MULTI-AGENT BASED MODELING AND SIMULATION OF METABOLIC NETWORKS

MOHAMMAD IFTEKHAR HOSSAIN

NATIONAL UNIVERSITY OF SINGAPORE

2008

MULTI-AGENT BASED MODELING AND SIMULATION OF METABOLIC NETWORKS

MOHAMMAD IFTEKHAR HOSSAIN
(B.Sc in Chemical Engineering, BUET, Bangladesh)

A THESIS SUBMITTED
FOR THE DEGREE OF MASTER OF ENGINEERING
DEPARTMENT OF CHEMICAL AND BIOMOLECULAR ENGINEERING
NATIONAL UNIVERSITY OF SINGAPORE
2008

Acknowledgements

This work is the most significant scientific accomplishment in my career so far and it would be impossible without the people who believed in me and supported me from their respective position. I would like to take this opportunity and thank them here.

First, I would like to express my deepest gratitude towards my supervisor, A/P Raj. Srinivasan for his excellent guidance and continued support throughout this work. His resourceful thoughts and invaluable ideas help me to explore new areas during the course of the research. In the same time I would like to thank my co-supervisor Dr. Lee Dong Yup for his valuable suggestion with his excellent scientific background during the course of research.

I am very grateful to A/P M.S. Uddin for providing mental support and fatherly guidance during the course of my study and stay in Singapore.

I would like to thank all my lab mates, Jonnalagadda Sudhakar, Ng Yew Seng, Kaushik Ghosh and Ang Bee Lee for maintaining a healthy, enjoyable and pleasant working environment.

I wish to thank my all friends for their help, support and love. They include, M M Faruque Hasan, Arief Adhitya, Rajib Saha, Shudipto Konika Dishari, Manish Mishra, Mohammad Moydul Islam, Shubhra Joyti Bhadra, etc.

I would like to express my deep gratitude and love for my parents, my brother, my sister and brother-in-law, who wholeheartedly supported me in my work with their blessing and love.

Finally, I offer my utmost gratitude to Almighty Allah, from whom all blessings flow.

Table of Contents

Summary

The cardinal role of metabolic engineering in the field of biotechnology is increasing day-by-day, as biotechnology has become a vital tool for almost every industry, including chemical, pharmaceutical, health care, and food industries. Effective genetic manipulation of cell metabolism for performance enhancement is a critical step in obtaining low cost and high yield production. Increasingly, mathematical models play an important role in this field; examples include computational tools for simulation, data evaluation, design of experiments, systems analysis, prediction, design, and optimization. The first step in developing a comprehensive metabolic model of a microorganism is to identify all the metabolic pathways for the organism from available databases (such as KEGG). Often, the databases are incomplete which leads to incorrect results when the resulting model is simulated. In this work, we present an agent-based modeling and simulation (ABMS) approach to analyze metabolic pathways for inconsistencies. In the proposed approach, the metabolic system is modeled using three types of agents: Reaction agent, Cytoplasm agent, and Scheduler agent. Each metabolic reaction in the system is represented by a Reaction agent. The Cytoplasm agent resembles the cellular environment and the Scheduler agent regulates the execution of reactions. Starting from the substrate (or minimal nutrient condition), reactions are qualitatively executed by the Scheduler in a sequential manner. The reachability of the final product indicates the completeness of the pathway. In case of an incomplete network, the minimal set of reactions necessary to reach the final pathway can also be identified by this approach. The proposed approach thus identifies gaps in the network through qualitative simulation and would hence serve as a precursor to numerical modeling & simulation.

We illustrate the approach using a metabolic model of *E. coli*, that includes Glycolysis, Pentose-Phosphate pathway, TCA cycle, Anaplerotic reactions, Pyruvate metabolism, Respiration and transport system reactions. We have also extended the same agent-based framework to perform dynamic simulation when kinetics of metabolic reactions are available. Simulation results are presented to illustrate the proposed modeling and simulation approach and its effectiveness is evaluated through comparison with published literature.

List of Figures

List of Tables

Nomenclature

Enzymes

ALDO /aldo	Aldolase/ Fructose bisphosphate aldolase class I, II
DAHPS	DAHP Synthases
eda	2-keto-3-deoxy-6-phosphogluconate aldolase
edd	phosphogluconate dehydratase
ENO/eno	Enolase
fbp	fructose 1-6 bisphosphatase
G1PAT	Glucose-1-phosphate adenyltransferase
G3PDH	Glycerol-3-phosphate dehydrogenase
G6PDH/zwf	Glucose-6-phosphate dehydrogenase
GAPDH/gapA	Glyceraldehyde 3-phosphate dehydrogenase
glk	Glucokinase
gnd	6 phosphogluconate dehydrogenase
MetSynth	Methionine synthesis
MurSynth	Mureine synthesis
PDH	Pyruvate dehydrogenase
PEPCxylase	PEP Carboxylase
PFK/pfkAB	phosphatefructokinase
PGDH	6-phosphogluconate
PGI/pgi	Glucose-6-phosphate isomerase/ phosphoglucose isomerase
PGK/pgk	Phosphoglycerate kinase
pgl	phosphogluconolactonase

PGLM	Phosphoglucomutase
PGM/pgml	Phosphoglycerate mutase
PK/pykFA	Pyruvate kinase
PTS	Phosphotransferase system
R5PI/rpiA	Ribose-phosphate isomerase/Ribose 5-phosphate isomerase
RPPK	Ribose-phosohate pyrophosphokinase
Ru5P/rpe	Ribose-phosphate epimerase
Synth1	synthesis 1
Synth2	synthesis 2
TA	Transaldolase
TIS/tpiA	Triosephosphate isomerase
TKa / tktA	Transketolase reaction a
TKb/tktB	Transketolase reaction b
tktAB	Transketolase reaction a,b
TrpSynth	Tryptophan synthesis

Metabolites

2pg/2PG	2-phosphoglycerate
3pg/3PG	3-phosphoglycerate
6pg/D6PGL	6-phosphogluconate
accoA/ACCOA	Acetyl-coenzyme A
adp/ADP	Adenosindiphosphate
amp/AMP	Adenosinmonophosphate
atp/ATP	Adenosintriphosphate

CO_2	Carbondioxide
coA/COA	Coenzyme A
dhap/T3P2	dihydroxyacetonephosphate
e4p/E4P	erythrose-4-phosphate
f6p/F6P	Fructose-6-phosphate
fdp/FDP	Fructose-1,6-biphosphate
g1p	Glucose -1-phosphate
g6p/G6P	Glucose-6-phosphate
gap/T3P1	glyceraldehyde-3-phosphate
glc/GLC	Glucose
glc_{ext}/GLC_{ext}	extracellular glucose
met	methionine
nad/NAD	diphosphopyridinuclotide
nadh/NADH	reduced diphosphopyridinuclotide
nadp/NADP	diphosphopyridinuclotide_phosphate
nadph/NADPH	diphosphopyridinuclotide
p/PI	inorganic phosphate
pep /PEP	Phosphoenolpyruvate
pgp /13DPG	1,3- biphospho glycerate
pyr /PYR	Pyruvate
Rib5p/R5P	ribose 5 phosphate
Ribu5p/RL5P	diphosphopyridinuclotide
Sed7p/S7P	sedoptulose – 7 –phosphate
SUCC	succinate
SUCCOA	succinyl co enzyme A

Trp	tryptophan
Xyl5P/X5P	xylulose -5-phosphate

Pathways

PPP	Pentose Phosphate pathway
TCA cycle	Tricarboxylic acid cycle

Subscript

i	index used for representing different metabolites
j	index used for representing various reactions
ext	index used to represent extracellular metabolites

Variable

C	Concentration of metabolite
r	Rate of reactions
$\gamma_{\max}$	Maximum allowable conversion
ξ	Extent of reaction

Chapter 1 Introduction

1.1 Introduction to Metabolic Engineering

Metabolic engineering mainly deals with the analysis and modification of metabolic pathways. This field emerged during the past decade as a result of the developments in a number of different technologies. Gradually, it is becoming the center of research endeavors in biological and biochemical engineering, cellular physiology, applied microbiology as well as in bioprocess and biotechnology. Although the notion of pathway manipulation for the purpose of endowing microorganisms with desirable properties is old, the perception of metabolic engineering as defining a discipline was first put forward by Bailey in 1991. Right after that, this new field was nurtured by the life science and engineering communities. Both these fields have found that this emerging field provides the opportunity to capture the potential sequences and other information generated from genomic research and usher a novel path for biological researches.

The focal point of the current practice of metabolic engineering is the manipulation of existing pathways or reactions producing a certain metabolite or macromolecule, and the introduction of new pathways or reactions into host cells. These activities can be classified into five major groups (Cameron and Tong, 1993; Stephanopoulos et al., 1998; Lee and Papoutsakis, 1999): (i) enhanced production of metabolites and other biologicals already produced by the host organism; (ii) production of modified or new metabolites and other biologicals that are new to the host organism; (iii) extending the substrate utilization range for cell growth and product formation; (iv) designing improved or new metabolic pathways for degradation of various chemicals, including xenobiotics; (v) modification of cell properties that facilitate bioprocessing such as enhance fermentation product and product recovery.

As an outcome of extensive research endeavors from the life science and engineering communities along with electrifying improvements in genomic research, increase in the number of sequenced genomes, and the profound advancements in experimental high-throughput analyses, increasing attention has been devoted to developing effective computational methods in biology. These methods comprise the development of comparative tools and maintenance of databases for the analysis of genomics data in the domain of bioinformatics, as well as the construction of models for the analysis and integration of the data in terms of the system properties. In the domain of metabolic analysis an important asset to such analyses is the reconstruction (partial or full genome scale) of cellular networks that includes the collection and visualization of all physiologically relevant cellular processes.

1.2 Modeling and Simulation in Metabolic Engineering

Mathematical modeling is one of the key methodologies of metabolic engineering. In order to reach the ambitious goal of development of targeted methods to improve the metabolic capabilities of industrially relevant microorganisms, tools that assist in the evolutionary process of genetic manipulations of the cell metabolism and the improvement of bioprocess conditions are required. From an engineering perspective, mathematical modeling is one of the most successful scientific tools available for this task. Based on a given metabolic model, different computational tools for the simulation, data evaluation, systems analysis, prediction, design and optimization of metabolic systems can be developed. From the analytical point of view, the application of such kind of modeling and simulation tools is substantially important.

The focus of modeling in cell physiology has always been on the understanding of metabolic systems in the sense of the general principles that govern the cellular function. The new aspect of modeling in metabolic engineering is the usage of models for the targeted direction of metabolic fluxes in the sense of a rational engineering design. Following are some potential activities in metabolic engineering where modeling and simulation can contribute significantly:

- *Understanding the system*: Mathematical models are the quantitative representation of knowledge with the ability to have a unique and objective interpretation. A model is mainly based on the understanding of the basic principles of the system. Comparing the model output with the real system output might be helpful explore additional understanding of the system. Based on a given model, mathematical methods can help obtain a better understanding of the system's structure and its qualitative behavior. For example, Goldbeter (1996) has modeled the biochemical rhythms and oscillation at cellular level and that model threw light on the mechanism of periodic behaviour at the molecular and cellular levels and explained how enzyme regulation or receptor desensitization can give rise to oscillations.
- *System analysis*: Mathematical model is a very easy but effective analytical tool for metabolic system. Model could be used to identify the functional units in a metabolic system, for the computation of steady state, for the determination of parameter sensitivity (Albe and Wright, 1992), for

investigation of dynamic behaviors, for computing theoretical limits of the systems metabolic capabilities (Edwards and Palsson, 1998), etc.

- *Interpretation and evaluation of data:* By analyzing mathematical model, one can achieve a better interpretation of the measured data. Reproduction of experimental data by mathematical model can provide a fair appreciation of the measured data. For example, the characterization of growth, nutrient uptake and product formation by macrokinetic models has become a standard procedure in bioprocess development (Takors et al., 1997).
- *Simulation:* Undoubtedly, the most frequent application of models is the exploration of the possible behavior of a system. Simulation scenarios based on rather crude mathematical models can help to achieve a rough understanding of the system behavior and to reject false hypotheses. Many conceptual studies based on more or less simple models belong to this category. Several interesting examples are presented by Heinrich and Schuster (1996).
- *Design and Prediction:* The outcome of future experiments can be predicted using a validated mathematical model and the ultimate goal of such tool is to provide a means to a rational design process for metabolic pathways.
- *Optimization:* Once a valid model with impressive predictive power is available, it can be used to handle the problem of optimal metabolic design.

However, the application of a model is always limited to a certain type of problem. For example, a stoichiometric network model is suitable for metabolic flux analysis but it contains no information about regulatory mechanisms. Thus, it has little predictive power with respect to pathway alterations. Likewise, model validation for regulatory models is usually done with measured data from a few physiological states (e.g. exponential growth in a batch culture).

1.3 Developing Network Model from Genome Sequence

Prior to analyzing cellular metabolism, the first step is to develop a network model from the genomic databases. This development is not straightforward, it may take more than a year to fully delineate a genome-scale model for an organism through a iterative process of network characterization and re-annotation. Though the main interest of this thesis relates to the analysis of metabolic networks, the basic practice of developing any other biological networks like protein-interaction, signaling or regulatory network is almost the same. Before a more through discussion of network reconstruction, definitions of some important terms are established.

- *Network reconstruction*: The objective of reconstruction is to provide a detailed description of network components and their interactions.
- *Genome annotation*: Genome annotation refers to characterization of an organism that includes information regarding function of cellular components, their interactions, spatial organization and evolutions, etc.
- *One dimensional genome annotation*: It involves the identification of genes in the genome and also assigning known or expected functionality to the identified gene product.
- *Two dimensional genome annotation*: It specifies physical and chemical interactions between cellular components. The delineation of characterized cellular component basically leads to the reconstruction of networks.

- *Metabolic pathway*: A metabolic pathway is a series of biochemical reactions occurring within a cell. In each pathway an essential chemical is modified into other essential chemicals by chemical reactions.
- *Metabolic network*: It is a collection of metabolic pathways, characterized by a complete set of physical and chemical interactions that determine the physiological and metabolic characteristics of the cell.
- *Metabolite connectivity*: Metabolic connectivity represents the participation of a metabolite in different reactions and is equal to the number of reactions a metabolite participates in.
- *Dead-end metabolite*: Dead-end metabolite is one kind of gap in the network. For incomplete networks some metabolites might only be produced or consumed. Such metabolite is termed as dead-end metabolite.
- *Blocked reaction*: Reactions not connected to the main network are blocked reactions. Blocked reaction is usually isolated from the rest of the network.
- *Network gap*: Any inconsistencies in the network that stops production of essential components are termed as gaps.

There are various approaches towards the reconstruction of a metabolic network, which are briefly discussed in the literature review section. As a first step in reconstruction, the genes with known or predicted functionality are specified from the genome sequence database.

In the next step, cellular components, such as gene products are specified and characterized in terms of their interactions. This step is very important as it deals with the biochemical accuracy of the network model. The metabolic properties of the model depend on these interactions. A systematic guide to this step as prescribed by Reed et al (2006) is shown in Figure 1-1.

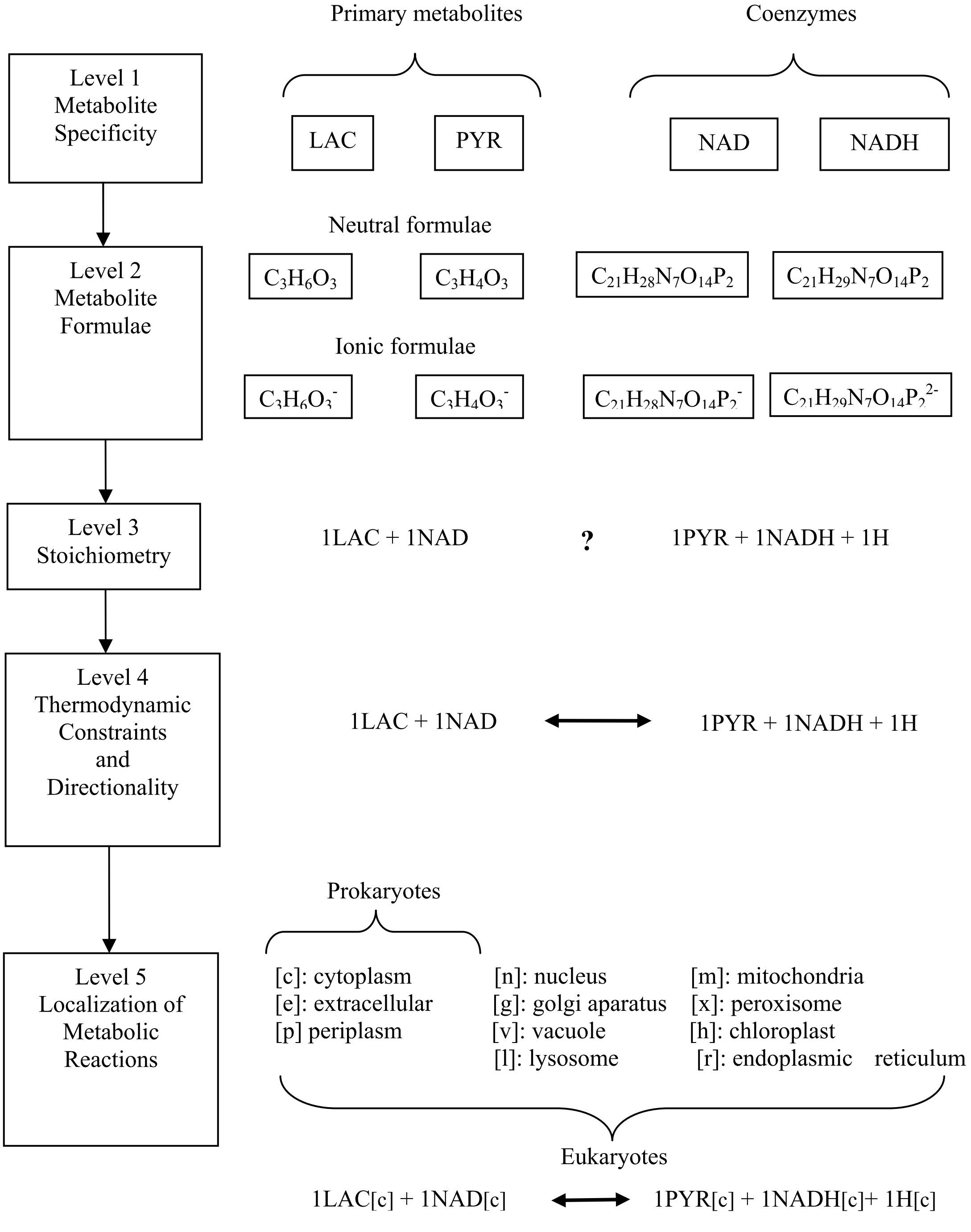

Primary metabolites
Coenzymes
Level 1 Metabolite Specificity
LAC
PYR
NAD
NADH
Level 2 Metabolite Formulae
Neutral formulae
$C_3H_6O_3$
$C_3H_4O_3$
$C_{21}H_{28}N_7O_{14}P_2$
$C_{21}H_{29}N_7O_{14}P_2$
Ionic formulae
$C_3H_6O_3^-$
$C_3H_4O_3^-$
$C_{21}H_{28}N_7O_{14}P_2^-$
$C_{21}H_{29}N_7O_{14}P_2^{2-}$
Level 3 Stoichiometry
1LAC + 1NAD ? 1PYR + 1NADH + 1H
Level 4 Thermodynamic Constraints and Directionality
1LAC + 1NAD ⟷ 1PYR + 1NADH + 1H
Level 5 Localization of Metabolic Reactions
Prokaryotes
[c]: cytoplasm
[e]: extracellular
[p] periplasm
[n]: nucleus
[g]: golgi aparatus
[v]: vacuole
[l]: lysosome
[m]: mitochondria
[x]: peroxisome
[h]: chloroplast
[r]: endoplasmic reticulum
Eukaryotes
1LAC[c] + 1NAD[c] ⟷ 1PYR[c] + 1NADH[c]+ 1H[c]

Figure 1-1: Defining biochemical interactions among metabolites

Once the metabolites' specifications have been completed, a primary network is constructed and the next step is to analyze the pathway and check for consistency with the wet-lab experimental outcome. This analysis may involve a biomass production capability or external flux measurement or measurement of intermediate metabolite concentration. If the results of the analysis differ from the experimental results, that indicates the presence of gaps in the network. So, the next two steps are to identify and fill those network gaps with the help of metabolic databases. These operations continue in an iterative fashion until consistent results with the experimental data are obtained. Then the reconstruction of metabolic network is completed and ready for further analysis.

1.4 Objective of the Thesis

The natural network of a living cell is gigantic, making the understanding of the full network difficult. It consists of many reactions and a huge number of metabolites participating in different pathways. For eukaryotes, even more complexities are added, as the cell contains a number of compartments. Furthermore, an organism is affected by environmental factors like substrate concentration and temperature. To reveal the complexity of biological networks and to interpret the huge Omics data, a clear and unambiguous representation is necessary, one that allows a step-wise composition and different description levels to build a hierarchical system. Due to the large scale of complexity, validation as well as an automatic qualitative and quantitative analysis is required.

This thesis strives to explore the potential application of agent-based modeling and simulation of cellular metabolic networks that help for static analysis to identify network gaps as well as dynamic simulation.

1.5 Thesis Overview and Organization

-Summary of Chapter 2: Prior Art

Chapter 2 provides a broad overview of the current literature, promises of metabolic engineering along with the current practices in this emerging field. It begins with Section 2.1, which defines metabolic engineering with a brief description of the evolution of this research area. This is then followed by a brief survey of metabolic network analysis in Section 2.1.1 and a detailed discussion on the potential applications of metabolic engineering approaches in Section 2.1.2. Section 2.2 throws some light on the importance of modeling of cellular system for metabolic engineering purposes, along with various modeling approaches. This justifies the needs and sets the stage for the present work, and explores the applications of a new modeling approach. Agent based modeling is described in detail in Section 2.3.

-Summary of Chapter 3: Agent Based Metabolic Network Analysis

Chapter 3 provides a detailed description of modeling cellular metabolic network using a multi agent system. It begins with the suitability of the agent based approaches in designing biological systems as it has the potential to replicate systems at its minimum individual components. Then it provides a description of how metabolic networks can be modeled as a multi agent system. In Section 3.1 the proposed agent based framework is explained by describing the structure and functionality of different interacting agents involved in the system. The next section explains the emergence of the network structure from the interaction between agents. In Section 3.3 the strategy applying the agent based model to detect gaps in metabolic networks is proposed, which is demonstrated with the help of the simple network of *E. coli*'s central metabolism in Section 3.4. Section 3.5 illustrates the method for filling gap resulting from missing reactions.

-Summary of Chapter 4: Dynamic Simulation

Chapter 4 describes the method of dynamic simulation of metabolism using agent based simulation techniques. It starts with a brief introduction of the central metabolism of *E .coli,* including the importance of dynamic analysis of metabolism. In Section 4.2 the dynamic model is described in details. The modified structure of the agent based framework along with a brief explanation of the dynamic model of individual agents is illustrated in Section 4.3. In the next section, the agent based simulation result is discussed and validated with experimental results

-Summary of Chapter 5: Summary, Conclusions and Recommendations

Chapter 5 concludes by justifying the Agent Based Modeling and Simulation (ABMS) approach for metabolic engineering purposes, summarizing the expected performance and assessing the usefulness of this work to several areas including computing technology and computational biology.

Chapter 2 Literature Review

2.1 Metabolic Engineering – An overview

Metabolic engineering, also known as molecular breeding (Kellogg et al.,1981), in vitro evolution (Timmis et al., 1988), pathway engineering (MacQuitty, 1988; Tong et al., 1991) and cellular engineering (Nerem, 1991) involves directed modification of cellular metabolism and properties through the introduction, deletion and/or modification of metabolic pathways by using recombinant DNA and other molecular biological techniques (Lee and Papoutsakis, 1999). This field has emerged as a result of overwhelming interest in utilization of improved strain of microorganisms for medical and industrial purposes. The primary goal of this field is to invoke desirable metabolic behavior in living cells. Recent advances in different scientific disciplines including molecular and computational biology, genetics, computer technology along with various application tools have led this young field to grow fast and become one of the most attractive research areas in the 21 century.

