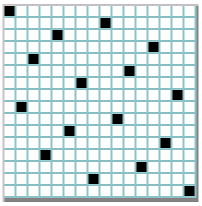
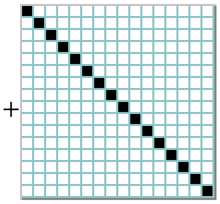
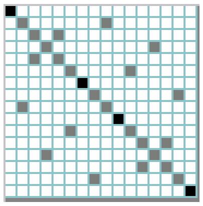
$$IMM= [\begin{matrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0
\end{matrix}$$

0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0
0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1]

To save the space, I would like use following bitmap to represent this matrix.

IMM= 

Since the real isotopomer flow is the means of these two fluxes, the IMM taking account the scrambling effect would be:

IMM=( + )/2 = 

In above bitmaps, black grid represents 1, gray one represents ½, and blank grid represents 0.

A MatLab® code (Table D-1) developed by Karsten Schmidt is adopted to create the IMM based on the AMM of the reaction: A + B → C + D. The AMM is automatically produced using the procedure described in appendix C.

Table D-1 genimm.m (Karsten Schmidt)

```
function [imm_AxC, imm_AxD, imm_BxC, imm_BxD] = genimm(...
         amm_AxC, amm_AxD, amm_BxC, amm_BxD)

% Author: Karsten Schmidt
% date:   20.11.95
%
% This Script generates the isotopomer mapping matrices
% of a reaction  A + B -> C + D on the basis of the respective
% atom mapping matrices.
%
% In case of only one substrate or product, the respective input
% AMMs can be empty matrices ([] or '') but must be present.

clear imm_AxC imm_AxD imm_BxC imm_BxD


% check input arguments

if (isempty(amm_AxC))
  error('first argument is an empty matix')
end

% get the number of carbon atoms in A, B, C and D

n_A = size(amm_AxC, 2);   % number of columns in matrix
n_B = size(amm_BxC, 2);   % number of columns in matrix
n_C = size(amm_AxC, 1);   % number of rows in matrix
n_D = size(amm_AxD, 1);   % number of rows in matrix

% fill matrices with zeros

imm_AxC = zeros(2^n_C, 2^n_A);
imm_AxD = zeros(2^n_D, 2^n_A);
imm_BxC = zeros(2^n_C, 2^n_B);
imm_AxD = zeros(2^n_D, 2^n_A);

for i = 1 : 2^n_A
  for j = 1 : 2^n_B
    p_A = bin(i-1, n_A);                 % labeling pattern of A
    p_B = bin(j-1, n_B);                 % labeling pattern of B

    if (n_B == 0)                 % no B in the reaction
      p_C=(amm_AxC*p_A')';               % labeling pattern of C
      p_D=(amm_AxD*p_A')';               %  labeling pattern of D
    else
       p_C=(amm_AxC*p_A'+amm_BxC*p_B')';      % labeling pattern of C
       if(n_D ~= 0)  % added by Tao Zhu
          p_D=(amm_AxD*p_A'+amm_BxD*p_B')';   % labeling pattern of D
       end %if
    end

    imm_AxC(dec(p_C)+1, i) = 1;          % imm from A to C
    imm_BxC(dec(p_C)+1, j) = 1;          % imm from B to C

    if (n_D ~= 0)
      imm_AxD(dec(p_D)+1, i) = 1;        % imm from A to D
      imm_BxD(dec(p_D)+1, j) = 1;        % imm from B to D
    end % if
  end % for j
end % for i
```

Table D-1 (continued)

```
if (n_B == 0)            % if there is no B then these IMMs
  imm_BxC = [];          % do not exist!
  imm_BxD = [];
end

if (n_D == 0)            % if there is no D then these IMMs
  imm_AxD = [];          % do not exist!
  imm_BxD = [];
end
```

BIBLIOGRAPHY

BIBLIOGRAPHY

1. Bailey J.E. 1998. Mathematical Modeling and Analysis in Biochemical Engineering: Past Accomplishments and Future Opportunities. Biotechnol. Prog. **14**:8-20.

2. Bonarius H. P.J., Schmid G., Tramper J., 1997. Flux Analysis of Underdetermined Metabolic Networks: The Quest For The Missing Constraints. Trends in Biotechnol. **15**:308-314.