Like other fields of engineering, metabolic engineering also encompasses the two defining phases of analysis and synthesis. From the engineering perspective of design and analysis, it is very important to have analytical tools such as a mathematical or computational model; e.g. a dynamic simulator of metabolism that is based on the fundamental physicochemical laws and principles. Such models can be used to systematically analyze and thus design a new or redesign an improved strain. The methods of recombinant DNA technology, DNA splicing or genetic engineering could then be applied to achieve the desired changes in the genotype of the organism of interest. On the design side, metabolic engineering focuses on integrated metabolic pathways instead of individual reactions. On the analysis side, it emphasizes on metabolic fluxes and their control, thus it implies a holistic examination of the complete biochemical reaction network and concerns itself with issues of pathway synthesis and thermodynamic feasibility.

2.1.1 Metabolic Network analysis

Metabolism is considered as the “chemical engine” for keeping the cellular system living. The last two to three decades of research on metabolic analysis has illustrated the need to quantify systemic aspects of cell metabolism. There are significant motivations for metabolic dynamics study. An extensive analysis and a quantitative description of cellular metabolism is not only important to implement metabolic changes to achieve specific functionality, but also has great importance to our understanding of cell biology. Important applications of metabolic analysis include strain design for the production of therapeutics, assessment of the metabolic consequences of genetic defects, synthesis of systematic methods to combat infectious disease and so forth (Liao, Hou and Chao, 1996). Quantitative and systematic analysis of metabolism is thus of substantial importance.

The mathematical modeling of metabolic networks dates back to the mid 1960s. The study of the genetic control and dynamic simulations of simple metabolic loops emerged with the availability of computers and knowledge of metabolic regulation. It received further impetus with the invention of modern computational and analytical tools and extensive research on cell biology. The systemic nature and the functional complexities of metabolism are now apparent. The focus then turned to developing methods that could shed light on various metabolic events. Methods for sensitivity analysis of metabolic regulation begun in the 1960s (Savageau, 1969) and continued into the 1970s (Heinrich et al., 1977 and Kacser and Burns, 1973) and resulted in the biochemical systems theory (BST), flux balance analysis (FBA) and the prominent metabolic control analysis (MCA).

The development of Recombinant DNA technology in early 1970s was of great historical significance and it ushered the era of engineering or designing the biological components. The first report of bacterial gene splicing appeared in 1972 (Jackson, D. A., Symons, R. H., and Berg, P., 1972). Gradually these new techniques of recombinant DNA or gene splicing has become very useful and prominent tool for the researchers to make changes in underlying cellular determinants and to alter the characteristics of industrial strain instead of being content with designing equipment and operating strategies. Consequently, various terms representing the potential application of recombinant DNA technology towards directed pathway modification were coined (in *vitro* evolution, cellular engineering, molecular breeding, etc) and the field of metabolic engineering emerged. (Bailey, J. E., 1991, & Stephanopoulos, G. & Vallino, J. J., 1991).

In the early years of metabolic engineering, improvement of cellular processes were performed through successive mutagenesis and selecting strains with desirable qualities. Despite the success of the approach for a number of cases, it has been found that the theoretical yield of the product is not always attainable through random mutagenesis and selection procedures. The advent of recombinant DNA technology as well as advances in molecular biology and genetic engineering empowers metabolic engineers with the increasing ability to create any desired cellular modification.

From the early stages of metabolic engineering, the intention of the researchers were elucidating the systemic behavior of metabolic networks; consequently designing and developing a complete kinetic model of cellular metabolism had become the primary scientific goal. This quest to get comprehensive dynamic models of metabolism for designing strain with perfectly directed pathways and perfect functionality still remains unfulfilled due to the lack of an overall comprehension of *in vivo* metabolic processes. However, with the recent advancement in technology, detailed information on metabolic components, in particular strains, is now increasingly available. This led us redesign or reconstruct the metabolic networks and also ascertain information regarding the structure and stoichiometry of the metabolic reaction networks.

2.1.2 Scope of Metabolic Engineering

Metabolic Engineering is a highly multidisciplinary field. Basic metabolic maps and comprehensive information about the mechanisms of biochemical reactions, their stoichiometry, regulation, enzyme kinetics are provided by biochemistry. Genetics and molecular biology supply necessary tools and knowledge for the construction of well characterized genomic database as well as for the studies on flux control. A detailed and more integrated picture of cellular metabolism can be gathered from the study of cell physiology and thus a comprehensive platform for metabolic rate study and physiological state representation. Applications of engineering approaches of integration, quantification and analysis to study biological system also can contribute to the field of metabolic engineering.

The primary goal of metabolic engineering is to control the flux (Stephanopoulos, G., 1999). Metabolic flux is defined as the rate at which material is converted via metabolic reactions and pathways. For flux control, the factors influencing the flux must be understood. Since fluxes are a determinant of physiological state, the complete understanding of flux control of cellular metabolism will help in explaining the genotype-phenotype relationship of cell. However, the goal of metabolic engineering can be classified into several distinct objectives (Cameron, D. C., and Tong, I. T., 1993) such as enhance the yield of the host's natural products or adding novel production capacity to the cell (basically addition of new genes), the addition of metabolic processes that the cell normally does not possess (Keasling Lab, 2007), and the general modification of cellular properties to improve the cell's potential utility.

Production of valuable chemical for therapeutic or industrial purposes applying microorganisms is another major application of metabolic engineering. Microbes are typically redesigned or modified to produce chemicals that are too expensive to produce by chemical synthesis. Examples of such compounds are vitamins like riboflavin (Sauer et al., 1997). Sometimes metabolic engineering strategies are employed to enhance the cell's native production (Ikeda et al., 1994) as products like acetic acid (Park et al., 1989), ethanol (Ohta, et al., 1991), amino acids (Ikeda, M., and Katsumata, R., 1994) and various antibiotics (Henriksen et al 1996), and so on. Metabolic engineering has also been used to impart new product production capability. Examples are biopolymers (Slater, S., Gallaher, T., and Dennis, D., 1992) , antibiotics (Weber et al,. 1991), pigments, etc.

Designing organisms with added metabolic function is of great importance for environmental and bio-pharmaceutical applications. Winter *et al.,* (1989) reported utilization of genetically engineered *E. coli* for effective degradation of trichloroethylene. Similarly Martin *et al.,* (2003) used engineered mevalonate pathway in *E. coli* for terpenoids production. Additional metabolic processes can be added such that different substrates can be used in an industrial process for the production of different metabolites, amino acids, vitamins, antibiotics, etc (Wood and Ingram, 1992).

Moreover, metabolic engineering allows alteration in genotype such that the phenotype exhibits cellular properties that are beneficial for the organisms utilized in industrial processes (Aristidou et al., 1990.,; Yang et al., 1999). Flux control allows redirection of metabolic flux away from toxic byproduct to less toxic ones (Aristidou et al., 1995).

New and diverse opportunities for metabolic engineering have emerged quickly in this genomic era (Alper and Stephanopoulos, 2004). This advancement in genomics has led us to a position to study metabolic characteristics as a function of the entire genome. However, extensive bioinformatics methods and experimental effort is required to reveal the hidden information regarding molecular interaction and genetic regulation.

2.2 Modeling of Metabolic reaction network

2.2.1 Current Modeling Approaches

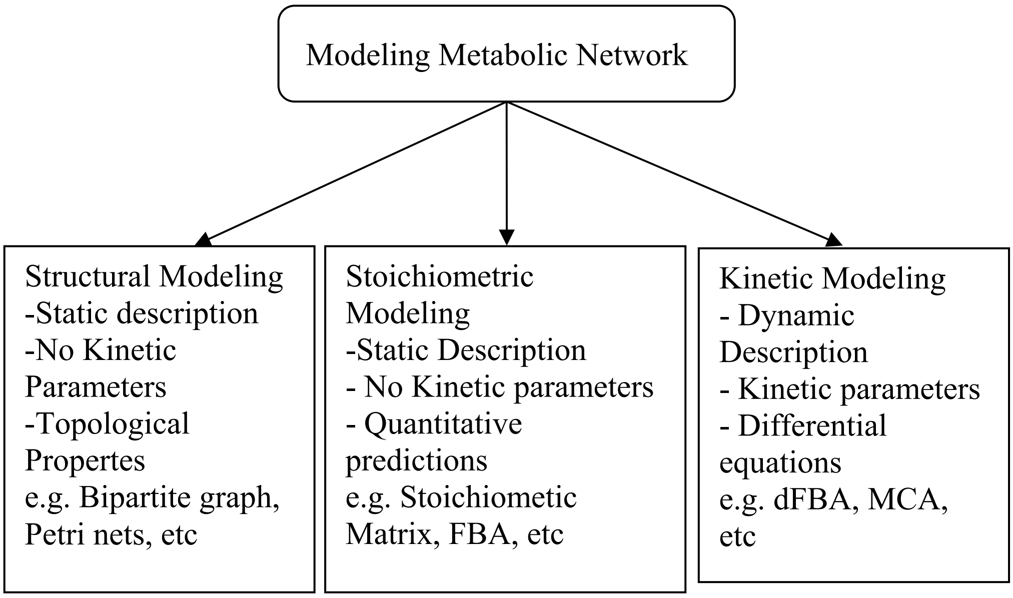


Figure 2-1: Major Metabolic Network Modeling approaches

Intracellular molecular networks such as a metabolic pathway can be modeled in multiple ways. These modeling approaches are broadly classified into three basic categories, namely topological or structural modeling, stoichiometric modeling and kinetic modeling. However, the power of a model strongly depends on its basic modeling assumptions, the simplifications made, and the data sources used.

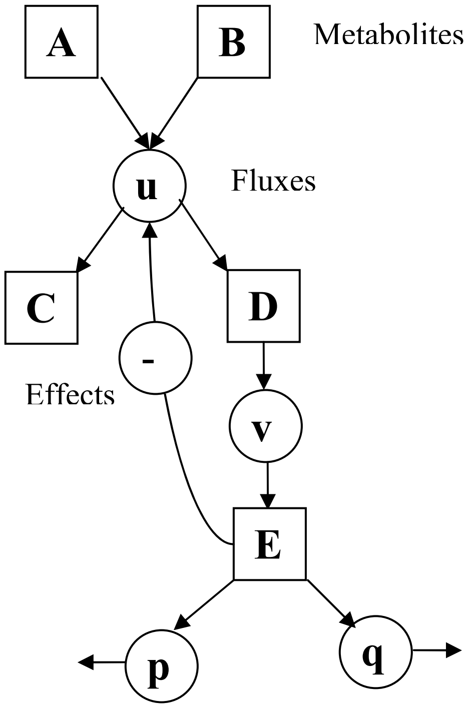
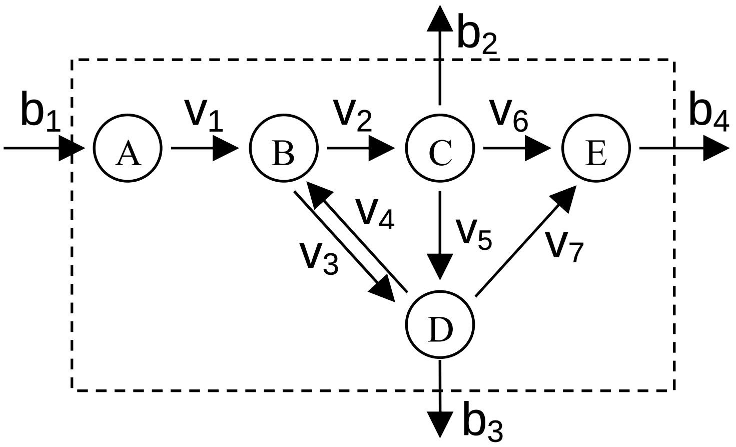


Figure 2-2: A simplified representation of Network Structure

Structural methods can detect possible regulatory structures but they do not answer if these regulation mechanisms are quantitatively relevant in a certain physiological state of the cell.

Another simple way of modeling is to assume the quasi-steady state and represent all the reactions involved in the networks using a stoichiometric matrix. The rows of the matrix represent the metabolites in the network and its columns represent the reactions in the network. Basically, it is a matrix representation of a system of linear algebraic equations and is amenable to all forms of mathematical operation.

Hence, it is widely used to describe the stoichiometric analysis of intracellular molecular networks and for genome scale metabolic modeling (Papin et al., 2004; Price et al., 2004). Figure 2-3 describes how a stoichiometric matrix is formed for a simplified metabolic network. As it pertains to genome scale metabolic studies, the stoichiometric matrix can be directly constructed from knowledge of an organism's metabolic genotype, which may now be realistically determined from the results of genome annotation (Schilling et al., 1999). It describes the topological structure and the architecture of the network, and its properties is a must for any simulation of biochemical reaction networks (Heinrich and Schuster, 1996). Stoichiometric model has been extensively used for metabolic flux analysis and flux optimization (Varma and Palsson, 1994).



Balance Equations:

A: $-v_1+b_1=0$
B: $v_1+v_4-v_2-v_3=0$
C: $v_2-v_5-v_6-b_2=0$
D: $v_3+v_5-v_4-v_7-b_3=0$
E: $v_6+v_7-b_4=0$

Matrix Notation
$\mathbf{S}\cdot\mathbf{v}=\mathbf{0}$

Stoichiometric Matrix

$$\mathbf{S}=\left[\underbrace{\begin{matrix}1&0&0&0&0&0&0\\1&-1&-1&1&0&0&0\\0&1&0&0&-1&-1&0\\0&0&1&-1&1&0&-1\\0&0&0&0&0&1&1\end{matrix}}_{\textit{Internal Fluxes}}\;\underbrace{\begin{matrix}-1&0&0&0\\0&0&0&0\\0&-1&0&0\\0&0&-1&0\\0&0&0&-1\end{matrix}}_{\textit{Exchange Fluxes}}\right]$$

Figure 2-3: Stoichiometric Matrix for a simplified Network (source: www.cs.technion.ac.il)

All the approaches discussed above mainly capture the structural aspects of the network without considering the kinetic properties of the enzymatic reactions involved in the network. Kinetic models, however, do incorporate enzyme kinetic information. From the stoichiometric description of the network, a kinetic model can be constructed by supplying the rate equations for all the reactions in the network. A rate equation expresses the rate of an enzymatic action in terms of its kinetic parameter and the concentrations of its substrates, products and effectors. For dynamic simulation of the system, the network should be characterized by its stoichiometric matrix, parameterized rate equations, initial conditions, and specified environment (identification of fixed concentrations, inflows and outflows). The state of the network is represented by a complete set of concentrations of intermediates. The set of initial concentrations of intermediates at the initial time point of the calculation of the dynamics of the network is referred to as the initial state of the network. Several researchers are trying to combine stoichiometric information with high quality kinetic data, whenever available. For example, Covert and Palsson, (*2002*) and Covert and Palsson, (*2003*) have incorporated the regulation of gene expression to flux balance analysis (FBA). Mahadevan et al (2002) have extended FBA (i.e., dFBA) to describe the dynamic behavior of metabolic system. Gadkar et al (2005) included kinetic expressions in dynamic FBA to optimize the concentration of a targeted product molecule. Based on the physicochemical conditions under which cellular reactions take place in the organism, the dynamics of the network can then be monitored using dynamic simulation techniques.

Over the past decades, the mathematical and numerical analysis of detailed kinetic core models has made numerous significant contributions to elucidate and understand the general principles of metabolic regulation and control. Such extensive effort led in the formulation of Metabolic Control Analysis (MCA), a mathematical tool to describe the control and regulatory properties of metabolic systems and more recently, extensive initiatives have been made to extend this “bottom-up” approach towards more comprehensive large scale dynamic model of cellular metabolism (Ishii et al, 2004). One example of bottom-up approach is multi agent based modeling and simulation approach.

2.2.2 Agent Based Modeling

Agent based modeling is fast emerging as a new paradigm for engineering complex, distributed systems. Agent technology is also suitable for the analysis, design, and construction of intelligent systems. Agent can be defined as a computer system that is situated in some environment, and that is capable of autonomous action in this environment in order to meet its design objectives. Multi-agent systems are systems composed of multiple interacting agents.

Wooldridge (1998) has described certain characteristics of an agent. According to Wooldridge, an agent, in general, is a system with the following properties

- *Autonomy*: agents can make decisions about what to do without direct external intervention of other systems.
- *Reactivity*: agents are situated in an environment, can perceive it (at least to some extent) and are able respond to the changes in it (i.e. are able to react).
- *Pro-activeness* (or *proactivity*): agents do not simply react to changes in the environment, but are also able to take the initiative.

- *Social ability*: agents can interact with other agents and participate in social activities.

2.2.3 Equation based model vs Agent based model

Various computer simulation models have been developed to better understand complex biochemical systems. These include equation-based models (EBM), agent-based models (ABM), deterministic models, and stochastic models. In 1998, Van Dyke Parunak and others compared the effectiveness of EBMs versus ABMs for modeling complex systems, and concluded that ABMs were more suitable for this purpose because ABMs can model overall behavior of complex systems based on the behavior of individual components. Thus, the overall systems behavior emerged from different interactions among individuals can be captured in ABMs.

Both ABM and EBM approaches simulate the system by constructing a model which is then executed in a computer. The differences are in the form of the model and how it is executed. In ABM, the model consists of a set of agents that encapsulate the behaviors of the various individuals that make up the system, and execution consists of emulating these behaviors. On the other hand, in EBM, the model is a set of equations, and execution consists of evaluating them. There are two basic differences between these two approaches:

- Relationships among the entities that are modeled
- Level of detail at which ABM and EBM focus their attention

Both these approaches mainly deal with two kind of entities; individuals and observables. Observables are measurable characteristics of interest that usually vary over time. They may be associated with individuals or with the collection of individuals as a whole.

EBM begins with a set of equations that express relationships among observables. The evaluation of these equations results in the evolution of the observables over time. These equations may be algebraic or ordinary differential equations with the ability to capture variability over time or partial differential equations with the ability to capture variability over time and space. Though those relationships result from the interactions of the individuals, but are not represented explicitly by EBM.

ABM begins not with equations that relates observables to one another, but with behaviors through which individuals interact with one another. These behaviors may involve multiple individuals directly or indirectly through a shared environment. The relationships between individuals and observables can be summarized as follows:

- Individuals are characterized, separately or in aggregate, by observables, and affect the values of these observables by their actions.
- Observables are related to one another by equations.
- Individuals interact with one another via their behaviors.

The second fundamental difference between ABM and EBM is the level at which the model is focused. EBM tends to make extensive use of system-level observables, since it is often easier to formulate convincible closed form equations with such quantities. The natural tendency in ABM is to define agent behaviors in terms of observables accessible to the individual agent. One agent behavior may depend on an observable generated by other individual, but does not directly access the representation of those individuals' behaviors. These fundamental differences in modeling the system impart some significant advantages in application of ABMs for complex system modeling.

- ABMs make it easier to distinguish physical space from interaction space. In many applications, physical space helps define which individuals can interact with one another. EBM such as ODE method cannot incorporate spatial arrangement at all. PDEs provide a parsimonious model of physical space but are unable to distinguish it from interaction space (Pogson et al., 2006).
- ABMs offer an additional level of validation. Both ABMs and EBMs can be validated at the system level, by comparing model output with real system behavior. In addition ABMs can be validated at the individual level, since the behaviors encoded for each agent can be compared with local observations on the actual behavior of the individuals.
- ABMs support more direct experimentation.
- ABMs are easier to translate back into practice. If the model is expressed and modified directly in terms of behaviors, implementing the recommended change is very simple and easy.
- In many cases, ABMs give more realistic results than EBMs, with manageable levels of representational details (Parunak et al., 1998).

However, one of the major challenges of ABMs is in designing a multi-agent model and simulator from actual process description to the large number of parameters in the model.

2.3 Agent Based Modeling and Simulation in Biology

Multi agent system is not a very widely used technique for modeling biological systems. However, currently agent based programming is becoming popular in different fields for modeling complex systems. The concept of using autonomous multi-agents to describe cells and cellular behavior was first proposed by Paton, (1993). Gonzalez *et al.,* (2003) developed a system named Cellulat. Cellulat represents proteins and other components participating in intracellular signaling programmed as "internal autonomous agents" where communication with external medium takes place through "interface autonomous agents".

Alur *et al.*, (2002) described a hybrid system where agents are characterized by a continuous state x and a collection of discrete modes. There are two types of agents. Process agents or P agents capture the dynamics involved in transcription, translation, protein binding, protein –protein interaction, cell growth, etc. System agents or S agents describe the accumulation or degradation of proteins, cells, DNA in terms of concentration or numbers. Each mode is represented by a set of ODEs and the current state. Change of state occurs through the set of ODE s of currently active modes.

Katare and Venkatasubramanian (2001) applied agent-based approach to study the behavior of microbes in a binary substrate environment. Their cellular model consists of one Nucleus agent, one environment agent and different types of cellular organelle agents.

Burleigh *et al* (2003) used swarms, another agent based approach to model the regulating process of *lac*-operon. The random movement of agents lead them to interact with other agents and these interactions are governed by simple rules.

Taivo Lints (Tallinn University of technology) are trying to implement JAVA Agent Development Environment (JADE) to model the mechanism that causes the initiation of DNA replication. This model contains four types of agents: Environment, Bacterium, DnaA (a protein) Factory and DNA. This simple agent based model was reasonably successful in simulating the cell division cycle.

2.3.1 Tools available for Agent Based Modeling

With the intense research in the realm of agent-based modeling under the distributed artificial intelligence domain, scores of tools have been developed for building ABMs, in particular by making use of object oriented programming like JAVA, C++, etc. The development of these ABM tools came up for applications in social simulations (Shoham, Y., and Tennenholtz, 1997) and studying complex behavior. StarLogo is among the earliest ABM tools and consequently some other ABM tools such as the SWARM, StarLogoT (a variant of StarLogo), REPAST, ASCAPE, NetLogo, etc came and became very useful tools for the programmer. The industrial circuit also has actively taken part in the development of these agent-based tools. Notable among them are RAISE and ABLE by International Business Machines (IBM) Corporation Limited, JADE by Telecom Italia and the open source project – ECLIPSE. **Jadex** is another most recent agent-oriented reasoning engine for writing rational agents with XML and the Java programming language. It is developed by the Distributed System Group, University of Hamburg, Germany and is an extension to the JADE^{TM} multi-agent platform. Currently, two mature adapters are available, the first adapter is available for JADE^{TM} and the second is the Jadex Standalone adapter which is a small but fast environment with a minimal memory footprint. (VSIS project web site, University of Hamburg, 2007). In this thesis, JADE has been used as the ABM platform.

2.3.2 Introduction to JADE

JADE (**J**ava **A**gent **DE**velopment Framework) is a software development framework for developing multi-agent systems and applications compatible to FIPA (The Foundation for Intelligent Physical Agents) standards for intelligent agents. It has been written in Java programming language and includes two main basic utilities - a

FIPA-compliant agent platform and a package to developed java based agents. JADE provides the following features for developing Multi Agent Systems (MAS):

- Distributed agent platform. The agent platform can be split among several hosts executing only one java application in each host. Agents are implemented as Java threads and live within Agent Containers that provide the run time support to agent execution.
- Graphical user interface to manage several agents and agent containers from a remote host.
- Built-in debugging tools.
- Mobility- an agent can be moved from one platform to another (if necessary) with its state and code.
- Jade schedules the agent behaviors (methods) in a non-preemptive manner. Its behavior model supports execution of multiple, parallel and concurrent agent activities.
- FIPA-compliant Agent Platform, which includes the AMS (Agent Management System), the DF (Directory Facilitator), and the ACC (Agent Communication Channel). These components are automatically activated at agent start-up.
- Many FIPA-compliant DFs can be started at the run time in order to implement multi-domain applications, where each domain is a logical set a agents, whose services are advertised through a common facilitator. Each DF inherits a GUI and is capable of registering, deregistering, modifying and searching for agent descriptions as well as federating within a network of DF's).

- Efficient intra-platform and inter-platform transport of ACL message. Messages are transferred as Java objects. When crossing platform boundaries, the message is automatically converted to the FIPA compliant syntax, encoding, and transport protocol.
- Support for user defined content languages and ontologies.
- In process interface to allow external applications to launch autonomous agents.

2.4 Reconstruction of metabolic network model

The successful application of computational methodology for metabolic network analysis depends on the availability of a complete and comprehensive model of the metabolic pathway. Metabolic reconstruction provides the opportunity to build such complete and comprehensive model. The objective of metabolic reconstruction is to provide a detailed description of metabolic network in terms of all the pathways, components and their interactions. The activities required for network reconstruction are summarized in figure 2-4.

The first step is to identify all the metabolic pathways for the organism from available databases like KEGG, ECOCyc, MetaCyc, etc. However, such reconstruction is not flawless as most of the time the available databases are incomplete. A key challenge in genome scale reconstruction is to elucidate these gaps and subsequently bridge them. Green et al (2004) formalized a bayesian framework to identify missing enzymes in a network using sequence homology related metrics. Data based approaches are also used to identify candidate genes by measuring similarities with metrics such as mRNA co expression data (Kharchenko et al. 2004) or by phylogenetic profiles (Chen and Vitkup, 2006). In addition to phylogenetic profile, Kharchenko et al. (2006) also proposed use of multiple types of combined evidence including clustering of genes on the chromosome and protein fusion events.

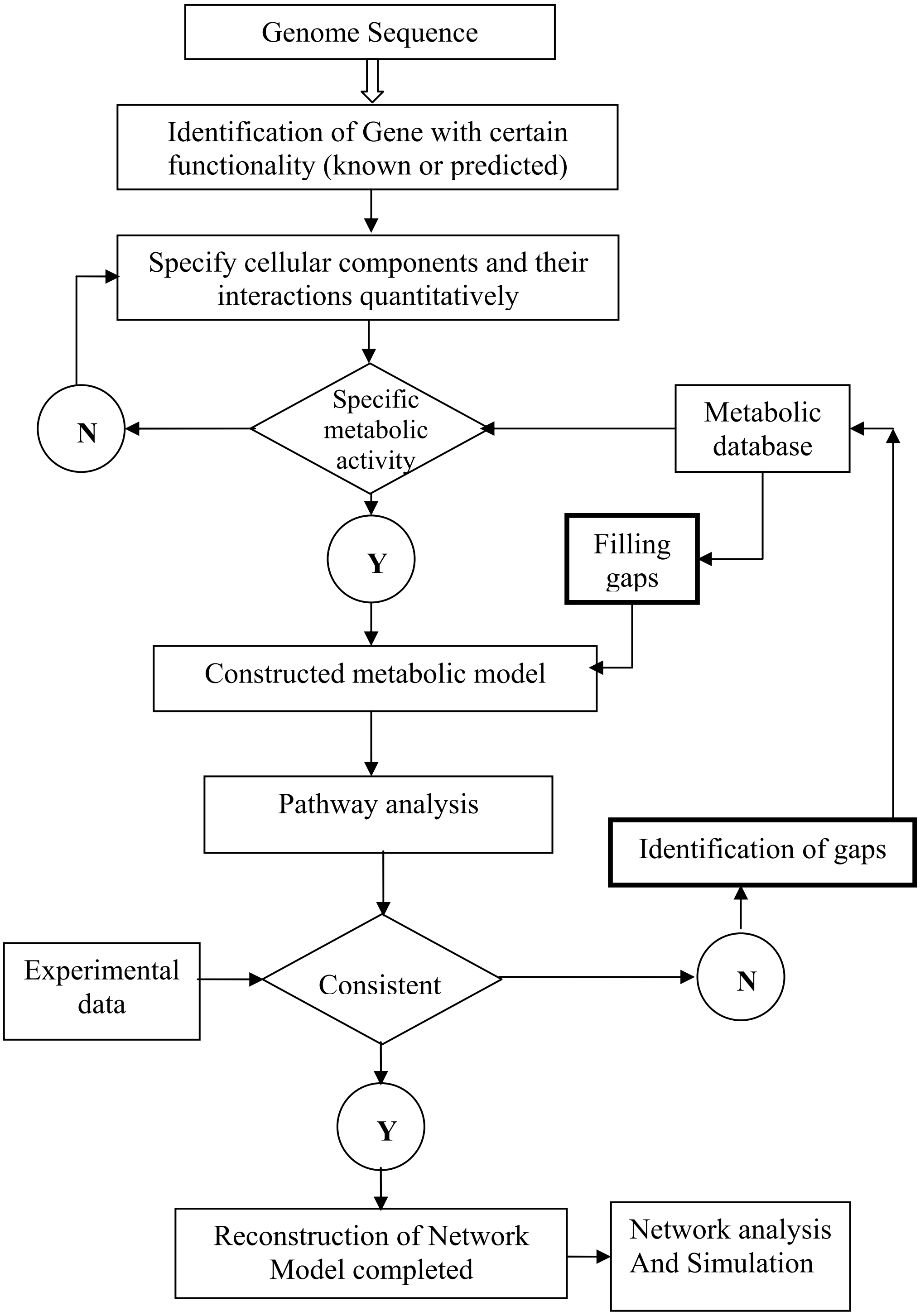

Genome Sequence
Identification of Gene with certain functionality (known or predicted)
Specify cellular components and their interactions quantitatively
N
Specific metabolic activity
Metabolic database
Y
Filling gaps
Constructed metabolic model
Pathway analysis
Identification of gaps
Experimental data
Consistent
N
Y
Reconstruction of Network Model completed
Network analysis And Simulation

Figure 2-4: Basic steps involved in network reconstruction

The common approach of all the methods mentioned above is to postulate a set of candidate genes and the evaluate their likelihood that these are present in the metabolic network using different scoring metrics. Besides these, Bodik and Rasche (2001), Pellegrini et al. (2001), Osterman and Overbeek (2003), and Notebaart et al. (2006) introduced variety of genome scale analyses to identify missing genes by conducting sequence based comparisons of entire genomes and inferring possible metabolic functions for different microorganisms. In 2006 Reed et al. identified the location of missing metabolic functions in *E. coli* by observing the difference in prediction based on designed genotypic and phenotypic behavior and actual in vivo growth.

2.5 Scope of the thesis

As discussed in the previous section, the key challenges in developing a comprehensive metabolic model from the available databases, is to elucidate the gaps in the network and subsequently filling those gaps. This requires extensive iteration when attempted. The focus of this work is to develop efficient methods to effectively identify any inconsistency in the network as well as to help the researcher fill them with minimum effort.

Chapter 3 Agent Based Modeling of Metabolic Networks

The promise of understanding the relationship between the genome and the physiological functionality of organism is the prime incentive to reconstruct metabolic networks as well as to apply developed mathematical or in *silico* models for biological discovery and engineering applications like, system analysis, prediction, design, optimization, etc. Various approaches are available for modeling biological systems. Due to its inherent complexities, developing complete model of biological systems is onerous and the common equation-based modeling approaches are often unable to capture emergent properties. Given its ability to capture the emergent properties of complex interactive system, agent based modeling is increasingly becoming popular in the scientific communities. Our goal is to explore the applicability of agent-based techniques to model the metabolic networks. Consequently, a new agent-based model has been proposed with the ability to simulate and analyze the metabolic network of prokaryotic cell.

One of the most important and significant feature of ABMs is the ability to model any system from the perspective of individual agents. From this perspective, a metabolic network could be considered as a network of individual biochemical reactions. Each reaction is an individual entity (agent) that interacts with other reaction agents based on common metabolites. Each reaction is therefore modeled as an agent with interactions via reactants, products, and enzymes – which are hence modeled as the agent's attributes. The interaction among the reaction agents can be modeled as occurring sequentially or in parallel. Here, we have adopted the sequential mode of interactions where each member among the set of reactions execute one after another based on the availability of their respective reactant metabolites and the activity of the corresponding enzymes.

The rest of this chapter describes the basic architecture in the agent-based modeling of metabolic networks. The next section will cover the details of the ABM architecture, agent' functionalities, and their interactions. Section 3.2 provides an illustration of the emergence of the network model from the interaction between the agents. The application of the agent-based model to finding gaps in metabolic networks is proposed in Section 3.3 and illustrated using several examples in Section 3.4. Methods for filling gaps are proposed and illustrated in Section 3.5.

3.1 Model Architecture

The metabolic network of a cell is modeled with three types of agents – Cytoplasm agent, Reaction agent, and Scheduler agent. The first two mimic physical entities in the cell while the last serves as a enabling entity for analysis and simulation. Prior to the description of metabolic network analysis, important features of these agents are explained.

3.1.1 Cytoplasm Agent

The Cytoplasm agent is a key agent in the ABM and mimics the cytoplasm in the cell. It is designed to serve as the intracellular environment and contains information regarding the metabolites' concentration (both intracellular and extracellular) as well as enzymes' activity. The cytoplasm agent communicates with the reaction agents to ensure that this information remains up-to-date. During simulation, it also bootstraps the whole execution process by initializing itself as well as the other agents. Next, we describe the implementation of the Cytoplasm agent.

The JADE implementation of the Cytoplasm agent consists of two main components – a text file named cytoplasmIni and a Java agent class file named CytoplasmAgent. The text file cytoplasmIni contains information about:

- System summary including the Number of Reactions, Metabolites and Enzymes
- Metabolites including the metabolite ID, Name, and initial Concentration
- Enzymes including the Enzyme ID, Name, and Activity

Since the Cytoplasm agent serves as the repository of the information regarding the state of the system, whenever there is any query regarding metabolite concentrations, or enzyme activity or effectors concentrations from any agent, the Cytoplasm agent serves that query. The Cytoplasm agent executes its activities using two principal behaviors – CytoplasmInitialization and CytoplasmQueryServer. The former behavior initializes the whole agent model by instantiating all the reaction agents and the scheduler agent via JAVA RMI (remote method invocation). The latter behavior serves as the main communication module and provides data storage and data updating capabilities. It also serves queries related to the network structure that are necessary for gap finding (see Section 3.3). Table 3-1 summarizes these activities of the Cytoplasm agent.

Table 3-1: Summary of Cytoplasm agent's activities

Behavior name and Type	Component	Functionality
CytoplasmInitialization OneShotBehaviour	readCytoplasmIniFile	Collecting initial condition of the cytoplasm from cytoplasmIni.txt file
	createReactionAgents	Create and initiate Reaction agents
	initiateScheduler	Create and Initiate Scheduler agent
	startExecution	Asked the Scheduler agent to trigger the first reaction.
CytoplasmQueryServer CyclicBehaviour	metabolicQueryHandler	Serves query regarding metabolites
	eznymeQueryHandler	Serves query regarding Enzyme
	reactantUpdateReceiver	Update concentration of corresponding reactant metabolites being informed from a reaction agent
	productUpdateReceiver	Update concentration of corresponding product metabolites being informed from a reaction agent
	updateScheduler	Inform Scheduler agent regarding the new metabolites
	printResult	Print the final Condition of cytoplasm in terms of its metabolites' concentrations as a text file.
	printReactionStatus	Prints the information regarding the execution of a reaction whether it is executed or not after simulation stops.
	createTriggeredReactionList	Makes a list by adding all the reaction fired during the simulation.
	metaboliteStatusList	Serves to distinguish between participating and non-participating metabolites
	startBacktracking	Search root cause of gap start from a metabolite defined by the user.
	serviceSearch	Search the DF for reactions that produce certain metabolite
	actionList	Reports required action to the user.

3.1.2 Reaction Agent

The Reaction agents form the core of the model and provide the individual-based perspective of the metabolic system. Prior to execution, a reaction agent searches for the information regarding the present concentration of its input metabolites (reactants) and output metabolites (product) as well as the activity of corresponding enzymes. If all the required metabolites are present and their enzymes active, then the reaction will be carried out and the corresponding products produced. Next, the concentrations of the metabolites are updated according to the stoichiometry and the extent of conversion based on the limiting metabolite as described below:

Method of calculation: Consider the reaction:

$$aA + bB \rightarrow cC + dD$$

and let $[A]_i$, $[B]_i$, $[C]_i$, $[D]_i$ be the initial concentrations of the metabolites. Assuming, γ_{max} as the conversion of the limiting reactant, the extent of reaction, ξ is calculated as

$$\xi = \gamma_{\mathbf{max}} \times \mathbf{min} \left\{ \frac{[A]_i}{a}, \frac{[B]_i}{b} \right\}$$

The concentration of the raw materials and products are then updated as:

$$[A] \leftarrow [A]_{in} - a \times \xi$$

$$[B] \leftarrow [B]_{in} - b \times \xi$$

$$[C] \leftarrow [C]_{in} + c \times \xi$$

$$[D] \leftarrow [D]_{in} + d \times \xi$$

The new concentrations are communciated to the Cytoplasm agent.

Like the Cytoplasm agent, the JADE implementation of the Reaction agent also consists of two components – an initialization file and a behaviors file.

The initialization text file consists of information about:

- Reactants including the number of reactants, reactants IDs, and their stoichiometric coefficients
- Products including the number of products, Products' IDs, and their stoichiometric coefficient.
- Enzymes including the number of Enzymes and their IDs

The behaviors Java file is used by all the reaction agents as a template for their activities. Reaction agents offer two behaviors – *ReactionInilializingBehaviour* and *ReactionSimulationBehaviour* as summarized in Table 3-2.

Table 3-2: Summary of Reaction agents' activities

Behaviour name and Type	Component	Functionality
ReactionInitialization OneShotBehaviour	readReactionInput	Collecting information regarding the corresponding reaction.
	registrationWithDF	Register with DF agent adding reactant metabolites as a service.
ReactionSimulation Behaviour	startReaction	• Send query to Cytoplasm regarding metabolites, enzymes. • Calculate new concentration after the reaction based on the stoichiometry • Send update information to Cytoplasm. • It also keep track of whether reaction fully executed or not.
	informScheduler	After the execution of each reaction it sends a message to Scheduler agent.

3.1.3 Scheduler Agent

The main purpose of the Scheduler agent is to monitor and control the execution of the reaction agents. The Scheduler agent monitors the concentrations of all the metabolites; when new metabolites are produced, it evaluates if new reactions become possible and then triggers them. The process continues until all possible reactions have been triggered.

The Scheduler agent is implemented in JADE using a cyclic behavior, named ScheduleManagementBehaviour. The main data structure in this agent is a list, which serves as the queue of reactions to be triggered. When the scheduler is initialized, it adds all possible reactions into the queue. When new metabolites are produced during the execution of a reaction, new reactions that become possible are included dynamically into the Reaction queue. The whole simulation runs while there are Reactions in the queue. Table 3-3 shows the basic activities of the Scheduler agent.

Table 3-3: Summary of Scheduler agent's activities

Behavior name and Type	Component	Functionality
ScheduleManagement Behaviour CyclicBehaviour	searchAgent	When informed about the new metabolite it search the DF agent for reactions use that metabolite as input and updates the Agent queue.
	triggerSimulation	On information from Reaction agent that the previous reaction has completed, or from Cytoplasm agent to start the simulation, the next available reaction is triggered for execution.
	requestPrintResult	When the execution of all possible reactions is completed, it sends a request to the Cytoplasm agent to print the final results.

3.1.4 Directory Facilitator

The Directory Facilitator (DF) agent is an in-built agent in JADE. The DF is a centralized registry that associates service-descriptions to agents. It plays a major role in the execution of the model. Each reaction agent is registered with DF. Specifically, each reaction agent explicates its ability to convert a set of inputs (reactants) to outputs (products) as a description. Whenever a metabolite is updated, the Scheduler agent queries the DF for agents which provide the service of metabolizing that metabolite. The input to the DF is the service description (Service name- <Metabolite-Id> and Service type- "metabolite") and the result from the DF is the details of the agent (as AID of the Reaction agents) if any that use that metabolite as input (reactant). The Scheduler agent updates the reaction queue by including the reaction agents obtained from the DF query. The DF also helps in gap identification of network as described in detail in Section 3.3.1.

3.1.5 Simulation and Emergence of Metabolic Network using the agent-based model

The entire simulation procedure is divided into two parts: initialization and execution. Simulation starts with the Cytoplasm agent.

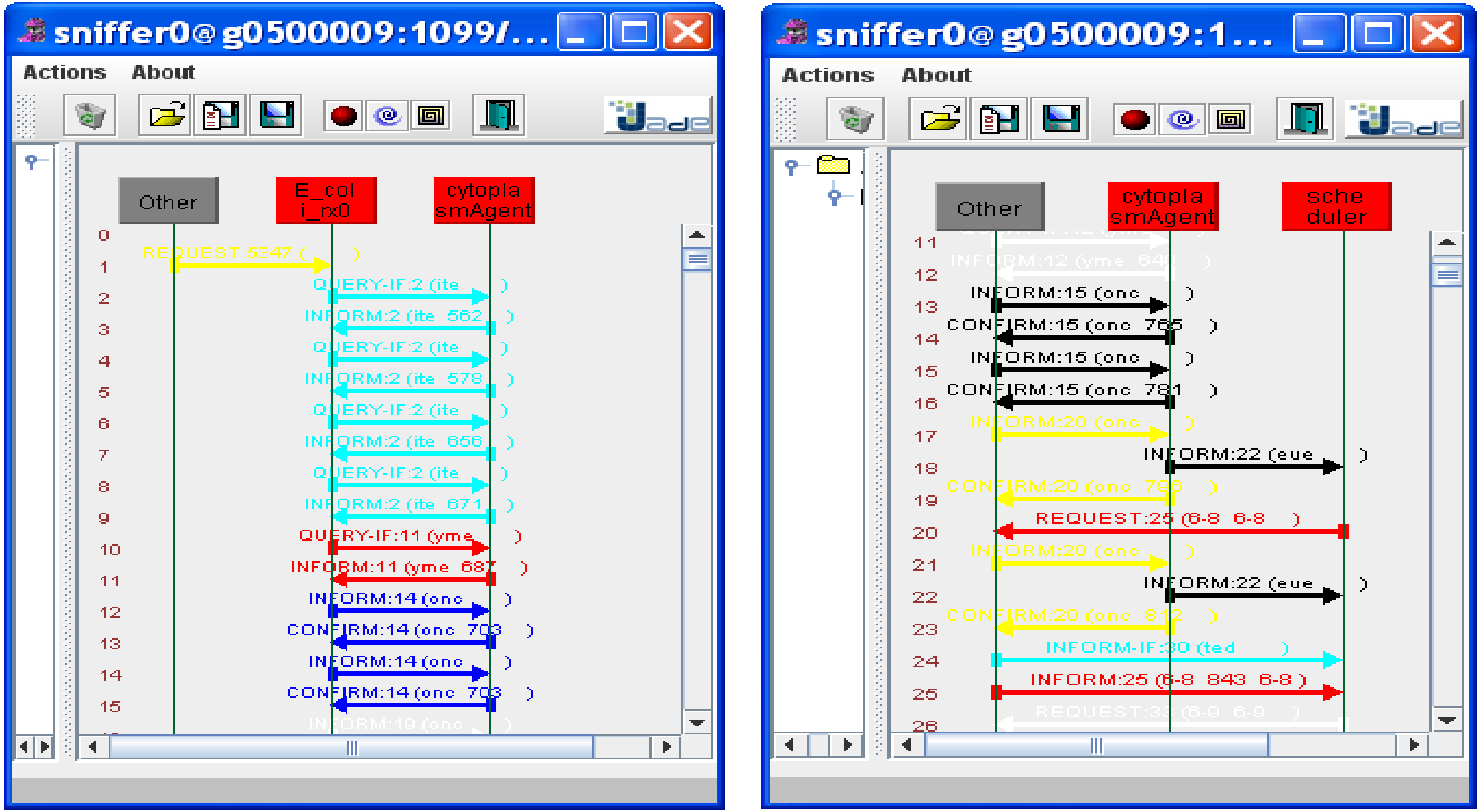
1. Cytoplasm agent is initialized and it gets information from the cytoplasmIni.txt file.
2. Based on the information, it creates all the Reaction agents and the Scheduler agent.
3. When created by Cytoplasm agent, each Reaction agent initiates itself by getting information from its initialization file. It also registers itself with the DF agent.
4. Then, the Cytoplasm agent sends a message to the Scheduler agent to initiate the queue with all available reaction agents.

When the initialization of the model is completed, it is ready for execution. This step also starts with the Cytoplasm agent.

1. The Cytoplasm agent sends a message to the Scheduler agent to start the execution.
2. The Scheduler agent picks the first element in the Reaction queue and sends a request message to that reaction agent to execute.
3. The Reaction agent then sends metabolite queries using ACLMessage QUERY_IF to the Cytoplasm agent
4. The Cytoplasm agent serves these queries by INFORM messages.

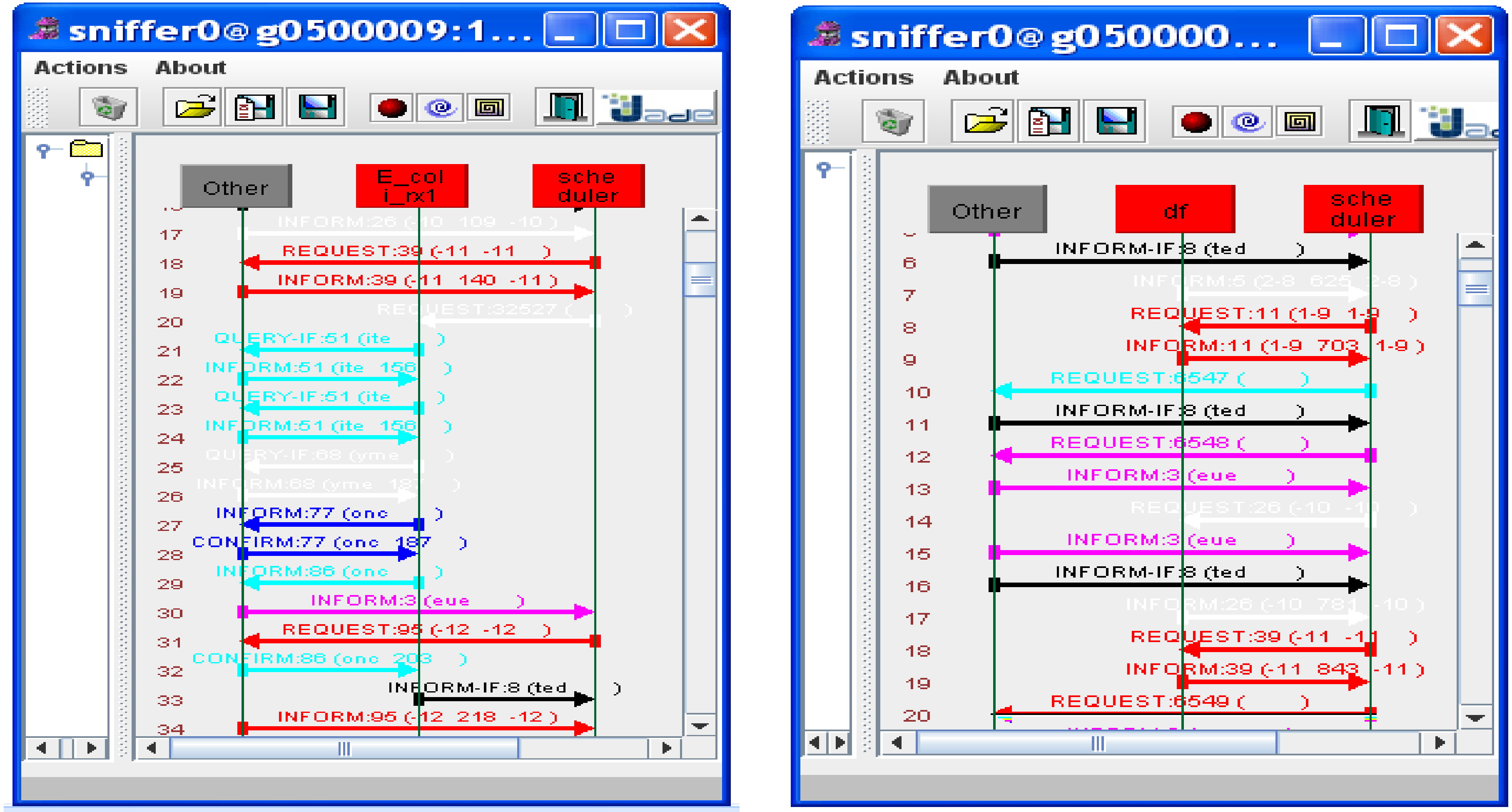
5. Similarly, the Reaction agent sends queries about enzyme activity to the Cytoplasm agent and the Cytoplasm replies to these queries.
6. Then, based on the reactant and the enzyme information, the Reaction agent calculates new concentration for the reactant and product metabolites
7. The Reaction agent sends the updated product and reactant information to the Cytoplasm agent and informs the Scheduler that it has completed its execution.
8. The Cytoplasm agent informs Scheduler agent of the new metabolites created by the Reaction agent.
9. The Scheduler agent identifies the reactions where these new metabolites participate and updates the Reaction queue with these.
10. When informed by the reaction agent that it has finished its execution, the Scheduler agent triggers the next reaction in the Reaction queue.
11. The above steps in the execution repeat until the Reaction queue is empty. Then Scheduler informs the Cytoplasm agent to print the final result and the reaction agents to print their status (whether executed or not).