3. Burgard AP, Maranas CD, 2001a. Probing the Performance Limits of the Escherichia coli Metabolic Network Subject to Gene Additions or Deletions. Biotechnol Bioeng 74:364-375

4. Burgard AP, Vaidyaraman S,Maranas CD, 2001b. Minimal Reaction Sets for Escherichia coli Metabolism under Different Growth Requirements and Uptake Environments. Biotechnol. Prog. 17, 791-797

5. Chou C.H, Bennett G.N., and San K.Y. 1994. Effect of modifying glucose uptake using genetic engineering techniques on high level recombinant protein production in *Escherichia coli* dense culture. *Biotechnol. Bioeng.* 44, 952-969.

6. Dauner M, Sauer U. 2001. Stoichiometric Growth Model for Riboflavin-Producing Bacillus subtilis. Biotechnol Bioeng 76,132–143.

7. Dhedia N.N., Hottiger T., and Bailey J.E. 1994. Overproduction of glycogen in *E. coli* blocked in the acetate pathway improves cell growth. *Biotechnol. Bioeng.* 44, 132-139.

8. Diaz-Ricci J.C., Regan L., and Bailey J.E. 1991. Effect of alteration of the acetic acid synthesis pathways on the fermentation of *Escherichia coli. Biotechnol. Bioeng*. 38, 1318-1324.

9. Edwards JS, Palsson PO, 2000. Robustness Analysis of the Escherichia coli Metabolic Network. Biotechnol. Prog. 16, 927-939

10. Farmer W.R. and Liao J.C. 1997. Reduction of aerobic acetate production by *Escherichia coli*. *Applied & Environ. Micro*. 63, 3205-10.

11. Farmer W.R. and Liao J.C. 2001. Precursor Balancing for Metabolic Engineering of Lycopene Production in *Escherichia coli.* Biotechnol. Prog, 17:57-61.

12. Fell, D. A., & Small, J.R. 1986. Fat synthesis in adipose tissue: an examination of stoichiometric constraints Biochemical Journal, 238, 781-786.

13. Fisher S.H. and Magasanik B. 1984. Synthesis of oxaloacetate in *Bacillus subtilis* mutants lacking the $\alpha-$ketoglutarate dehydrogenase enzyme complex. *J. Bacteriol*. 158: 55-62.

14. Flores, S., Gosset, G., Flores, N., Graaf, A. A. and Bolívar F. 2002. Analysis of Carbon Metabolism in Escherichia coli Strains with an Inactive Phosphotransferase System by 13C Labeling and NMR Spectroscopy. Metabolic Engineering 4:124–137.

15. Fry B., Zhu T., Phalakornkule C., Koepsel R., Domach M. M., Ataai M. M.2000. Characterization of Cell Growth, Acid Production In Pyruvate Kinase Mutant of *B. subtilis*. Appl. Environ. Microbiol. **66**:4045-4049.

16. E. Gawrilow, M. Joswig, G.M. Ziegler, et al.: POLYMAKE --- a framework for analyzing convex polytopes, TU Berlin 1996

17. Glaser R. 1999. FCAL, Ver.2.3.0. Zurich: ETH Zurich.

18. Goel A., Domach M.M. and Ataai M.M. 1999. Metabolite fluxes, pools and enzyme measurements suggest a tighter coupling of the energetic and biosynthetic reactions requires reduction in pyruvate kinase flux. *Biotechnol Bioeng*. 64, 129-134.

19. Goel A., Ferrance J., and Ataai M. M.. 1993. Analysis of metabolic fluxes in batch and continuous cultures of *Bacillus Subtilis*. *Biotechnol. Bioeng.* 42, 686-696.

20. Goel A., Lee J., Domach M.M., and Ataai M.M. 1996. Coordination of glycolysis and TCA cycle reaction networks: citrate-glucose metabolism eliminates acids and reveals potential metabolic engineering strategies. *Annals. N.Y. Acad. Sci.* 782, 1-16.

21. Goel A., Lee J., Domach M.M., and Ataai, M.M. 1995. Suppressed acid formation by cofeeding glucose and citrate in *Bacillus* Cultures: emergence of pyruvate kinase as a potential metabolic engineering site. *Biotechnol. Prog.* 11, 380-386.

22. Goryanin I., Hodgman C.T., Selkov E. 1999. Mathematical Simulation and Analysis of Cellular Metabolism and Regulation. Bioinformatics **15**:749–758.

23. Hartley, R. 1985, Linear and Nonlinear Programming: An Introduction to Linear Methods in Mathematical Programming, Ellis Horwood.

24. Houck, C., Joines, J and Mike Kay. 1995. A Genetic Algorithm for Function Optimization: A Matlab Implementation. NCSU-IE TR 95-09.