Figure 3-1 and 3-2 demonstrate some of the information exchange between the agents during execution. Figure 3-3 shows the sequence of message transfer among various agents.



(a) (b)

Figure 3-1: Inter agent interactions via ACLMessage Protocol : (a) Reaction agent – CytoplasmAgent, (b) Cytoplasm agent– Scheduler agent



```
17 INFORM:26 (-10 109 -10 )
18 REQUEST:39 (-11 -11 )
19 INFORM:39 (-11 140 -11 )
20 REQUEST:32527 ( )
21 QUERY-IF:51 (ite )
22 INFORM:51 (ite 156 )
23 QUERY-IF:51 (ite )
24 INFORM:51 (ite 156 )
25 QUERY-IF:68 (yme )
26 INFORM:68 (yme 187 )
27 INFORM:77 (onc )
28 CONFIRM:77 (onc 187 )
29 INFORM:86 (onc )
30 INFORM:3 (eue )
31 REQUEST:95 (-12 -12 )
32 CONFIRM:86 (onc 203 )
33 INFORM-IF:8 (ted )
34 INFORM:95 (-12 218 -12 )
```

(a)

```
6 INFORM-IF:8 (ted )
7 INFORM:5 (2-8 625 2-8 )
8 REQUEST:11 (1-9 1-9 )
9 INFORM:11 (1-9 703 1-9 )
10 REQUEST:6547 ( )
11 INFORM-IF:8 (ted )
12 REQUEST:6548 ( )
13 INFORM:3 (eue )
14 REQUEST:26 (-10 -10 )
15 INFORM:3 (eue )
16 INFORM-IF:8 (ted )
17 INFORM:26 (-10 781 -10 )
18 REQUEST:39 (-11 -11 )
19 INFORM:39 (-11 843 -11 )
20 REQUEST:6549 ( )
```

(b)

Figure 3-2: Inter agent interactions via ACLMessage Protocol : (a) Reaction agent –Scheduler Agent, (b) DF– Scheduler agent

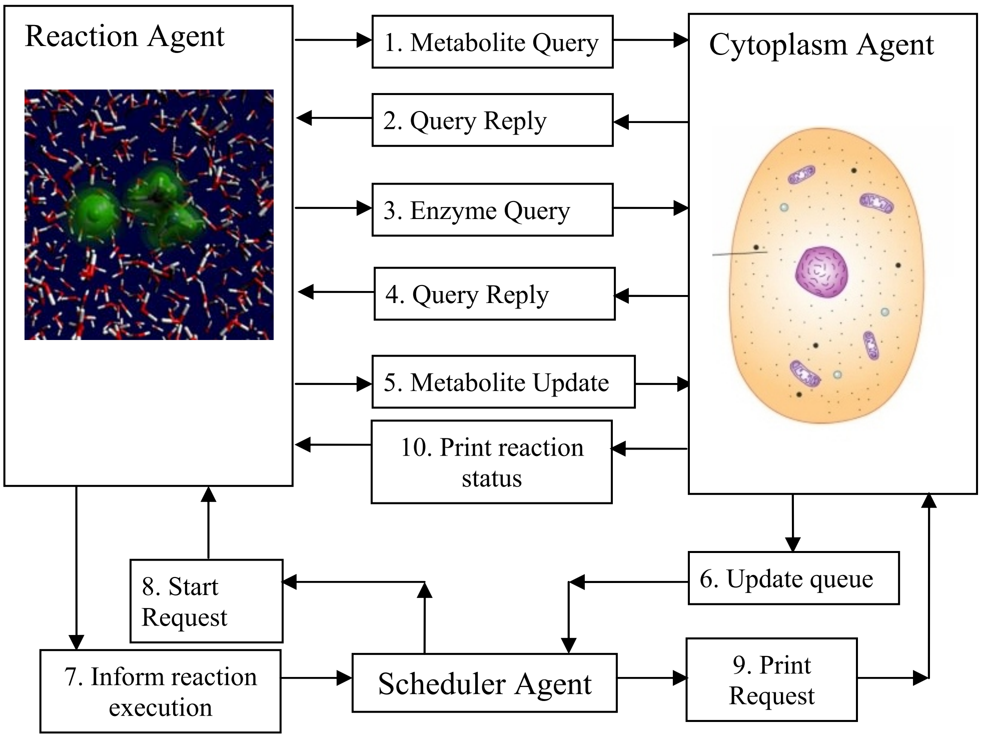


Figure 3-3: Sequence of interactions among agents using message exchange

3.2 Illustration of Agent-based Execution of Metabolic Network

Next, we illustrate with an example how the above procedure leads to the emergence of the network structure. Consider a set of 11 metabolic reactions. Details of the raw materials, products and enzymes for each of the reactions is depicted in Figure 3-4.

1. The process starts with the substrate, glucose. Since, only glucose is available at that moment the scheduler agent initializes the reaction queue only with reaction glk – the only reaction that uses GLC as reactant.

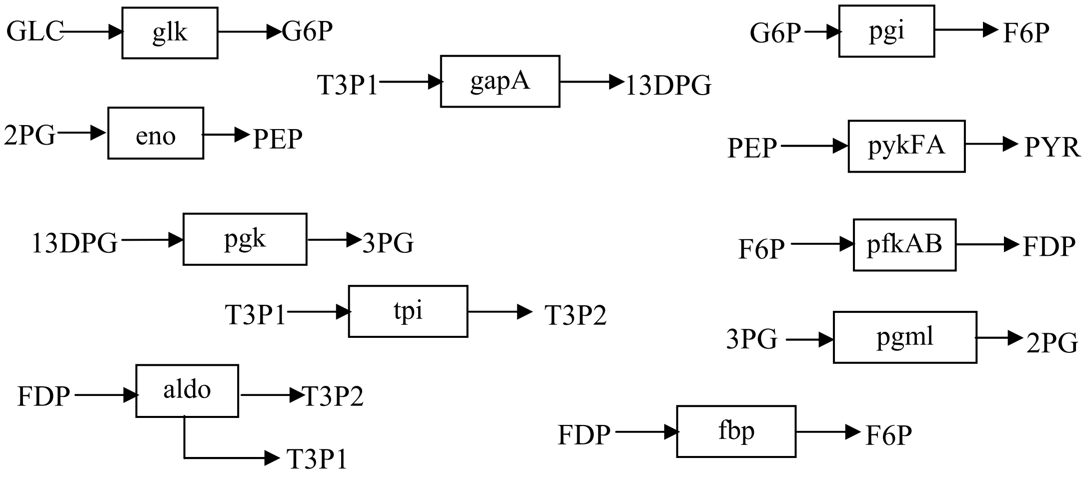


Figure 3-4: A simple metabolic network

2. The Scheduler then requests the Reaction glk agent to execute. Reaction glk occurs according to the procedure described earlier and some of GLC is converted to G6P as product of that reaction. This change is reported to the Cytoplasm, which in turn informs the Scheduler agent of the presence of the new metabolite (G6P). The Scheduler agent queries the DF and identifies reaction pgi, where G6P participates. This reaction is then inserted in the scheduler queue as shown in Figure 3-5.
3. Next, the Scheduler agent triggers the reaction agent pgi which in turn produces F6P. Again, the Cytoplasm agent informs the Scheduler agent regarding the new metabolite produced. Scheduler agent consequently updates the queue with reaction pfkAB. Figure 3-4 illustrates the status of the queue during the course of the emergence of the network structure.

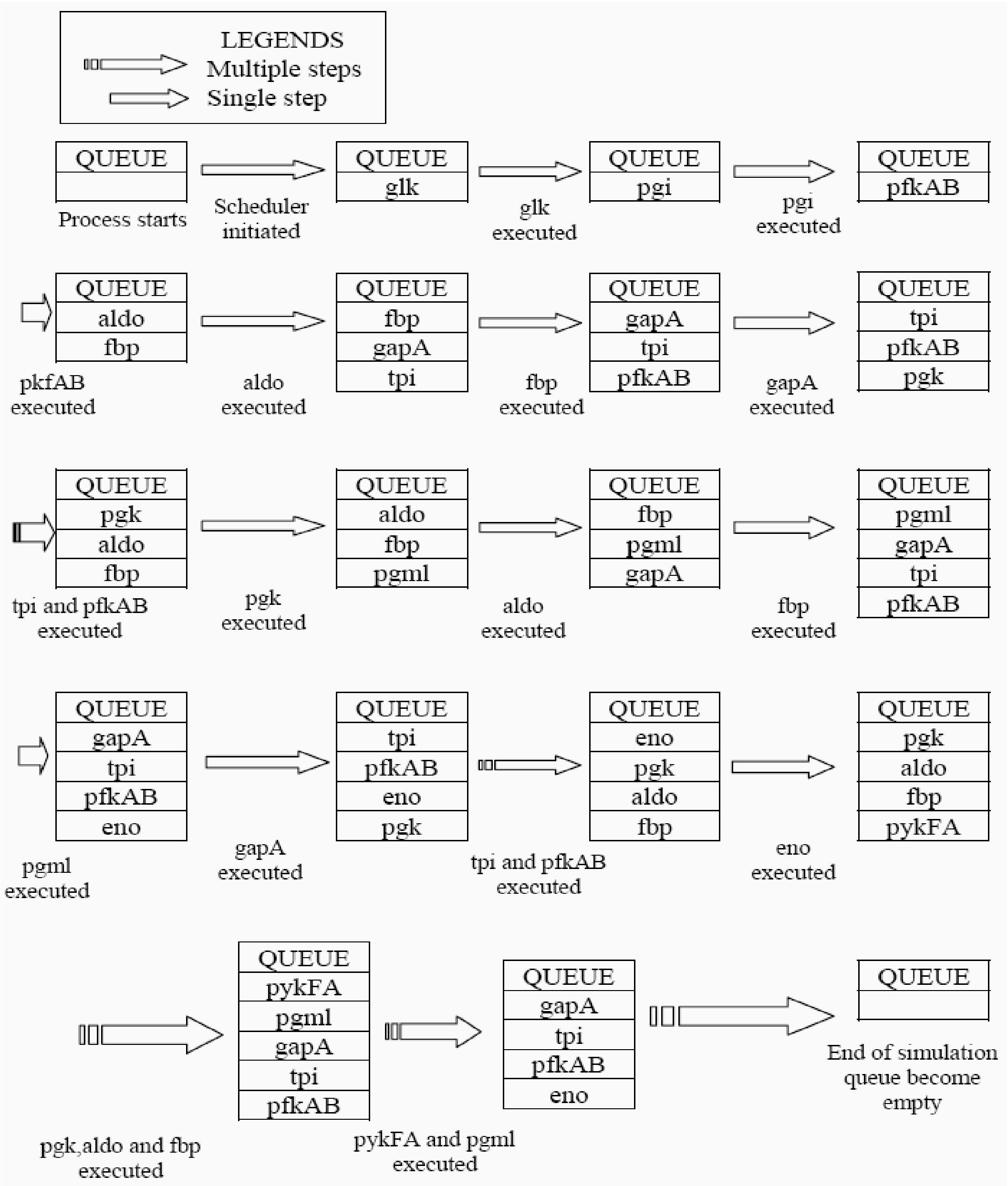


Figure 3-5: Evolution of the agent queue during the emergence of the Metabolic network

4. The interactions among different reaction agents continue thorough their inputs (reactants)-outputs (products) relationship. The execution of one reaction results in triggering other reactions by producing the necessary precursors. Finally, the network shown in Figure 3-6 emerges as a consequence of all the interactions.

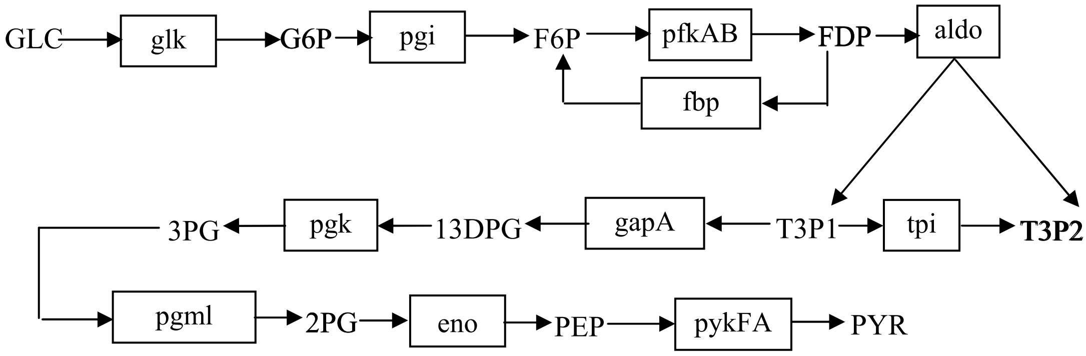


Figure 3-6: Emergent Reaction Network for Example

3.3 Application of Agent-based Model to Identifying network gaps

The agent-based model and the simulation strategy can also be directly extended to identifying gaps in the network. As discussed in the previous chapter, the recent trends in large-scale sequencing projects has resulted in the accumulation of complete genome sequence information for a number of species, and integrated pathway databases such as KEGG allow us to analyze organism-specific connectivity maps of metabolites based on the annotation of the genomes. Often times, such network model is incomplete in terms of enzyme activity or missing reactions; such inconsistencies are termed as a gap in the network. These inconsistencies may lead to the erroneous prediction of genetic interventions for targeted overproduction or the elucidation of misleading organizational principles and fallacious properties of the metabolic network. Therefore, identifying networks gaps and subsequently bridging them has become a major challenge for the modelers. It is difficult to implement automation in gap identification and filling by conventional equation-based approaches.

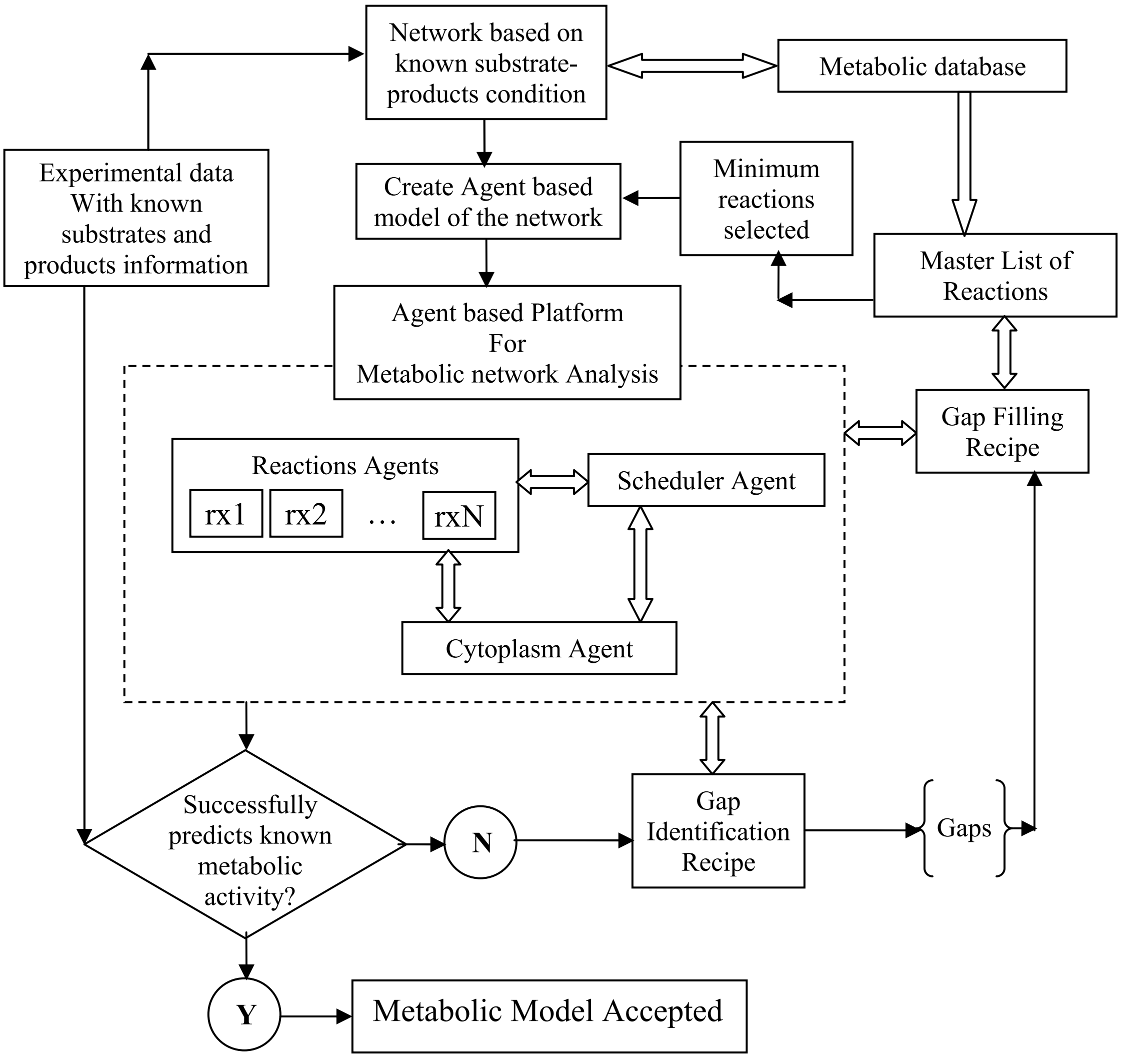

Network based on known substrate-products condition
Metabolic database
Experimental data With known substrates and products information
Create Agent based model of the network
Minimum reactions selected
Master List of Reactions
Agent based Platform For Metabolic network Analysis
Gap Filling Recipe
Reactions Agents
rx1
rx2
…
rxN
Scheduler Agent
Cytoplasm Agent
Successfully predicts known metabolic activity?
N
Gap Identification Recipe
Gaps
Y
Metabolic Model Accepted

Figure 3-7: Activities required for finding and filling the network gap

An effective gap detection and filling methodology have been implemented using the proposed agent-based model. Figure 3-7 summarizes the major steps. The basic idea for identification of inconsistencies is to simulate the network model and qualitatively predict the final product of the pathway as well as the essential intermediates. When compared with the experimental results, if the model is successful in predicting the final product and the intermediates, it means the model is consistent; otherwise, there must be some gaps in the model. Inconsistencies might arise from (1) enzyme inactivity, (2) insufficient metabolites, or (3) due to missing reactions. For a complex network with numerous interactions like many metabolic pathways, a single gap may propagate through the network and create several inconsistencies. Our goal is to find the root cause(s) for the inconsistencies and fill them to develop complete network.

Procedure

1. Collect experimental data for various cellular products for a minimal substrate condition.
2. Develop a model based on the understanding of the system, relying on literature and metabolic databases. Identify the required substrate, co-enzymes, and set of essential reactions.
3. Encode the information by creating text files for initializing Cytoplasm and Reaction agents.
4. Starting from the substrates, execute the agent-based model. This results in the final set of metabolites (initial and newly formed ones) and a list of executed and non-executed reactions.
5. Evaluate if the model is producing the same products as the experiment. If not, there are some missing elements in the model.
6. From the results, evaluate if (a) the enzyme for a reaction is not active, (b) the reactants (metabolites) for a reaction insufficient, or (c) the model is incomplete and does not include one or more essential reactions.

If gaps have been identified, putative missing elements can be generated through search. An efficient search method that exploits the agent structure is proposed, as explained in detail next.

3.3.1 Search-based Method for Identifying Gaps

Prior to explaining the detailed search method, some basic features relevant to the search algorithm are explained. The search strategy mines the information produced during simulation to identify gaps. As mentioned above, this information includes all the inconsistencies identified during simulation including unexecuted reactions and non-participating metabolites. First, we explain how the relevant data is extracted. A key requirement for network gap identification is tracking the execution status of the reaction agents. We therefore introduce a new attribute to each Reaction agent, called is_executed. When a reaction is executed during the course of qualitative simulation, its is_executed attribute is set to TRUE. Two lists are also created in the Cytoplasm, which is the repository of information, to store information regarding (un)executed reactions – unTriggeredReactionList and triggeredReactionList.; Each executed reaction is also included in the triggeredReactionList. At the end of execution, all reactions whose is_executed status is false are put into an unTriggeredReactionList. Similarly, we introduce an attribute for metabolites called "touched" to represent the participation status of the metabolites. Again, two lists are created to store metabolites based on their status, touchedMetaboliteList and unTouchedMetaboliteList. Four other lists have been created in the Cytoplasm. The purpose of the metaboliteList and subsidiaryMetaboliteList is to keep track of the potential root metabolite that leads to the gap(s). Similarly, reactionList keeps track of the search in terms of reactions. The actionList is used to store the necessary actions to bridge the gaps. Next, we report the detailed search algorithm.

Algorithm:

1. The required information for gap-search is initialized during the course of generating the network structure thorough qualitative simulation. The touchedMetaboliteList, the unTouchedMetaboliteList and the triggeredReactionList are also populated in this stage.

2. Before search begins, the desired final product of the network, is identified by querying the user. If this metabolite is in the unTouchedMetaboliteList, a network gap is flagged.

3. The metabolite is then included into the metaboliteList and the subsidiaryMetaboliteList.

4. Next, the system picks the first element from the metaboliteList and searches for the reactions producing that particular metabolite as product.

5. Then the resulting reactions are checked with the triggeredReactionList. If the triggeredReactionList does not contain the specified reaction, the reaction would be included into the reactionList. **If the reaction search results null, that indicates that the metabolite is not produced in the system and a necessary reaction is missing.**

6. In the next step, the first element of the reactionList is selected and checked for its precursor (reactant metabolite). Once the information regarding the reactant is available, their presence in the untouchedMetaboliteList and subsidiaryMetaboliteList is then checked.

7. If the reactant metabolite is non-participating and not included into the subsidiaryMetaboliteList then the metabolite is put in both the metaboliteList and subsidiaryMetaboliteList. That indicates that the product metabolite is not the root cause and the system removes it from the metaboliteList. If none of the reactant metabolite is non-participating then there might be two possible

reasons, either the enzyme is not active or the product metabolite is not produced sufficiently in the system.

8. In the next step, the system will check for the enzyme information of the corresponding reaction. If enzymes are not active then the enzyme name is included into the actionList. If enzymes are active, then it means that the product is not produced sufficiently in the system.

9. If multiple reactions are identified in step 5, the system would pick the next element and repeat Steps 6-8 above.

Thus, the search propagates backward along the pathway to identify all the root causes for the gap. Figure 3-8 describes this overall backtracking strategy for gap identification. We illustrate this next using a case study.

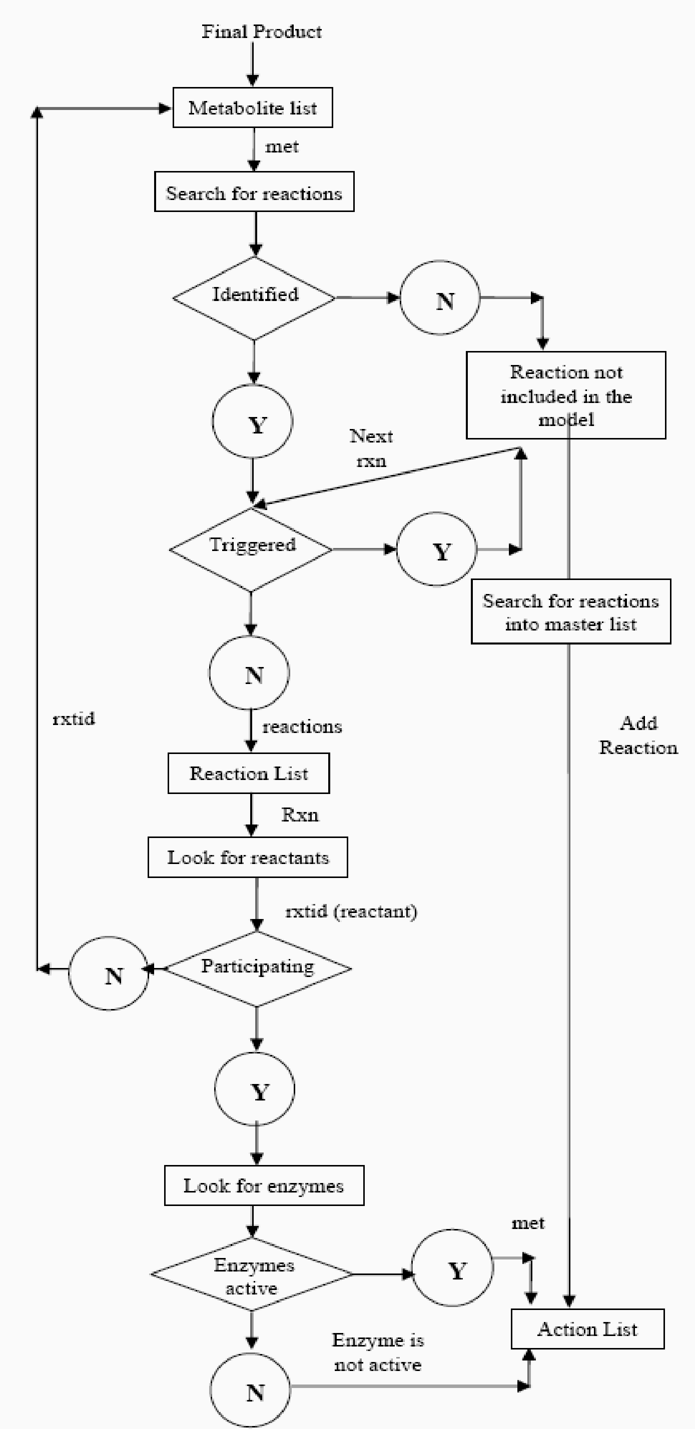


Figure 3-8: Strategy for back tracking from the desired product to find gap

3.4 Case study: Finding gap in central metabolic model of *E. coli*

For this case study, a small model of *E. coli* central metabolism with 74 reactions and 66 metabolites has been selected. Different types of gaps were created in the glycolysis pathway – originating from enzyme inactivity, a dead-end metabolite, and due to reactions. Each is described in detail next.

Example 1: Gap due to inactive enzymes in linear pathway

As a first demonstration, we choose the glycolysis pathway, where glucose is converted to final product pyruvate. If there are no inconsistencies in the network, for minimal substrate condition the model can be used to predict the final product of the pathway through qualitative simulation. In forward propagation, starting from the substrate glucose, the network structure evolves as the consequence of the interactions between the reactions agents as shown in Figure 3-9.

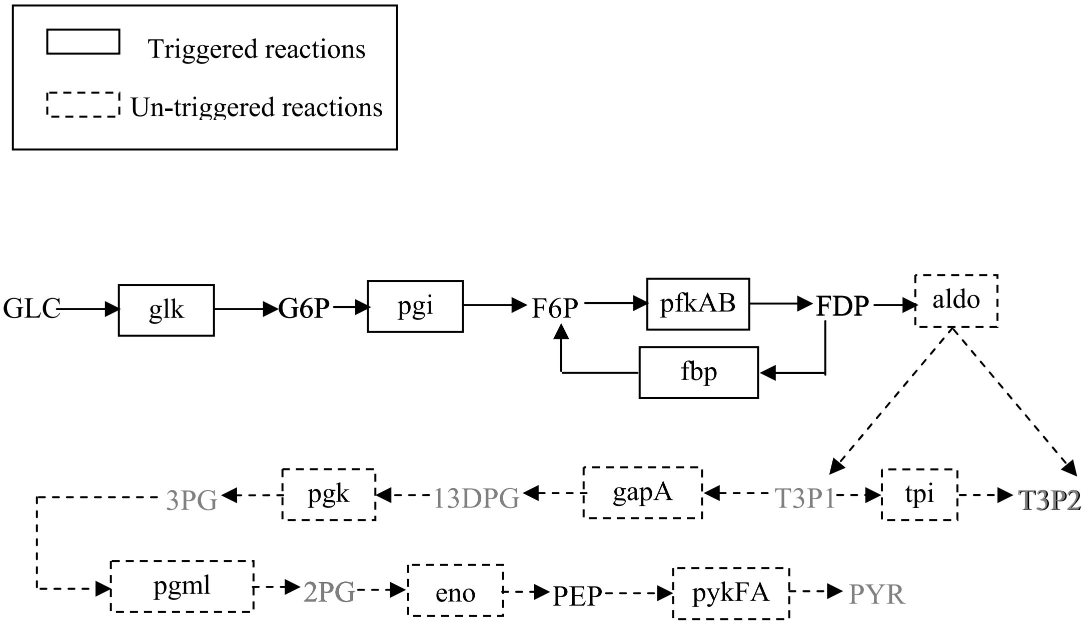


Figure 3-9: Emergent Reaction Network for Example after deactivating enzyme for aldolase reaction.

When the qualitative simulation is conducted with the enzyme for the Aldoase reaction inactive, the simulation results summarized in Table 3-4 are produced. For, the network analysis value of γ_{max} is set to 0.5. Since, the analysis works qualitatively to identify the connection among different metabolites, any positive value will ensure that all possible reactions involved a particular metabolite can occur at that very moment. Example 1 on page 51 has been repeated using two different value of γ_{max} (i.e. γ_{max} = 0.2 and γ_{max} = 0.8). In each case the results are coincide. Table 3.5 summarized the simulation result for both γ_{max} =0.2 and γ_{max} =0.8.

Table 3-4: Summary of the result for finding gap due to inactive enzyme

Triggered Reactions	Un-triggered Reactions	Nonparticipating metabolites	Final result
Glk	Aldo	13DPG	Fructose_bisphosphate_aldolase _class_II is not active check forE_coli_rx4
Pgi	Tpi	2PG	
pfkAB	gapA	3PG	
Fbp	Pgk	T3P1	
	pgml	T3P2	
	Eno	PEP	
	pykFA	PYR	

Table 3-5: Simulation results for γ_{max} =0.2 and γ_{max} =0.8

γ_{max} =0.2		γ_{max} =0.8	
Triggered Reactions	Final result	Triggered Reactions	Final result
Glk	Fructose_bisphosphate_aldolase_class_II is not active check forE_coli_rx4	glk	Fructose_bisphosphate_aldolase_class_II is not active check forE_coli_rx4
Pgi		pgi	
pfkAB		pfkAB	
Fbp		fbp	

For finding this gap, the automated search begins with the final product pyruvate (PYR). When the DF is queried for reactions producing pyruvate, only one reaction – pykFA is identified. The algorithm will then check whether the enzyme for the reaction is active. If enzyme is not active then it will put the enzyme name in to action list. It then moves on to the next step to identify the reactants (in this case PEP) for the reaction and check whether they are participating or not. If the reactants are nonparticipating, then system will store them into a subsidiaryMetabolite list and remove pyruvate from that list. Then PEP is picked and the search repeated for the reactions producing PEP. Reaction eno (in Figure 3-8) is the only candidate available. The algorithm then puts eno into the reactionlist and searches for its reactants. Thus the search continues following the same strategy and identifies T3P1 is not produced in the system. T3P1 is a product from reaction aldolase (aldo) and FDP is the precursor for that reaction. Since FDP is participating metabolite, so the aldo reaction is shortlisted. Finally, the inconsistency is identified to be the enzyme Fructose bisphosphate aldolase class II being inactive. Figure 3-10 summarizes the steps as the search continues towards gap identification.

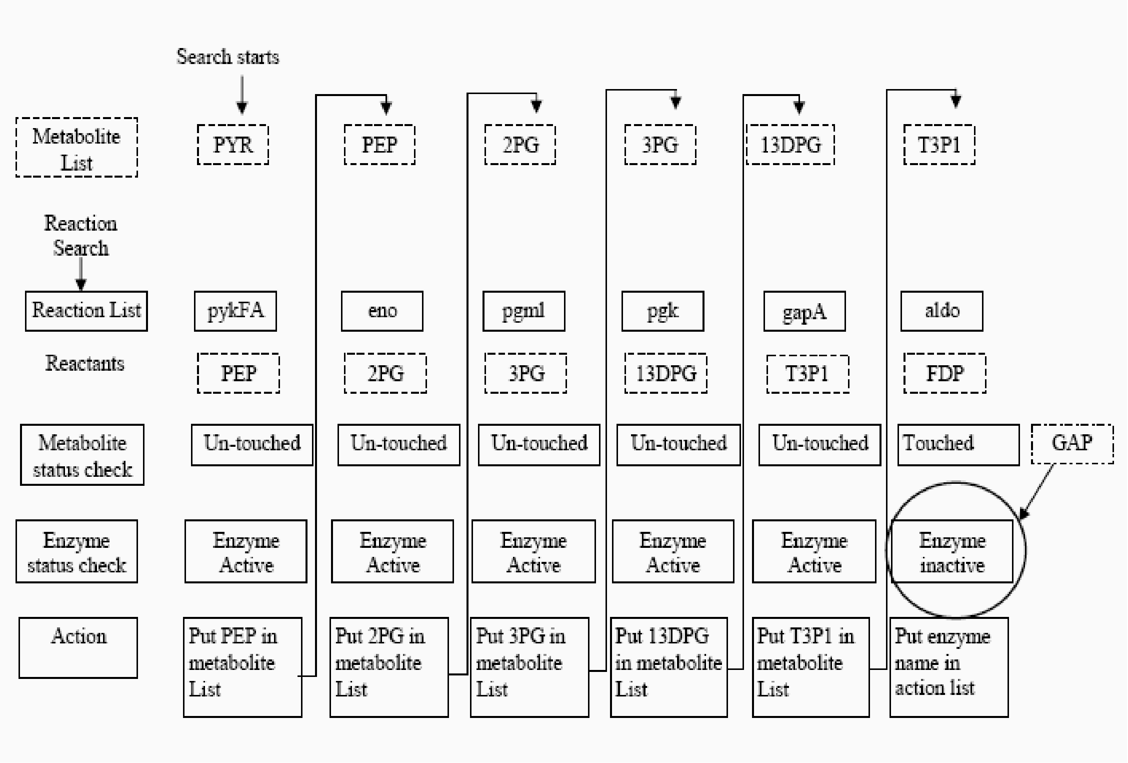


Figure 3-10: Summary of system status during gap identification in example 1

Example 2: Gap due to inactive enzymes in Branched pathway

To demonstrate the performance of the strategy for successful identification of gap in a connected pathway, the Phosphate Pentose Pathway (PPP) is considered along with the glycolysis pathway, where glucose is converted to final product pyruvate. Figure 3-11 shows the branched network consisting of glycolysis and PPP pathways. This time two different enzymes are made inactive. The system successfully identified the specified gaps as summarized in Table 3-6.

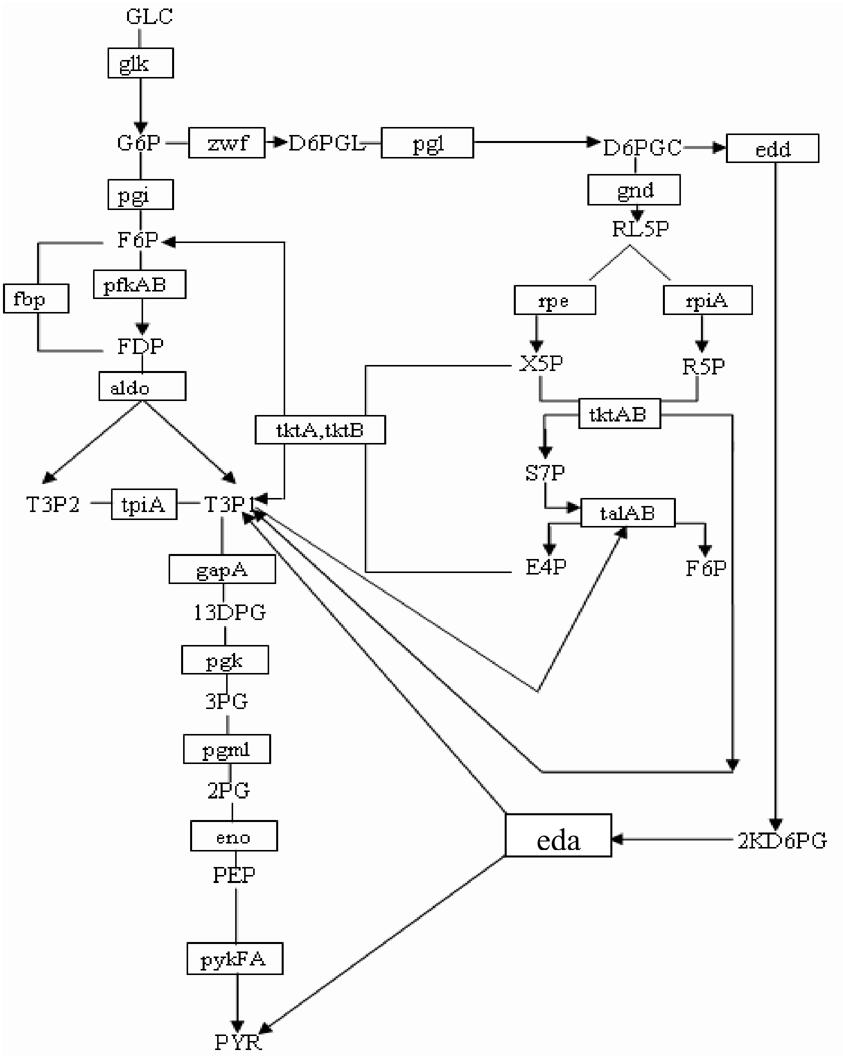


Figure 3-11: Metabolic network consisting of glycolysis and PPP pathways

Table 3-6: Summary of the result for finding gaps in branched network

Triggered Reactions	Un-triggered Reactions	Nonparticipating metabolites	Final result
Glk	pgk	2PG	Phosphogluconate_dehydratase is not active Check for E_coli_rx7 Phosphoglycerate_kinase is not active Check for E_coli_rx19
Pgi	pgml	3PG	
pfkAB	eno	T3P1	
Fbp	pykFA	T3P2	
tpiA	edd	PEP	
gapA	eda	PYR	
Zwf		2KD6PG	
Pgl			
Gnd			
rpiA			
Rpe			
tktAB			
talAB			
tktA, tktB			

Example 3: Gap due to Missing reaction

To illustrate the performance of the proposed framework in identifying gaps due to missing reactions, Reaction rpiA is intentionally removed from the network shown in Figure 3-11. As shown in the Figure 3-12, metabolite ribose 5-phosphate (R5P) is produced only by reaction rpiA. If the network does not include this reaction, then R5P cannot be produced; as a result, reaction tktAB cannot proceed and hence S7P is not produced. Since S7P is a precursor, reaction talAB cannot execute due to lack of reactant. As a result there is no production of E4P in the system, which in turns prevents the execution of reactions tktA, tktB and so on. Figure 3-12 summarizes the effect of the missing reactions and Table 3-7 summarizes the results.

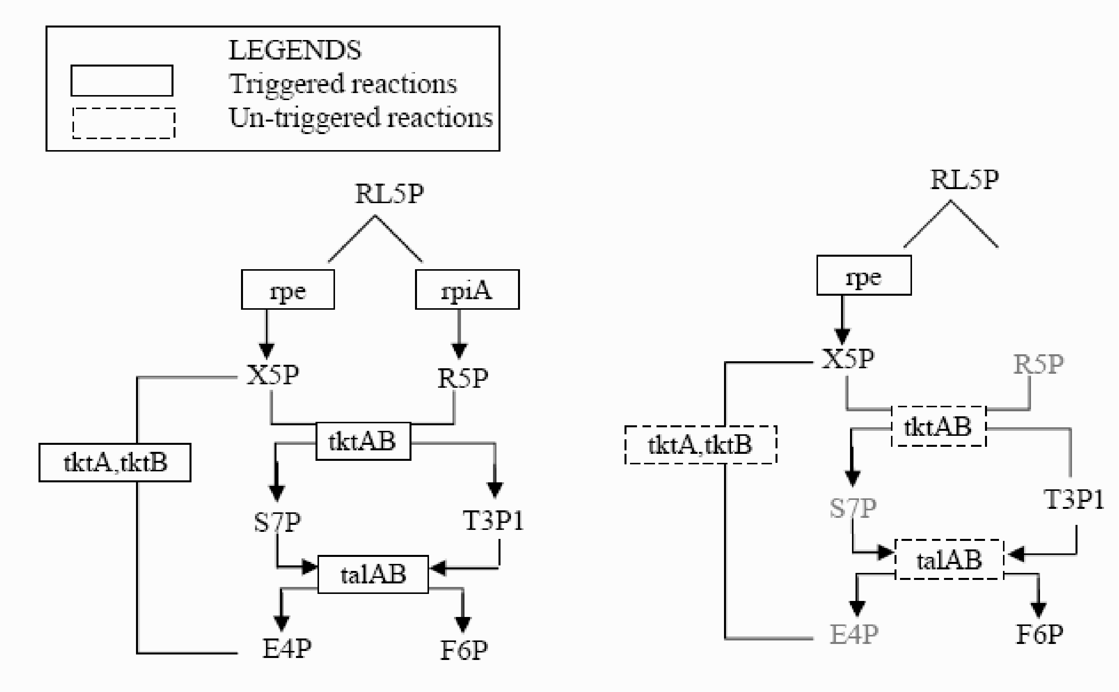


Figure 3-12: Effect of missing reaction rpiA

Table 3-7: Summarized result for identifying gaps due to missing reaction

Triggered Reactions	Un-triggered Reactions	Nonparticipating metabolites	Final result
Glk	tktAB	R5P	R5P not produced Search databases for reaction
Pgi	talAB	S7P	
pfkAB	tktA, tktB	E4P	
Fbp			
tpiA			
gapA			
Pgk			
Pgml			
Eno			
pykFA			
Zwf			
Pgl			
Gnd			
Edd			
Rpe			
Eda			
Eda			

Here, a simple case involving a single missing reaction has been demonstrated using the strategy described above. The same backtracking strategy can also be extended in the future for the case of multiple missing reactions. Once gaps have been identified, they need to be filled. A suitable strategy is described next.

3.5 Strategy for Filling Gaps using the Agent-based Model

For filling gaps, a master list of reactions that contains all reactions possible in the organism is required. That can be developed from metabolic databases and literature.

First, the proposed strategy for filling gaps is illustrated using the example metabolic network shown in Figure 3-13. Three reactions are deemed to be missing in the original model and needed to be identified from a master list containing these and many other (irrelevant) reactions.

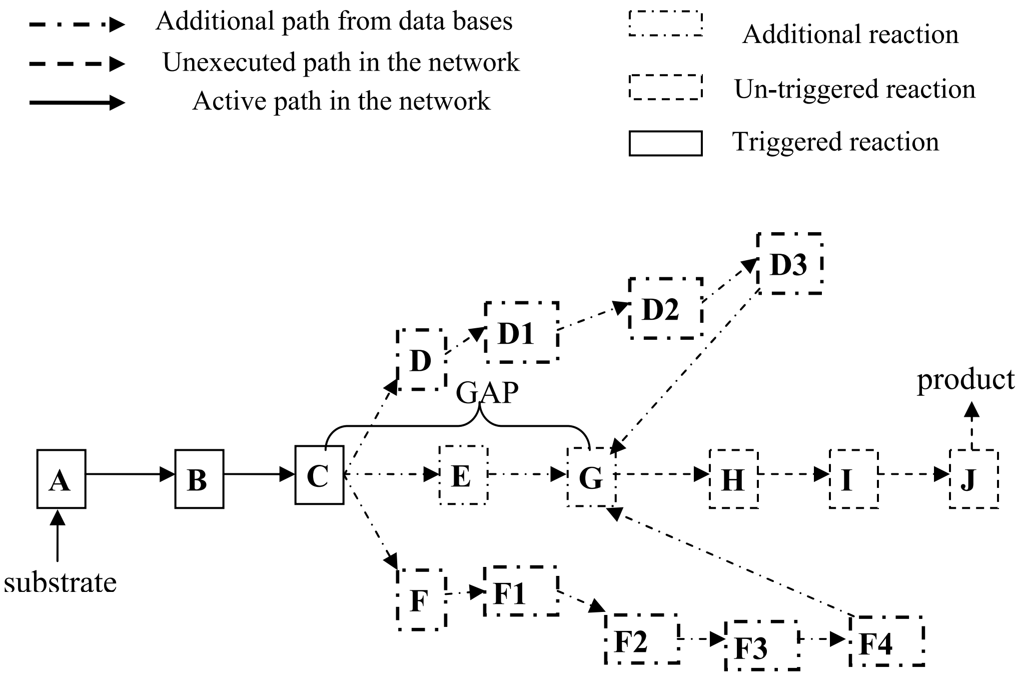


Figure 3-13: Illustration of gap due to the missing reactions

In Figure 3-13, the final product is the output of reaction J. To fill the gap reaction G must be triggered. Thus, the inputs for reaction G are required. Gap has been identified after reaction C. Our objective is to find the shortest possible path from reaction C to reaction G.

Filling the metabolic gap involves search for the proper reaction sequence to produce the desired product. Implementation of the search algorithm needs maximum allowable gap size to limit the search. Let the maximum allowable gap size for the above example be 3. This small value of allowable gap size has been selected to facilitate the search for the shortest possible path to fill that gap. Usually, biochemical reaction network is highly interacting system and the targeted metabolite may be produced from different pathways. Selection of a small number (as 3 in the example) will ensure the quick finding of the shortest possible reaction path to get the desired product from the substrate. If the selected gap size is too small to find a possible path, the gap size can be increased and the search repeated.

1. After finding the gap in the network using the strategy described in Section 3.4, we need to search the master list for all possible reaction sequences that would be able to complete the gap. In the above example, three possible paths from reaction C to reaction G can be identified from the master list: E, F and D. We next apply breadth-first search to find the shortest possible sequence.
2. Once a complete sequence of reactions have been found within the specified gap size, these missing reactions can be examined by the user and included in to the model.

The breadth-first algorithm is explained in detail next.