25. Ingraham JL, Maaloe O, Neidhardt FC. 1983. ``Growth of the Bacterial Cell,'' Sinauer Associates, Sunderland, MA.

26. Jensen K.F. and Pederson S. 1990. Metabolic growth rate control in *Escherchia coli* may be a consequence of subsaturation of the macromolecular apparatus with substrates and catalytic components. *Microbial. Rev.* 54, 89-100.

27. Klapa, M. I., Park, S. M., Sinskey, A. J., and Stephanopoulos, G. 1999. Metabolite and isotopomer balancing in the analysis of metabolic cycles: I. Theory. Biotech. Bioeng. 62 : 375-391.

28. Ko, Y.-F., W. E. Bentley, and W. A. Weigand. 1993. An integrated modeling approach to describe the energy efficiency of E. coli fermentation under oxygen-limited conditions; cellular energetics, carbon flux, and acetate production. Biotechnol. Bioeng. 42:843–853.

29. Lee J., Goel A., Ataai M.M. and Domach M.M. 1997. Supply-side analysis of growth of *B. subtilis* on glucose-citrate medium: feasible network alternatives and yield optimality. *Applied and Env. Microbiol.* 63, 710-718.

30. Lee S, Phalakornkule C, Ataai MM, Domach MM, Grossmann IE. 2000. Recursive MILP Model for Finding All the Alternate Optima in LP Models for Metabolic Networks. Comput. Chem. Eng. 24:711-716.

31. Liao J.C., Oh M.K. 1999. Toward Predicting Metabolic Fluxes in Metabolically Engineered Strains. Metab. Eng. **1**: 214-223.

32. Majewski RA, Domach MM. 1990. Simple constrained optimization view of acetate overflow in E. coli. Biotechnol. Prog. 35:732-738.

33. Mandelstam J, McQuillen K, and Dawes I. 1982. ``Biochemistry of Bacterial Growth," Halsted Press, New York.

34. Maniatias, T., Fritsch, E. F., Sambrook, J. 1982. Molecular cloning: a laboratory manual; Cold Spring Harbor Press: New York.

35. Marr A.G. 1991. Growth Rate of *Escherchia coli. Microbial. Rev.* 55, 316-333.

36. Matheiss TH, Rubin DS, 1980. A survey and comparison of methods for finding all vertices of convex polyhedral sets. Mathematics of Operations Research. 5:167-185.

37. Mendes, P. 1997. Biochemistry by Numbers: Simulation of Biochemical Pathways with Gepasi 3. Trends Biochem. Sci. **22**: 361–363.

38. Neidhardt, F. C., Ingraham, J. L., Schaechter, M. 1990. Physiology of the bacterial cell: A molecular approach. Sinauer Associates, Inc., Sunderland, MA.

39. Pfeiffer T., Sánchez-Valdenebro I., Nuno J.C., Montero F., Schuster S. 1999. METATOOL: for Studying Metabolic Networks. Bioinformatics. **15**:251-257.

40. Phalakornkule C., Fry B., Zhu T., Koepsel R., Ataai M. M., Domach M. M., 2000. ^{13}C NMR Evidence for Pyruvate Kinase Flux Attenuation Underlying Suppressed Acid Formation in *B. subtilis* Cultures. Biotechnol. Prog. **16**:169-175.

41. Phalakornkule C., Lee, S., Zhu, T., Koepsel, R., Ataai, M. M., Grossmann, I.E. and Domach. M. M. 2001. A MILP-Based Flux Alternative Generation and NMR Experimental Design Strategy for Metabolic Engineering. *Metabolic Engineering*, 3,124-137.

42. Ponce E., Flores N., Martinez A., Bolivar F., and Valle F. 1998. Simulation of glucose catabolism through the pentose pathway by the absence of the two pyruvate kinase isoenzymes in *Escherichia coli*, Biotech. Bioeng. 58, 292-295.

43. Ponce E., Flores N., Martinez A., Valle F., and Bolivar F. 1995. Cloning of the two pyruvate kinase isoenzyme structural genes from *Escherichia coli*: the relative roles of these enzymes in pyruvate biosynthesis. *J. Bact*. 177, 5719-22.

44. Pramanik J, Keasling JD, 1998. Effect of Escherichia coliBiomass Composition on Central Metabolic Fluxes Predicted by a Stoichiometric Model. Biotech. Bioeng. 60:230–238.

45. Sauer U, Cameron DC, Bailey JE. 1998. Metabolic capacity of Bacillus subtilis for the production of purine nucleosides, riboflavin, and folic acid. Biotechnol Bioeng 59:227–238.