Breadth-first search

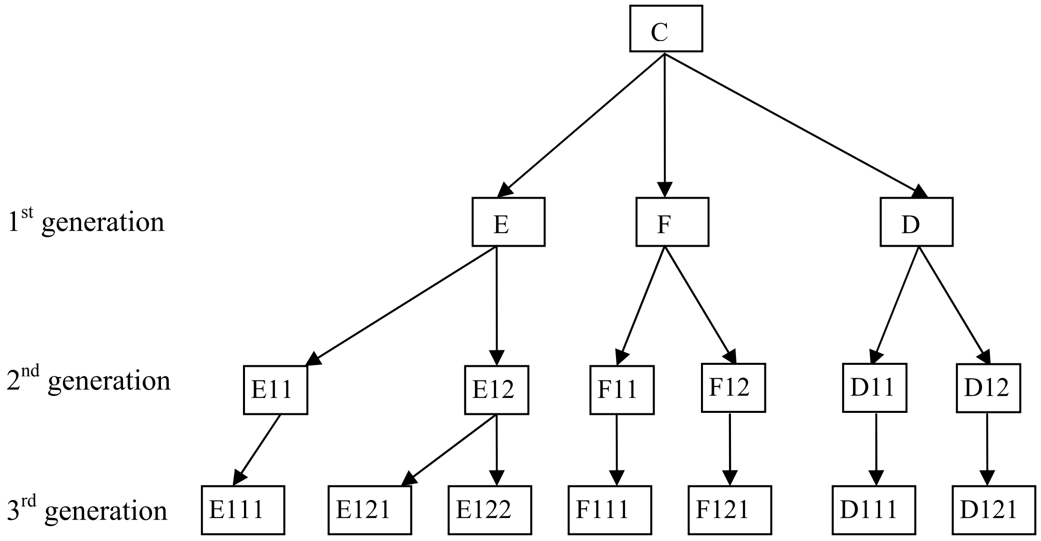


Figure 3-14 : Breadth-first search tree

The main concept of breadth-first search is to explore the root node first, then all the nodes generated by the root node then their successors, and so on (Russell & Norvig, 1995). In general, all the nodes in generation **n** in the search tree should be explored before the nodes of generation **n+1**.

1. Reaction C is the root node for the search. The algorithm takes the product as input and searches the master reaction list for reactions using the product of reaction C as inputs (reactants).
2. The resulting reactions form the next layer (1^{st} generation in the search tree) of the search tree. In the example above, D, E and F are the 1^{st} generation nodes.
3. Next these child reactions are checked to see if any of them is the same as any of the existing reactions in the network (for example G). If so, the search stops, otherwise it continues to the next generation.

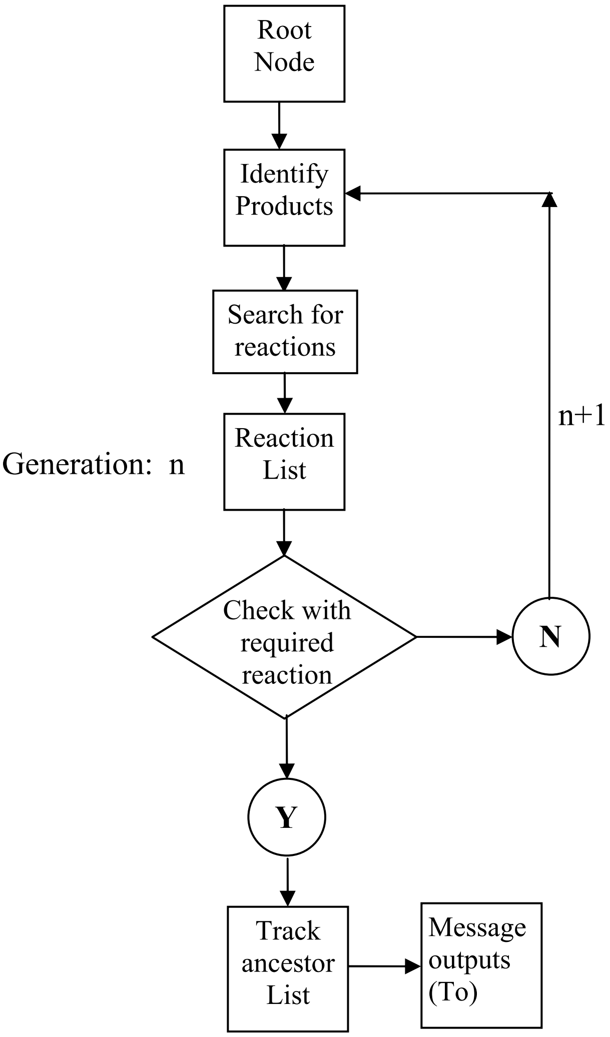


Figure 3-15: Steps involved in the breadth-first search

4. Next, all the nodes at the 1st generation are explored in a similar fashion as the root node, by repeating steps 1 and 2. This results in the 2nd generation nodes. In the Figure 3-14, the 2nd generation reactions are E11, E12, F11, F12, D11, and D12. Step 3 is repeated for all these candidates.

5. Thus, the search will continue until the the required reaction is found or the maximum allowable gap size is reached. For the example problem, the required reaction (reaction G) is found in the 2nd generation nodes.
6. Next, by tracing the ancestors from G, the shortest possible path to fill the gap in the network is identified as C-E-G.

We illustrate the proposed algorithm using an example.

Example of filling a gap in the metabolic network

The Glycolysis pathway, where final product pyruvate is produced from the substrate, glucose, is considered as an example. To illustrate the above gap filling strategy, reactions pgi, pfkAB and aldo are not included in the metabolic model. As shown in Figure 3-16, T3P1 cannot be produced from G6P due to this gap which in turn stops the production of PYR (pyruvate). Following the above strategy with a Master List containing a total of 11 reactions, ROUTE 1 and ROUTE 2 in Figure 3-16, are identified as the two possible routes to produce T3P1. Table 3-8 summarizes the results obtained for the gap filling scheme.

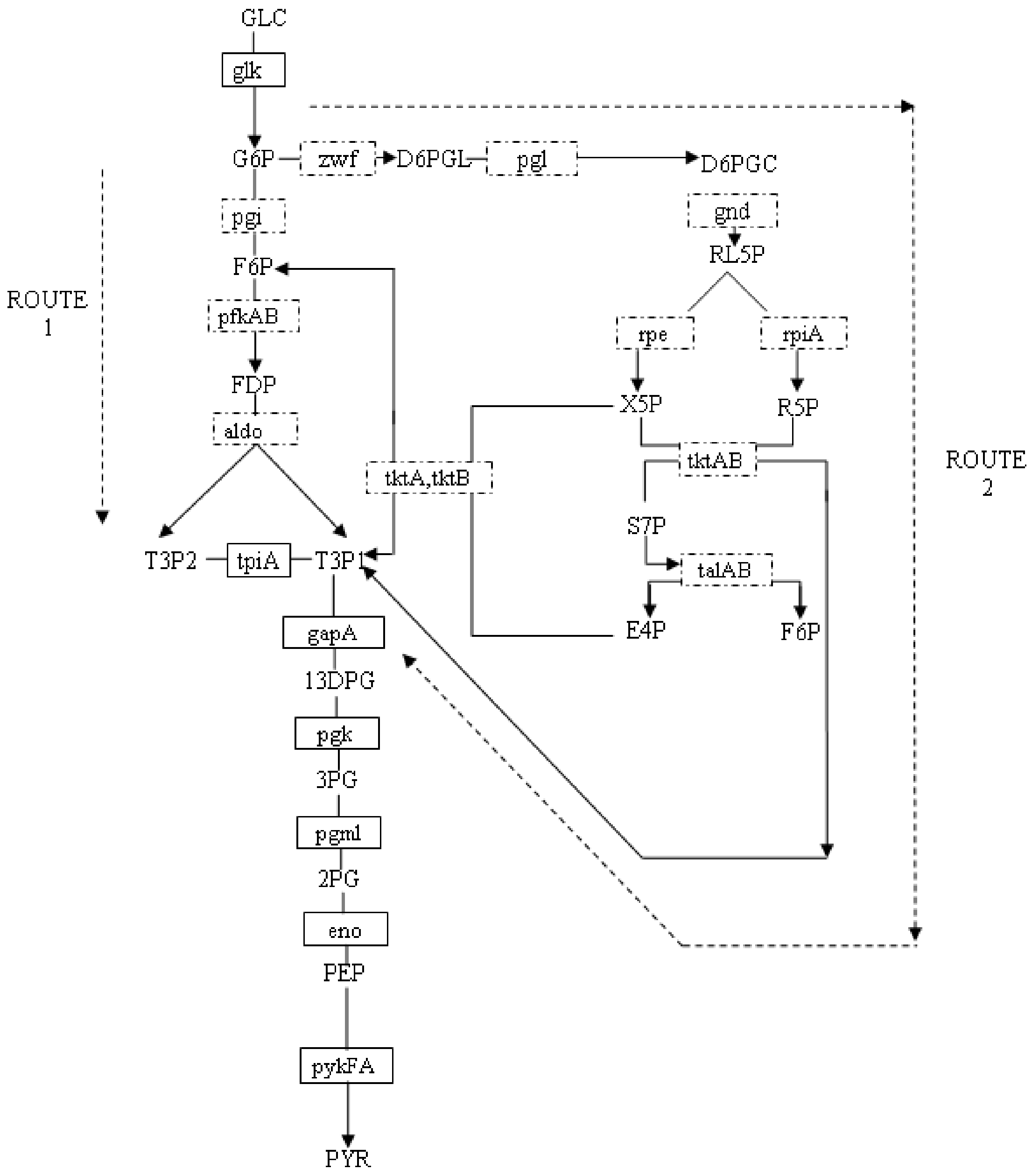


Figure 3-16: Alternative routes for the production of T3P1

Table 3-8: Result for identifying and filling gaps with missing reactions

Target Node	Starting Node	1st generation	2nd generation	3rd generation	Final result
T3P1	G6P	F6P	FDP	T3P2	T3P1 is not produced Check for additional path from Master Reaction List Add following reactions to the model: E_coli_rx3 E_coli_rx2 E_coli_rx1
				T3P1	
		D6PGL	D6PGC	RL5P	

3.6 Concluding remarks

Metabolic reconstruction of microbial as well as of eukaryotic organisms using bioinformatics based techniques with experimental evidence is one of the major elements of metabolic engineering. Techniques that enable researchers to ensure complete and comprehensive model of organisms are necessary. In this chapter, an agent-based modeling approach has been proposed to analyze metabolic pathways. This framework consists of three major classes of agents – Cytoplasm agent, Reaction agent, and Scheduler agent. The basic structure of the agent based model is described along with the individual agent functionalities and inter-agent interactions. The key benefit of this agent-based model is that it enables one to identify inconsistencies in metabolic network through qualitative simulation. Efficient methods for gap identification and gap filling have been proposed and demonstrated using various case studies.

Chapter 4 Dynamic Simulation of E. coli central metabolism using ABS

In previous chapter, the detailed architecture of the new agent based framework has been described and the performance of this proposed framework has been demonstrated in terms of its ability to identify different kind of gaps in any metabolic network model. The work presented in this chapter will show the applicability of the same agent based framework to emulate the kinetic behavior of cellular metabolome.

The metabolic system selected for the simulation is the central carbon metabolism of *E. coli*. Section 4.1 includes a brief description of the carbon metabolism in *E. coli*. Section 4.2 provides the details of the kinetic model that includes mainly the Glycolysis and PPP. Some modification is implemented into the basic architechture to capture the dynamic properties of the system. The modified architecture is described with necessary illustration in section 4.3. Simulation results are discussed and illustrated in section 4.4.

4.1 Central Metabolism of E. coli

E. coli is the most studied prokaryotic model microorganism. Because of its long history of laboratory culture and ease of manipulation, *E. coli* also plays an important role in modern biological engineering and industrial microbiology. In the case of *E. coli*, metabolic engineering studies have primarily focused on the central metabolic pathways that are active during growth on glucose as the sole carbon source: the Embden–Meyerhof (EM) pathways, the pentose phosphate pathway (PPP) and the tricarboxylic acid cycle (TCA). These pathways are the main components of the central metabolism. Figure 4-1 briefly describes the fundamental activities of the central metabolism in *E. coli.*

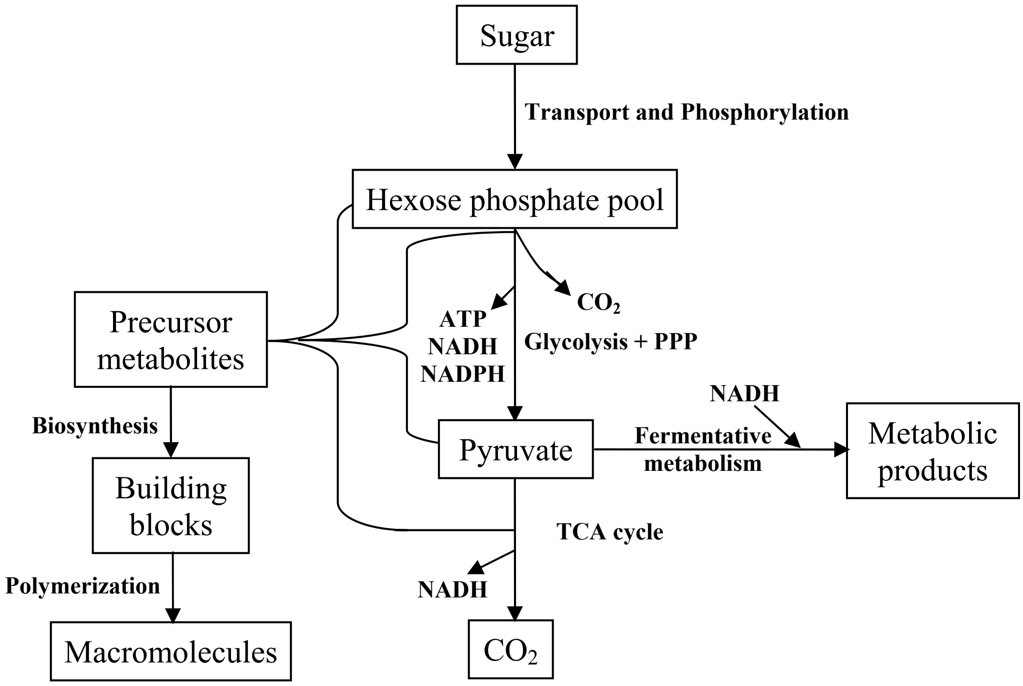


Figure 4-1: A brief representation of activities encompassed in central metabolism

The substrate that supplies carbon skeletons for the biosynthesis of the building blocks is usually called the carbon source. The most frequently used substrates in industrial processes are different kinds of sugars like glucose, sucrose, fructose, galactose, and lactose. The sugar is transported into the cell where it is first phosphorylated and then enters the hexose monophosphate pool consisting of hexose monophosphates, glucose 1-phosphate (G1P), glucose 6-phosphate (G6P) and fructose 6-phosphate (F6P). Phosphorylation may occur independently or in concomitance with the transport process. The hexose monophosphates undergo glycolytic reactions, the end product of which is pyruvate. Pyruvate in turn, may be further converted by different routes depending on the energetic state of the cells. Under oxygen enriched condition, most of the pyruvate enters the TCA cycle where it is oxidized completely to carbon dioxide and water in oxidative phosphorylation. However, under oxygen limiting condition or in anaerobic organisms, pyruvate may be converted into metabolic products like lactic acid, acetic acid, and ethanol via fermentative pathways. Some of the intermediates in glycolysis and the TCA cycle serve as precursor metabolites for the biosynthesis of building blocks like various amino acids, and polysaccharides. These building blocks are polymerized into macromolecules, which are finally assembled into different cellular structures.

Central metabolic pathways are the source of precursor compounds for all other pathways, and also the significant energy source and reducing power for cellular processes. The active pathways and the production of essential metabolite in central carbon metabolism are critical components of a multidimensional physiological representation of the organism, since this central backbone of metabolism provides energy, cofactor regeneration, and building blocks for biomass synthesis and controls the extent and nature of by-product excretion. It serves three important purposes: firstly, it generates Gibbs free energy, in the form of ATP, that help to fuel other cellular reactions; secondly, it produces reducing agent in the form of cofactor like NADPH, required in biosynthetic reactions; and thirdly, it produces precursor metabolites required in the biosynthesis of building blocks. It is therefore no surprise that central carbon metabolism has been and continues to be the primary target of basic studies and metabolic engineering efforts. Metabolic engineering of the central carbon metabolism is required to improve the productivity and yield of native compounds, in addition to producing novel products not expressed in the natural hosts.

4.2 Case study: Dynamic model of Glycolysis and PPP in *E. coli*

The dynamic model of carbon metabolism described by Chassagnole et al., (2002) is used here to validate the proposed agent based dynamic simulation of cellular metabolism. The dynamic responses of intracellular metabolites to a pulse of glucose were experimentally measured for *E. coli* K-12 strain W3110 culture.

The model includes glucose transport system, glycolysis, biosynthetic and anaplerotic reactions and pentose-phosphate pathway consisting 30 metabolic reactions. The stoichiometry of these reactions was taken from the EcoCyc database, and from Neidhardt et al., (1996). The structural model of central carbon metabolism is shown in Figure 4-2. Table 4-1 provides the kinetic description of different enzymatic reactions and the kinetic expressions for these reactions are presented in Table 4-2. These kinetics are based on various published literature resulting from in vitro investigations with purified enzymes. The kinetic expressions used for the dynamic simulation are same as published in the work of Chassagnole et al., (2002).

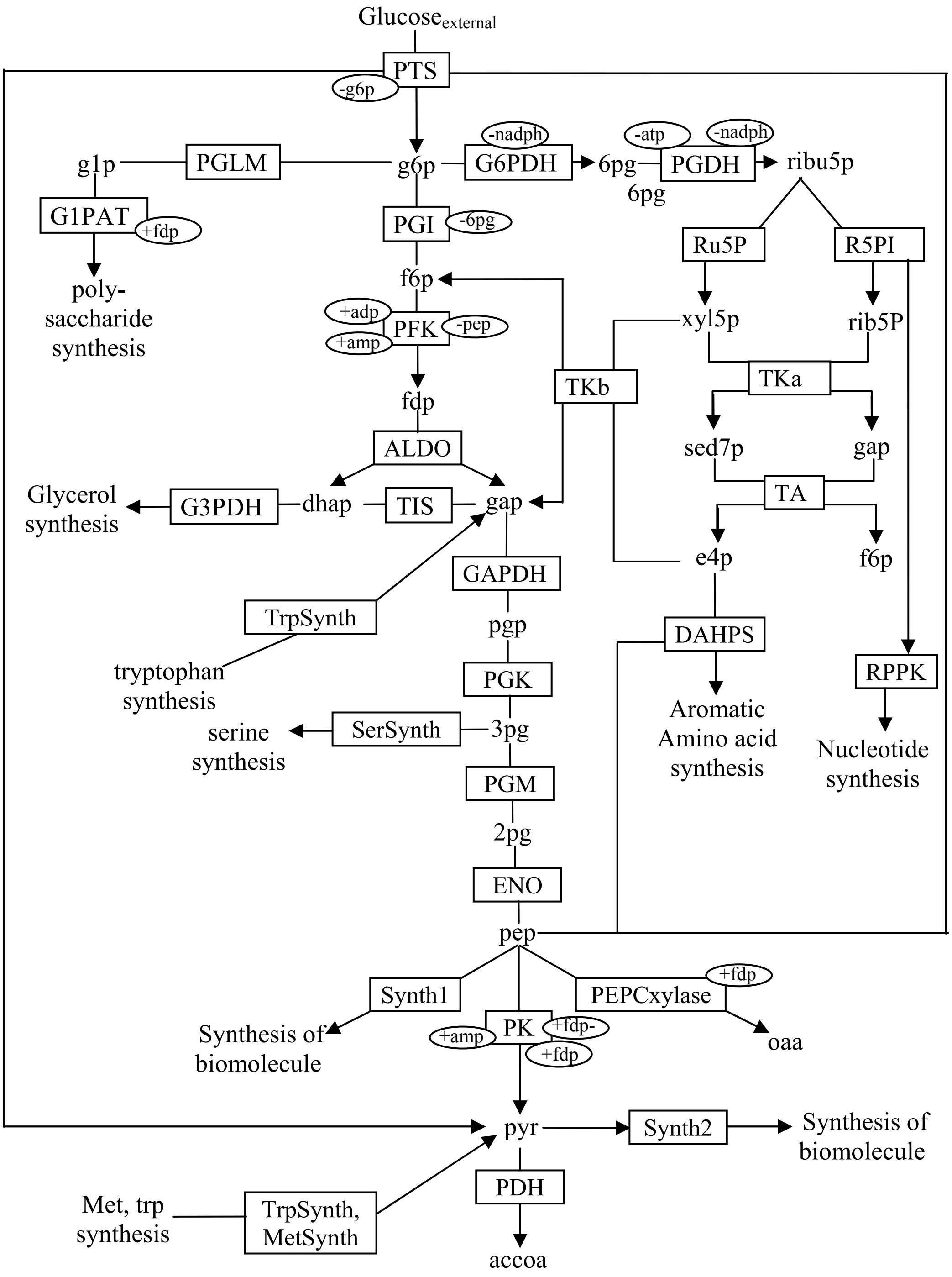


Figure 4-2: Structural model of Glycolysis and pentose phosphate pathways.

Table 4-1: Kinetic description of different enzymatic reactions

Reactions	Kinetic Description	Kinetic type
PTS	Transferase	A
PGI	Reversible Michaelis-Menten	B
PFK	Allosteric	C
ALDO	Ordered uni-bi mechanism	D
TIS	Reversible Michaelis-Menten	E
GAPDH	Two substrate reversible Michaelis-Menten	F
PGK		F
PGM	Reversible Michaelis-Menten	E
ENO		E
PK	Allosteric	G
PDH	Hill equation	H
PEPCxylase	Two substrate equation with allosteric	I
PGLM	Reversible Michaelis-Menten	E
G1PAT	Allosteric	J
RPPK	Michaelis-Menten	H
G3PDH		H
Serine Synthesis SERS		H
Synthesis 1		H
Synthesis 2		H
DAHPS	Hill equation	K
G6PDH	Two substrate irreversible Michaelis-Menten	L
PGDH		M
RU5P	Reversible mass action Kinetics	N
R5PI		N
TKA		O
TKB		O
TA		O
Mureine Synthesis MURS	Constant level corresponding to the steady state condition	P
Tryptophan synthesis TRPS		P
Methionine synthesis METS		P

Table 4-2: Kinetic rate expressions

PTS:

Error! Objects cannot be created from editing field codes.

PGI:

$$r_{PGI} = \frac{r_{PGI}^{\max}\left(C_{g6p} - \frac{C_{f6p}}{K_{PGI,eq}}\right)}{K_{PGI,g6p}\left(1 + \frac{C_{f6p}}{K_{PGI,f6p}\left(1 + \frac{C_{6pg}}{K_{PGI,f6p,6pginh}}\right)} + \frac{C_{6pg}}{K_{PGI,g6p,6pginh}}\right) + C_{g6p}}$$

PFK:

$$r_{PFK} = \frac{r_{PFK}^{\max} C_{atp} C_{f6p}}{\left(C_{atp} + K_{PFK,atp,s}\left(1 + \frac{C_{adp}}{K_{PFK,adp,c}}\right)\right)\left(C_{f6p} + K_{PFK,f6p,s}\frac{A}{B}\right)\left(1 + \frac{L_{PFK}}{\left(1 + C_{f6p}\frac{B}{K_{PFK,f6p,s}A}\right)^{nPFK}}\right)}$$

$$A = 1 + \frac{C_{pep}}{K_{PFK,pep}} + \frac{C_{adp}}{K_{PFK,adp,b}} + \frac{C_{amp}}{K_{PFK,amp,b}}$$

$$B = 1 + \frac{C_{adp}}{K_{PFK,adp,a}} + \frac{C_{amp}}{K_{PFK,amp,a}}$$

ALDO:

$$r_{ALDO} = \frac{r_{ALDO}^{\max}\left(C_{fdp} - \frac{C_{gap}C_{dhap}}{K_{ALDO,eq}}\right)}{K_{ALDO,fdp} + C_{fdp} + \frac{K_{ALDO,gap}C_{dhap}}{K_{ALDO,eq}V_{ALDO,blf}} + \frac{K_{ALDO,dhap}C_{gap}}{K_{ALDO,eq}V_{ALDO,blf}} + \frac{C_{fdp}C_{gap}}{K_{ALDO,gap,inh}} + \frac{C_{gap}C_{dhap}}{K_{ALDO,eq}V_{ALDO,blf}}}$$

TIS: $$r_{TIS} = r_{TIS}^{\max} \frac{(C_{dhap} - \frac{C_{gap}}{K_{TIS,eq}})}{K_{TIS,dhap}(1 + \frac{C_{gap}}{K_{TIS,gap}}) + C_{dhap}}$$

GAPDH: $$r_{GAPDH} = \frac{r_{GAPDH}^{\max}(C_{gap}C_{nad} - \frac{C_{pgp}C_{nadh}}{K_{GAPDH,eq}})}{(K_{GAPDH,gap}(1 + \frac{C_{pgp}}{K_{GAPDH,pgp}}) + C_{gap})(K_{GAPDH,nad}(1 + \frac{C_{nadh}}{K_{GAPDH,nadh}}) + C_{nad})}$$

PGK: $$r_{PGK} = \frac{r_{PGK}^{\max}(C_{adp}C_{pgp} - \frac{C_{atp}C_{3pg}}{K_{PGK,eq}})}{(K_{PGK,adp}(1 + \frac{C_{atp}}{K_{PGK,atp}}) + C_{adp})(K_{PGK,pgp}(1 + \frac{C_{3pg}}{K_{PGK,3pg}}) + C_{pgp})}$$

PGM: $$r_{PGM} = r_{PGM}^{\max} \frac{(C_{3pg} - \frac{C_{2pg}}{K_{PGM,eq}})}{K_{PGM,3pg}(1 + \frac{C_{2pg}}{K_{PGM,2pg}}) + C_{3pg}}$$

ENO: $$r_{ENO} = r_{ENO}^{\max} \frac{(C_{2pg} - \frac{C_{pep}}{K_{ENO,eq}})}{K_{ENO,2pg}(1 + \frac{C_{pep}}{K_{ENO,pep}}) + C_{2pg}}$$

PK: $$r_{PK} = \frac{r_{PK}^{\max} C_{pep} \left(\frac{C_{pep}}{K_{PK,pep}} + 1\right)^{(nPK-1)} C_{adp}}{K_{PK,pep} \left(L_{PK} \left(\frac{1 + \frac{C_{atp}}{K_{PK,atp}}}{\frac{C_{fdp}}{K_{PK,fdp}} + \frac{C_{amp}}{K_{PK,amp}} + 1} \right)^{nPK} + \left(\frac{C_{pep}}{K_{PK,pep}} + 1 \right)^{nPK} \right) (C_{adp} + K_{PK,adp})}$$

PDH: $$r_{PDH} = r_{PDH}^{\max} \frac{C_{pyr}^{nPDH}}{K_{PDH,pyr} + C_{pyr}^{nPDH}}$$

PEPCxylase: $$r_{PEPCxylase} = r_{PEPCxylase}^{\max} \frac{C_{pep} \left(1 + \left(\frac{C_{pep}}{K_{ENO,eq}} \right)^{nPEPCxylase,fdp} \right)}{K_{PEPCxylase,pep} + C_{pep}}$$

PGLM: $$r_{PGLM} = r_{PGLM}^{\max} \frac{\left(C_{g6p} - \frac{C_{g1p}}{K_{PGLM,eq}}\right)}{K_{PGLM,g6p} \left(1 + \frac{C_{g1p}}{K_{PGLM,g1p}}\right) + C_{g6p}}$$

G1PAT: $$r_{G1PAT} = \frac{r_{G1PAT}^{\max} C_{g1p} C_{atp} \left(1 + \left(\frac{C_{pep}}{K_{ENO,eq}} \right)^{nG1PAT,fdp} \right)}{(K_{G1PAT,g1p} + C_{g1p})(K_{G1PAT,atp} + C_{atp})}$$

RPPK: $r_{RPPK} = r_{RPPK}^{\max} \dfrac{C_{rib5p}}{K_{RPPK,rib5p} + C_{rib5p}}$

G3PDH: $r_{G3PDH} = r_{G3PDH}^{\max} \dfrac{C_{dhap}}{K_{G3PDH,dhap} + C_{dhap}}$

Serine synthesis: $r_{SerSynth} = r_{SerSynth}^{\max} \dfrac{C_{3pg}}{K_{SerSynth,3pg} + C_{3pg}}$

Mureine synthesis: $r_{MurSynth} = r_{MurSynth}^{\max}$

DAHPS: $r_{DAHPS} = \dfrac{r_{DAHPS}^{\max} C_{e4p}^{nDAHPS,e4p} C_{pep}^{nDAHPS,pep}}{(K_{DAHPS,e4p} + C_{e4p}^{nDAHPS,e4p})(K_{DAHPS,pep} + C_{pep}^{nDAHPS,pep})}$

Tryptophan synthesis: $r_{TrpSynth} = r_{TrpSynth}^{\max}$

Methionine synthesis: $r_{MetSynth} = r_{MetSynth}^{\max}$

G6PDH: $r_{G6PDH} = \dfrac{r_{G6PDH}^{\max} C_{g6p} C_{nadp}}{(C_{g6p} + K_{G6PDH,g6p})(1 + \dfrac{C_{nadph}}{K_{G6PDH,nadph,g6pinh}})(K_{G6PDH,nadp}(1 + \dfrac{C_{nadph}}{K_{G6PDH,nadph,nadpinh}}) + C_{nadp})}$

PGDH: $$r_{PGDH} = \frac{r_{PGDH}^{\max} C_{6pg} C_{nadp}}{(C_{6pg} + K_{PGDH,6pg})(C_{nadp} + K_{PGDH,nadp}(1 + \frac{C_{nadph}}{K_{PGDH,nadphinh}})(1 + \frac{C_{ATP}}{K_{PGDH,atp,inh}})}$$

RU5P: $$r_{Ru5P} = r_{Ru5P}^{\max}\left(C_{ribu5p} - \frac{C_{xyl5p}}{K_{Ru5P,eq}} \right)$$

R5PI: $$r_{R5PI} = r_{R5PI}^{\max}\left(C_{ribu5p} - \frac{C_{rib5p}}{K_{R5PI,eq}} \right)$$

TKa: $$r_{TKa} = r_{TKa}^{\max}\left(C_{rib5p} C_{xyl5p} - \frac{C_{sed7p} C_{gap}}{K_{TKa,eq}} \right)$$

TKb: $$r_{TKb} = r_{TKb}^{\max}\left(C_{xyl5p} C_{e4p} - \frac{C_{f6p} C_{gap}}{K_{TKb,eq}} \right)$$

TA: $$r_{TK} = r_{TK}^{\max}\left(C_{gap} C_{sed7p} - \frac{C_{e4p} C_{f6p}}{K_{TK,eq}} \right)$$

Synthesis 1: $$r_{Synth1} = r_{Synth1}^{\max} \frac{C_{pep}}{K_{Synth1,pep} + C_{pep}}$$

Synthesis 2: $$r_{Synth2} = r_{Synth2}^{\max} \frac{C_{pyr}}{K_{Synth2,pyr} + C_{pyr}}$$

For substrate (here mainly sugar) transport system, equation of balance depends on the dynamic of the transport system. One very common example is phosphotransferase system, for which the balance equation is taken from Liao et al. (1996) and expressed as:

$$\frac{\Delta C_{glc}^{extracellular}}{\Delta T} = D(C_{glc}^{feed} - C_{glc}^{extracellular}) - \frac{C_x r_{transport\ \ reaction}}{\rho_x}$$

where, C_{glc}^{feed} is the glucose (sugar) concentration in the feed, $C_{glc}^{extracellular}$ is the extracellular glucose concentration, C_x is the biomass concentration and, ρ_x is the specific weight of the biomass.

The dynamic behaviors of co-metabolites such as ATP, ADP, NAD, and NADP are also represented by analytical functions shown in Table 4-3. These have been taken from Chassangole *et al*, (2002) and Rizzi *et al*. (1997) corrected for typographical errors.

Table 4-3: Analytical function for co-metabolites

Analytical function for co-metabolites proposed by Chassagnole et al.
$C_{atp} = 4.27 - 4.163\dfrac{t}{0.657 + 1.43t + 0.0364t^{2}}$
$C_{adp} = 0.582 + 1.73(2.731^{-0.15t})(0.12t + 0.000214t^{3})$
$C_{amp} = 0.423 + 7.25\dfrac{t}{7.25 + 1.47t + 0.17t^{2}} + 1.073\dfrac{t}{1.29 + 8.05t}$
$C_{nad} = 1.314 + 1.314(2.73^{(-0.0435t-0.342)}) - \dfrac{(t + 7.871)(2.73^{(-0.0218t-0.171)})}{8.418 + t}$
$C_{nadh} = 0.0934 + 0.0011(2.371^{-0.123t})(0.844t + 0.104t^{3})$
$C_{nadp} = 0.195 - 0.0440\dfrac{t}{2.8 + 0.0271t + 0.01t^{2}} + 0.0182\dfrac{t}{4.31 + 0.526t}$
$C_{nadph} = 0.062 + 0.332(2.718^{-0.464t})(0.0166t^{1.58} + 0.000166t^{4.73} + 1.13\times10^{-10}\times t^{7.89}$ $+ 1.36\times10^{-13}\times t^{11} + 1.23\times10^{-16}\times t^{14.2})$

Steady state concentrations for some of the metabolites are available from direct measurement of intracellular metabolites. The remaining concentrations were estimated using thermodynamic relations such as near equilibrium assumptions or relation from published literature (Chassagnole et al., 2002). All measured and estimated steady state concentrations are listed in Table 4-4.

4.2.1 Glucose pulse experiment

To validate the model, an experiment was carried out using standard stirred-tank bioreactor with an *E. coli* culture under glucose limitation growth condition. Starting from steady state conditions at D =0.1 ± 0.002 h^{-1} with a extracellular glucose concentration of 12 ± 1.5 mg/L, 3ml sterilized solution of glucose was injected to conduct the pulse experiment. After injection, samples were collected every 3 s. Chassagnole et al. (2002) reported the results shown in Figure 4-3.

Table 4-4: Estimated and Measured Steady-state concentrations of Metabolites

Metabolite	Type	Concentration mM
2PG	Estimated	0.399
3PG	Estimated	2.13
6PG	Measured	0.808
ADP	Measured	0.595
AMP	Measured	0.955
ATP	Measured	4.27
DHAP	Estimated	0.167
E4P	Estimated	0.098
F6P	Measured	0.60
FDP	Measured	0.272
G1P	Measured	0.653
G6P	Measured	3.48
GAP	Measured	0.218
Glucose (extracellular)	Measured	0.0556
NAD	Measured	1.47
NADH	Measured	0.1
NADP	Measured	0.195
NADPH	Measured	0.062
PEP	Measured	2.67
PGP	Estimated	0.008
PYR	Measured	2.67
RIB5P	Estimated	0.398
RIBU5P	Estimated	0.111
SED7P	Estimated	0.276
XYL5P	Estimated	0.138

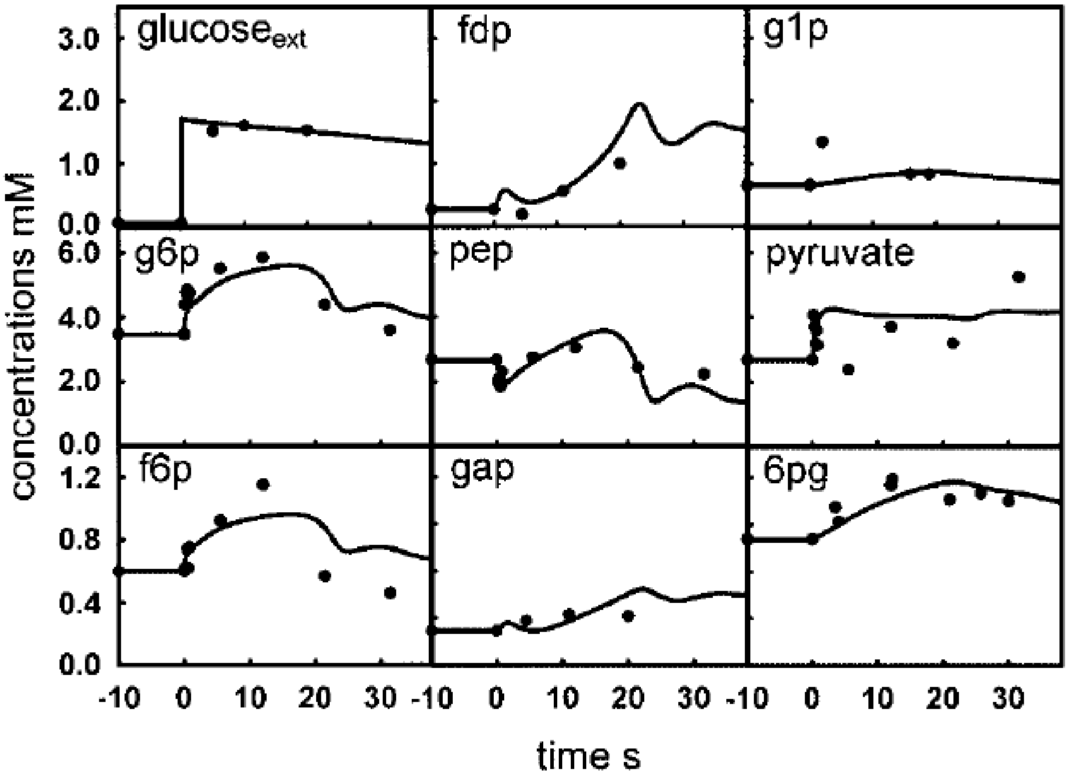


Figure 4-3: Comparison between experimental data and model predictions (Source: Chassagnole et al. 2002)

4.3 Dynamic Simulation using Agent-based model

The main purpose of a dynamic model is to describe the variation of the internal metabolome due to the continuously changing cellular state quantitatively in terms of metabolite concentration. The critical part of developing precise dynamic model is greatly related with incorporating characteristic kinetic information on the cellular dynamics, especially on different enzymatic reaction kinetics. The same agent-based framework can be extended to perform dynamic simulation provided the kinetics of metabolic reactions are available.

The process dynamic simulation is mainly based on the execution of reactions involved in the system. Like the stoichiometric agent model, here also the execution of reaction is designed to occur in a sequential manner. The response of the cell internal metabolome due to the dynamic system excitation is used to identify the dynamic system behavior by a stepwise internalization of metabolites. A group of reactions executed one after another based on the availability of their reactant metabolites and the corresponding enzymes activity. Prior to execution, a reaction agent search for the information regarding the present concentration of its input metabolites (reactants) and output metabolites (product) as well as the activity of corresponding enzymes. If all the required metabolites are present and enzyme is active, then the reaction will carried out to produce corresponding products. It updates the present concentrations of metabolites according to the stoichiometry and reaction kinetics and reports this change to the storage system. Thus all the reactions possible at that moment are executed in a sequential manner and information storage system is updated with the new values of metabolites. The change represents the total change of cellular metabolic system for that small time period. Another system is keeping the time and tracking the reactions' execution. Based on the updated information of metabolites a new set of reactions are now ready to be executed at the next time step. The process continues until the desired time set for the simulation.

The basic architecture for agent-based dynamic simulation followed the same framework as in the qualitative simulation described in Chapter 3. The same basic agents namely the Cytoplasm agent, Reaction agent, and Scheduler agent were used. In the following we describe modifications required to the Reaction and Scheduler agent in order to provide the dynamic simulation capability.

4.3.1 Reaction Agent

In order to capture the dynamics, the Reaction agents were embedded with additional attributes relating to effectors, the number of effectors and their kinetic parameters. Further, changes were made to reflect changes in cellular metabolome according to the kinetics. Specifically, the rate of reaction was determined using the kinetic expression of corresponding enzymatic reaction. In general, the rate of enzymatic reaction *j* is given by:

$$r_j = \mathrm{r_j^{max}} f_j(C_j, K_j)$$

where K$_j$ is the kinetic parameter vector and C$_j$ is the concentration vector of metabolites and effectors involved in the reaction. Typical example of $f_j(C_j, K_j)$ are Michaelis-Menten equation, Hill equation, etc. For reversible reactions, the direction of reaction is determined based on the sign of reaction rate, $\mathbf{r}_j$ calculated using kinetic expressions. If $\mathbf{r}_j$ is positive, reaction *j* will occur in forward direction, if $\mathbf{r}_j$ is zero reaction, *j* will not proceed and if negative, it will proceed in reverse direction.

The rate of change in internal metabolome in terms of metabolic concentration due to reaction *j* for a small time ΔT is given by:

$$\frac{\Delta C_{ij}}{\Delta T} = \pm v_{ij} r_j$$

where, r_j is the rate of reaction *j*, v_{ij} is the stoichiometric coefficient of metabolite *i* in reaction *j* and '-' sign is for reactants and '+' sign is for products So, concentration of metabolite *i* at time *T*, after reaction *j* has completed is given by:

$$C_i(T + \Delta T) = C_i(T) \pm v_{ij} r_j \times \Delta T$$

The agent based simulation can also directly implement the analytical functions for the co-metabolites.

4.3.2 Other Agents

Modifications were also required for the scheduler agent in order to simulate the dynamics. The sequential execution of reactions is controlled by the Scheduler agent. Specifically, it was extended to keep track of time during simulation time.

An additional agent (Injection agent) was added to emulate the glucose injection in the experimental study. The Injection agent informs the Scheduler agent regarding the time of injection. Being informed from the Scheduler agent at the specified time, Injection agent performs injection pulse by changing the glucose concentration inside the cytoplasm. Figure 4-4 illustrates the message exchange between the Injection agent, the Scheduler, and Cytoplasm agents.

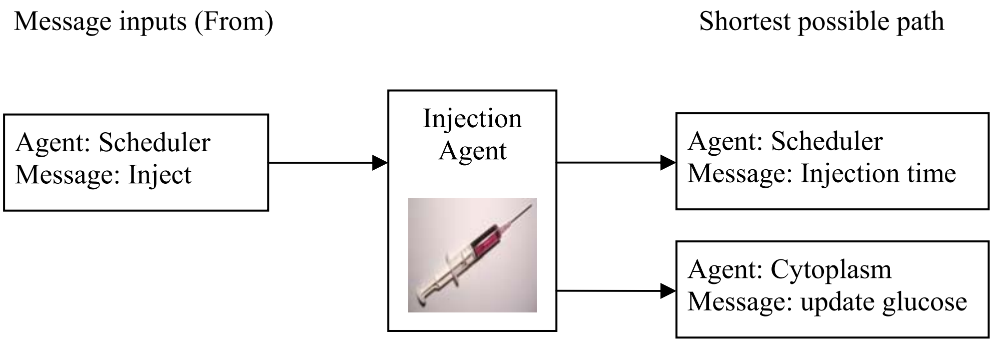


Figure 4-4 Message exchange for Injection Agent

4.3.3 Steps in Agent-based Dynamic simulation

In summary, the steps involved in the agent-based dynamic simulation are as follows:

1. At time T = 0, the simulation begins with the cytoplasm agent. Cytoplasm agent initiates itself by reading information regarding the initial condition of the system. Next, it uses JADE agent container to activate other agents in the system.
2. Scheduler agent initiates itself by filling the agent queue with all the reactions involved in the system.

3. Then Cytoplasm agent sends a message to scheduler to start triggering the reactions and at that time point, Scheduler agent triggers the first reaction agent from the agent queue.
4. The reaction agent sends the queries regarding the reactants, products, enzymes, and effectors and received the query replies from the Cytoplasm agent.
5. Reaction agent executed based on the kinetics of the reaction.
6. Reaction agent sends message to Cytoplasm agent to update metabolite concentrations of the system and confirm Scheduler agent about the completion of reaction.
7. Scheduler agent then triggers the next reaction in the queue.
8. Thus at a certain time point Scheduler agent sequentially triggers all the reactions possible at that time point.
9. When the reaction queue became empty the scheduler update the time of the process as T + ΔT. Again all the reactions involved in the model are added to the reaction queue and simulation continues.

4.4 Simulation Results

4.4.1 Steady state Simulation

The agent-based simulation was used to estimate the steady-state concentrations for intermediate metabolites. Starting from arbitrary conditions close to the reported steady state value, the ABS is able to reach the same Steady state conditions as described in the literature. Figure **4-5** (a) and (b) show the concentration profile of the intracellular metabolites in the course of reaching the steady state.

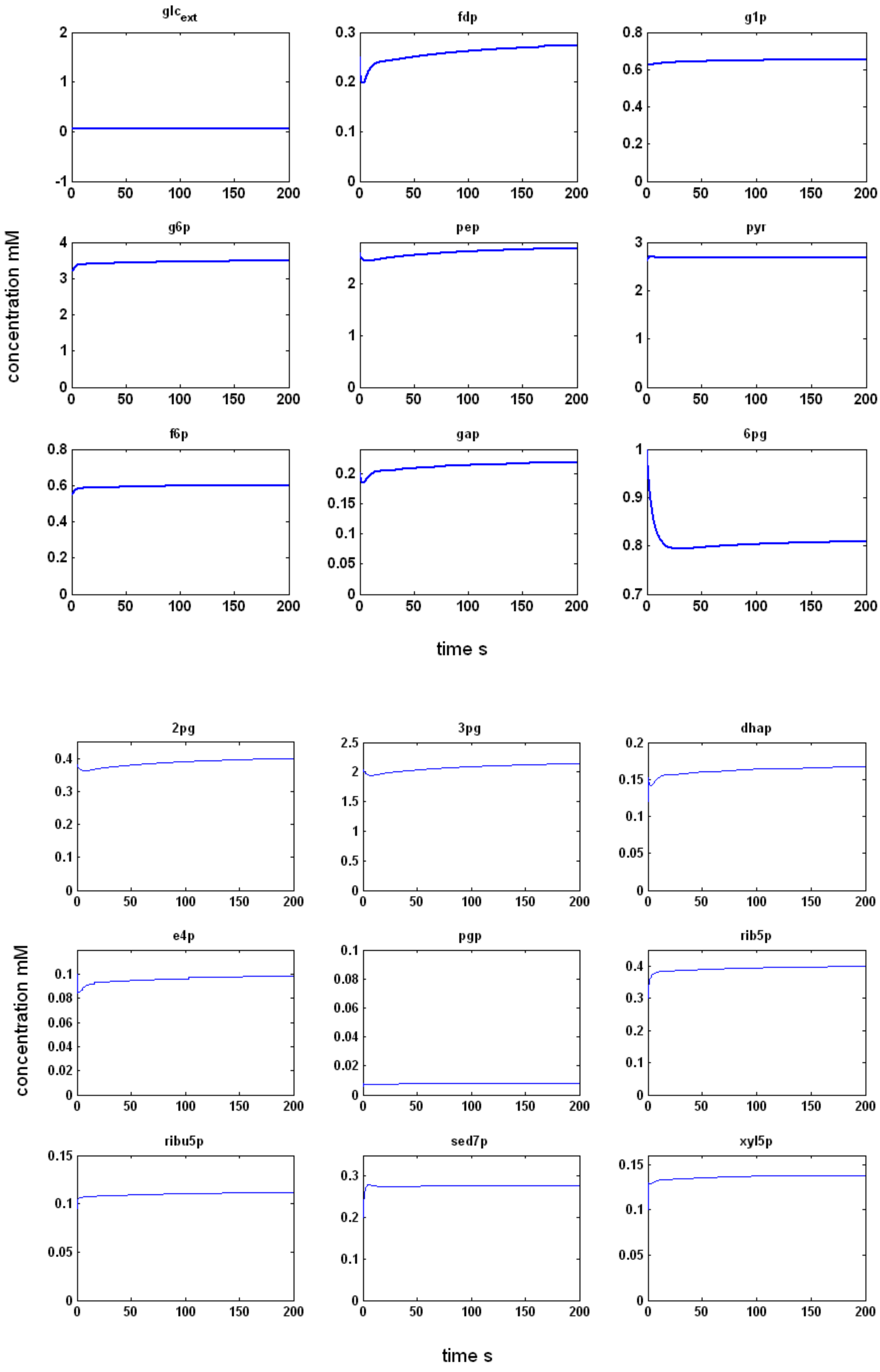


Figure 4-5: System reaching Steady-State for metabolites: (top): glc$_{ext}$, fdp, g1p, g6p, pep, pyr, f6p, gap and 6pg , (bottom): 2pg, 3pg, dhap, e4p, pgp, rib5p, ribu5p, sed7p, xyl5p

The resulting steady state values are used as the initial condition for the simulation of dynamic response of intracellular metabolites in glucose pulse experiment. Table 4-5 includes the steady state values of the metabolites as generated from the simulation along with the value reported by Chassagnole et al. (2002).