46. Sauer, U., Hatzimanikatis, V., Bailey, J. E., Hochuli, M., Szyperski, T.,Wu¨thrich, K. 1997. Metabolic fluxes in riboflavin-producing Bacillus subtilis. Nature Biotechnol. 15: 448–452.

47. Schaff J., Fink C.C., Slepchenko B., Carson J.H., Loew L.M.1997. A General Computational Framework for Modeling Cellular Structure and Function. Biophys. J. **73**:1135–1146.

48. Schilling CH, Letscher D, Palsson BØ. 2000a. Theory for the systemic definition of metabolic pathways and their use in interpreting metabolic function from a pathway-oriented perspective. J Theor Biol 203:229–248.

49. Schilling CH, Palsson BØ. 2000b. Assessment of the metabolic capabilities of Haemophilus influenzae Rd through a genome-scale pathway analysis. J Theor Biol 203:249–283.

50. Schmidt K., Carlsen M., Nielsen J. and Villadsen J. 1997. Modeling Isotopomer Distributions in Biochemical Networks Using Isotopomer Mapping Matrices. Biotech. Bioeng. 55: 831 – 840.

51. Schmidt, K., Marx, A. Graaf, A.A, Wiechert, W., Sahm, H., Nielsen, J. and Villadsen, J. 1998. ^{13}C Tracer Experiments and Metabolite Balancing for Metabolic Flux Analysis: Comparing Two Approaches. Biotech. Bioeng., 58: 254-257.

52. Sedgewich R. 2001. Algorithms in C Part 5: Graph Algorithms. Addison-Wesley Pub Co.

53. Schuster S, Dandekar T. and Fell. D.A. 1999. Detection of elementary flux modes in biochemical networks: a promising tool for pathway analysis and metabolic engineering. Trends in Biotechnology. 17:53-60.

54. Shiloach, J., J. Kaufman, A. S. Guillard, and R. Fass. 1996. Effect of glucose supply strategy on acetate accumulation, growth, and recombinant protein production by

Escherichia coli BL21 (lambda DE3) and Escherichia coli JM109. Biotechnol. Bioeng. 49:421–428.

55. Stephanopoulos, G and Vallino, J.J. 1991. Network rigidity and metabolic engineering in metabolite overproduction. Science **252**:1675-1681.

56. Stephanopoulos, A. Aristidou and J. Nielsen. 1997. Metabolic Engineering: Principles and Methodologies, Academic Press.

57. Tempest, D. W., Meers, J. L., and Brown, C. M. 1970. Influence of environment on the content and composition of microbial free amino acid pools. J. Gen. Microbiol. 64, 171-185.

58. Tomita M. 2001. Whole-Cell Simulation: A Grand Challenge of the 21st Century. TRENDS in Biotechnology. **19**: 205-210.

59. Varma A, Palsson BØ. 1994a. Metabolic flux balancing: basic concepts, scientific and practical use. Bio/Technology 12: 994–998.

60. Varma A, Palsson, B.O. 1994b. Stoichiometric flux balance models quantitatively predict growth and metabolic by-product secretion in wild type Escherichia coli W3110. Applied & Environmental Microbiology, 60: 3724-3731.

61. Wiechert W., Graaf, A.A. 1997. Bidirectional Reaction Steps in Metabolic Networks: I. Modeling and Simulation of Carbon Isotope Labeling Experiments. Biotech. Bioeng., 55: 101 – 117.

62. Zhu T, Domach MM, Ataai MM. Software development of convex analysis of stoichiometric model and NMR simulation. Metabolic Engineering. Submitted.

63. Zhu T., Koepsel R.R., Grossmann I.E., Ataai M.M., Domach M.M. Theoretical Improvement of Folic Acid Production in *B. subtilis* and *E. coli*: Strain Design Using Convex Analysis. Manuscript in Preparation.

64. Zhu T., Phalakornkule C., Koepsel R. R., Domach M. M., Ataai M. M., 2001. Cell Growth and By-Product Formation in a Pyruvate Kinase Mutant of *E. coli.* Biotechnol. Prog. **17**: 624-628.

65. Zupke, C. and Stephanopoulos, G. 1994. Intracellular Flux Analysis in Hybridomas Using Mass Balance and in Vitro ^{13}C NMR. Biotech. Bioeng. 45: 292-303.

66. Zupke, C., Tompkins R.,Yarmush, D.and Yarmush, M. 1997. Numerical Isotopomer Analysis: Estimation of Metabolic Activity. Analytical Biochemistry. 247:287-293