Table 4-5: Steady state concentration of the metabolites

Metabolite	Concentration mM	
	Literature	Simulation
2PG	0.399	0.398
3PG	2.13	2.131
6PG	0.808	0.808
ADP	0.595	0.595
AMP	0.955	0.955
ATP	4.27	4.27
DHAP	0.167	0.167
E4P	0.098	0.098
F6P	0.60	0.60
FDP	0.272	0.272
G1P	0.653	0.653
G6P	3.48	3.481
GAP	0.218	0.218
Glucose (extracellular)	0.0556	0.0556
NAD	1.47	1.47
NADH	0.1	0.1
NADP	0.195	0.195
NADPH	0.062	0.062
PEP	2.67	2.672
PGP	0.008	0.008
PYR	2.67	2.669
RIB5P	0.398	0.398
RIBU5P	0.111	0.111
SED7P	0.276	0.276
XYL5P	0.138	0.138

4.4.2 Dynamic Simulation

4.4.2.1 Selecting step size

As stated earlier, a set of reactions occurred at a particular point of time in an iterative manner for a small time period, ΔT. The optimum step size for ΔT is a major concern to capture the dynamics of biochemical reactions as they are usually not very fast. Starting from arbitrary initial condition, for different size of ΔT, we have plotted concentrations against time profile of various intermediates and noticed that below ΔT = 0.0001s, the effect of ΔT on the concentration profile become insignificant. So, ΔT =0.0001s has been taken for the simulation. To demonstrate the effect of ΔT on simulation result, concentration profile of metabolite g6p, pep, pyr and f6p are plotted for ΔT = 0.001s, 0.0001s and 0.00001s. For each case the simulation is run for 35 simulation second. For ΔT = 0.001 s, 0.0001s and 0.00001s, it takes roughly 19 minutes, 238 minutes and 2790 minutes respectively. Figure 4-6 to figure 4-8 illustrates the effect of step size on capturing the system dynamics. The profiles for ΔT =0.0001s and ΔT = 0.00001s are identical to each other. While in case of ΔT = 0.0001s the simulation is more than 10 times faster compared to the case for ΔT = 0.00001s. This implies that ΔT =0.0001s is sufficient to capture the dynamic responses of the system.

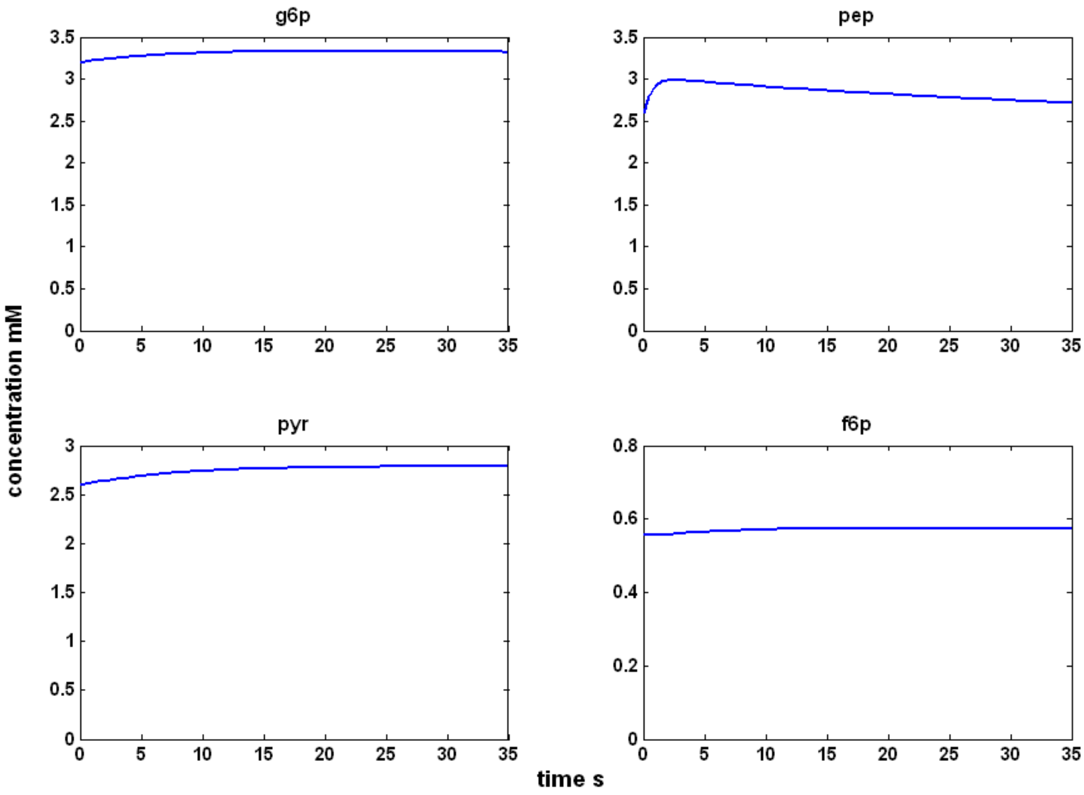


Figure 4-6: Effect of **ΔT** on Concentration (**ΔT** =0.001s)

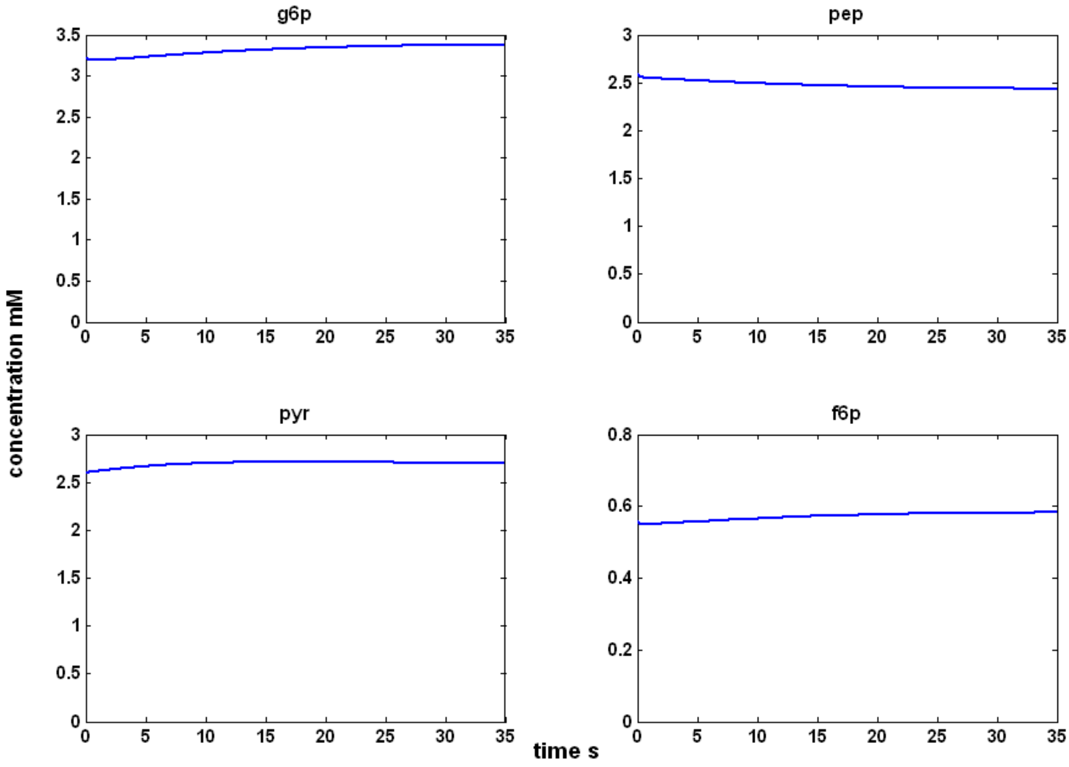


Figure 4-7: Effect of **ΔT** on Concentration (**ΔT** =0.0001s)

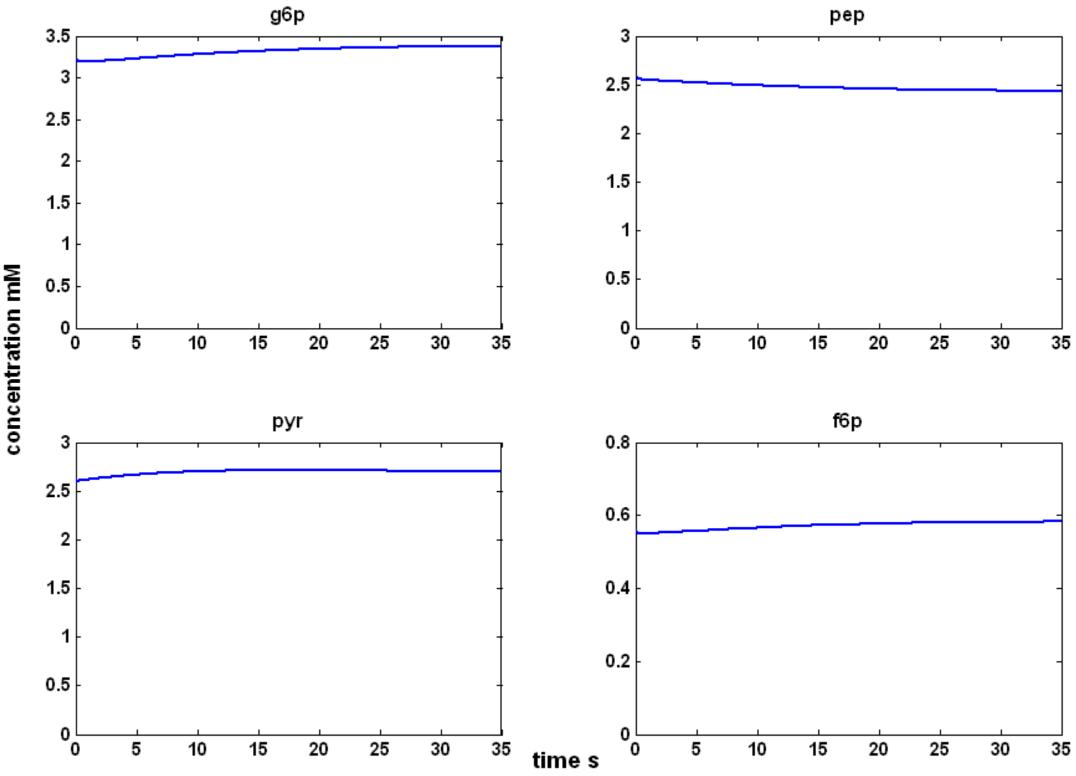


Figure 4-8: Effect of **ΔT** on Concentration (**ΔT** =0.00001s)

4.4.2.2 Validation of Simulation results

The time course of the co-metabolite concentrations resulting from the functional expression are verified with experimental data.

The simulation results fit the observed trends for most metabolites collected by manual sampling after the pulse of glucose into the bioreactor as reported by Chassagnole et al. (2002). Figure 4-10 compares the current simulation with the experimental results reported in Chassagnole's work. The same model has been simulated using MATLAB ode45 solver. Figure 4-11 shows the results of the MATLAB simulation. Table 4-6 includes the performance of agent-based simulation and simulation using MATLAB.

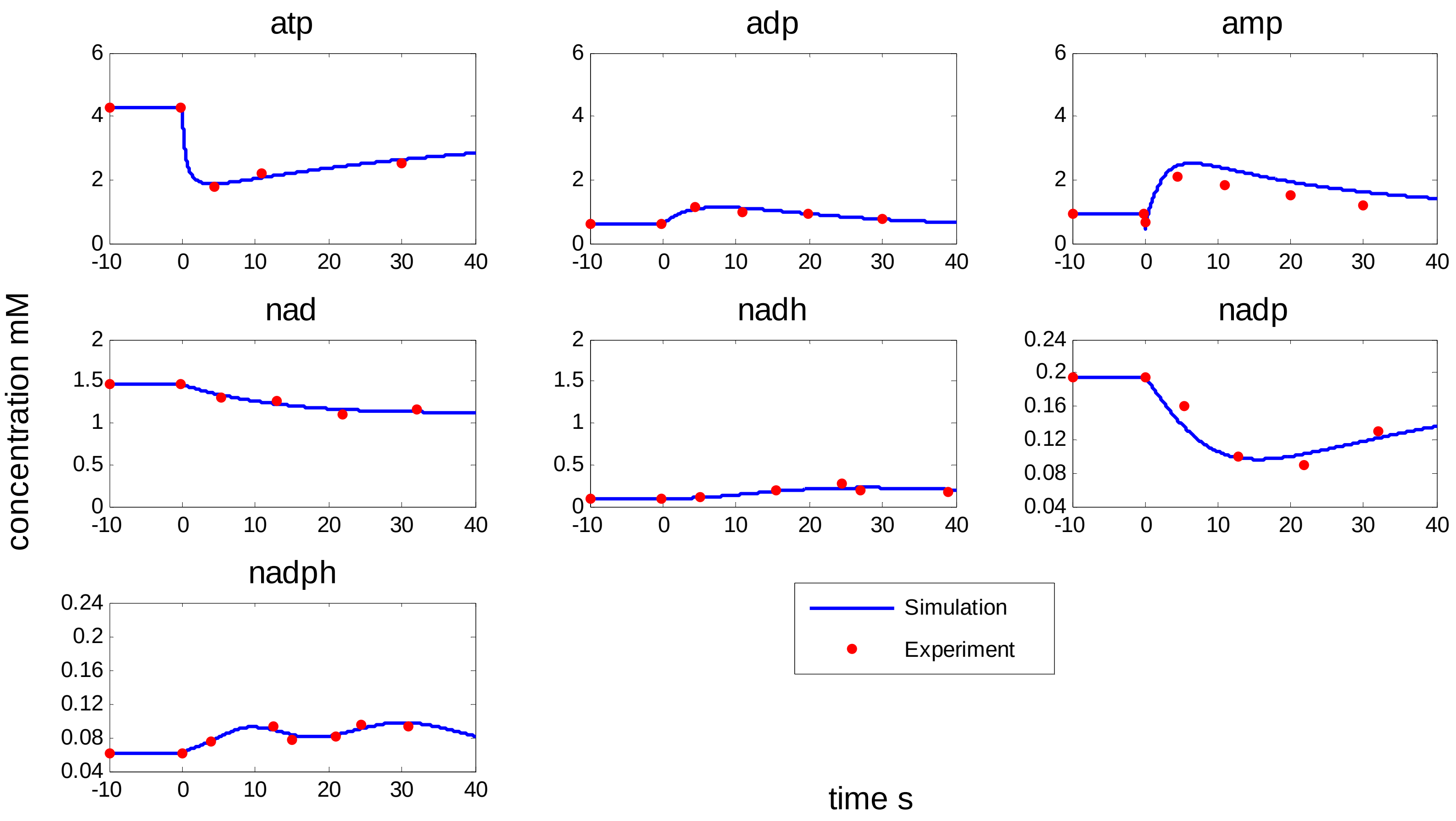


Figure 4-9: Time course for the co-metabolites

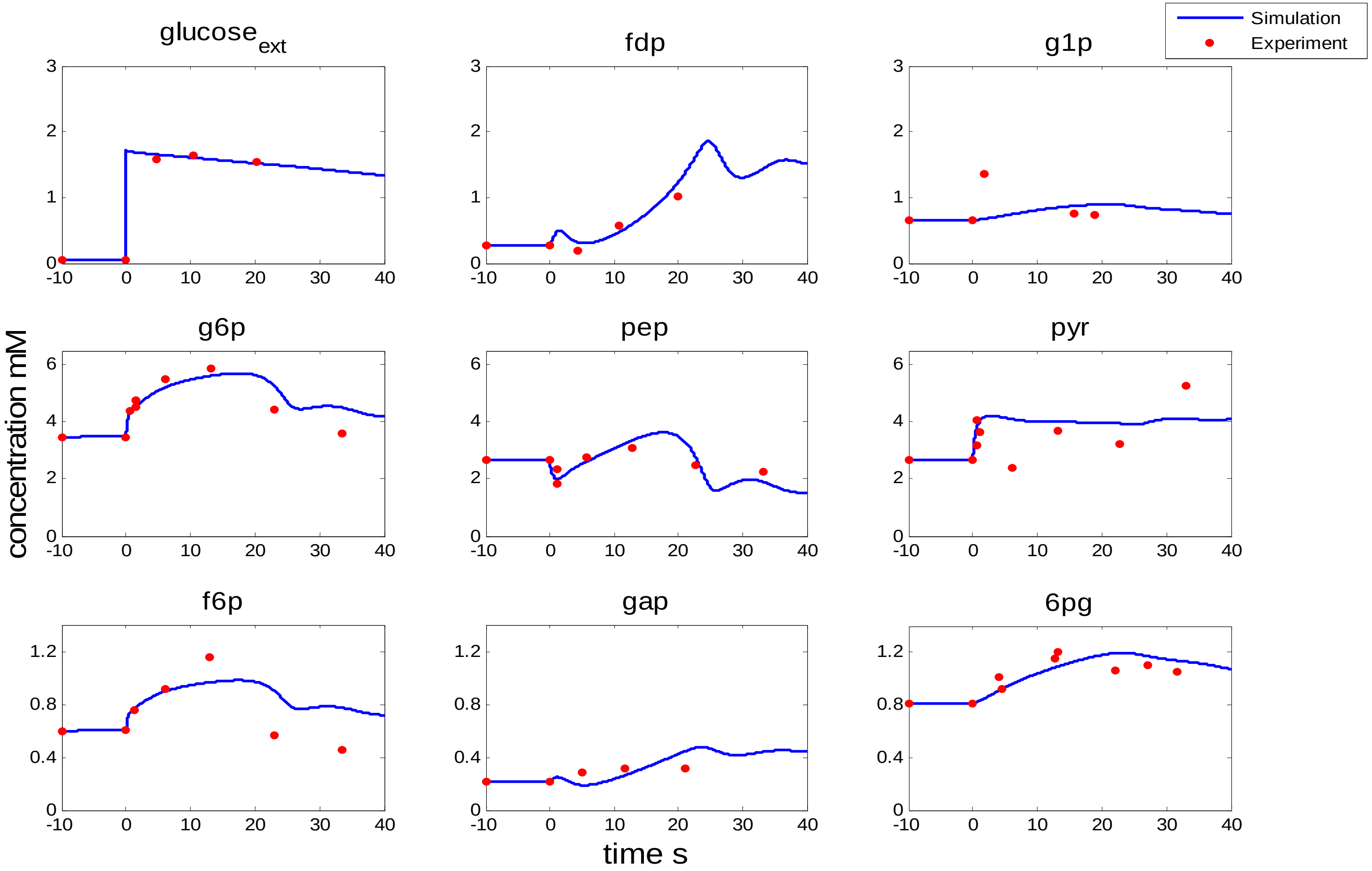


Figure 4-10: Comparison between experimental data (red dots) and model simulations (blue lines) in response to a glucose pulse at time zero in steady state culture.

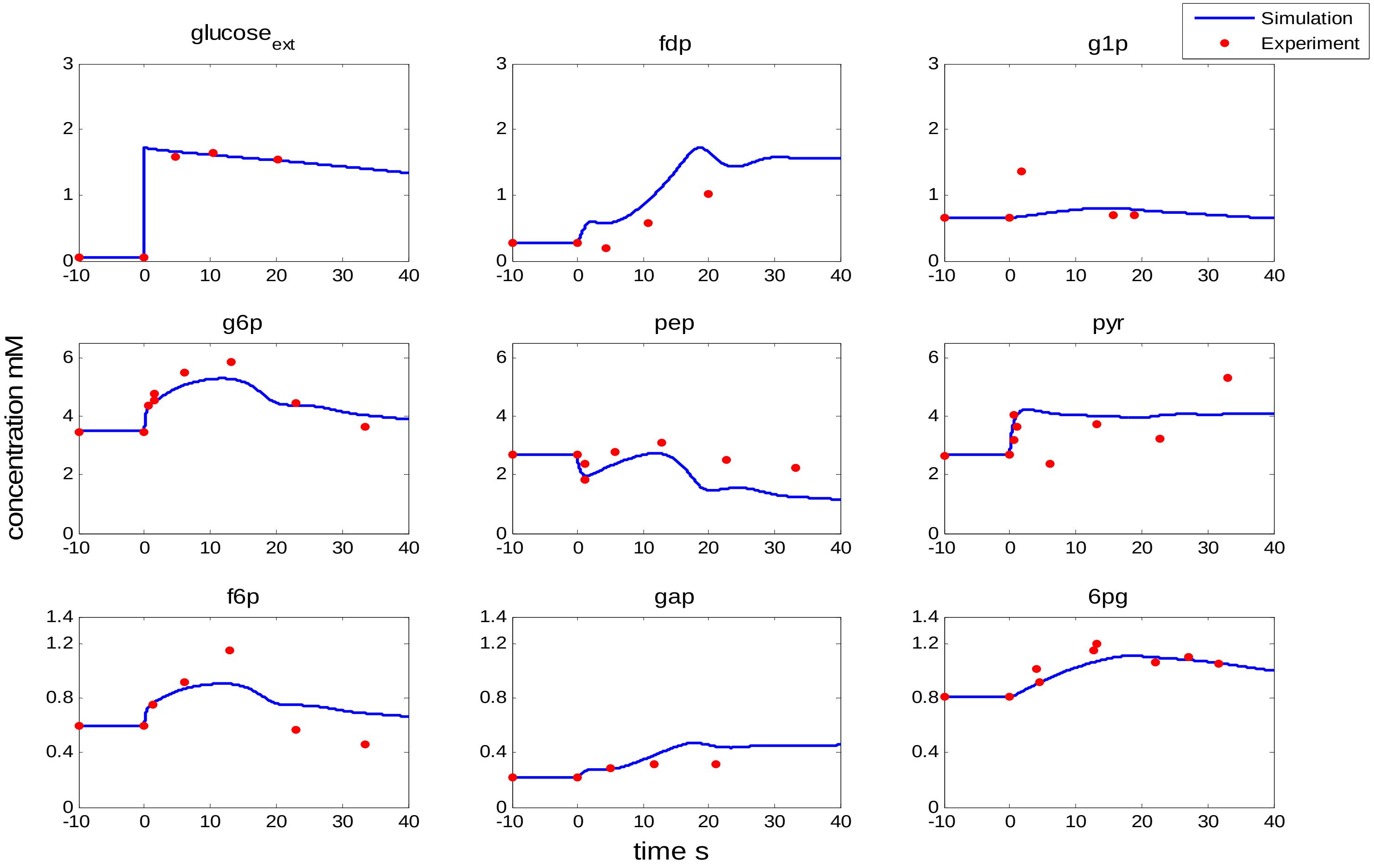


Figure 4-11: Comparison between experimental data (red dots) and model simulations by MATLAB (blue lines) in response to a glucose pulse at time zero in steady state culture.

Table 4-6: Comparison between Agent-based simulation and MATLAB Simulation

Metabolite	Glu_{ext}	fdp	g1p	G6p	pep	pyr	f6p	gap	6pg
Agent-Based Simulation									
% error	1.04	19.69	17.93	6.99	5.28	15.07	19.44	15.0	5.50
Simulation Time			50 sec simulation need 6792 sec of computational time						
MATLAB Simulation using ode45 solver									
% error	1.70	67.72	15.71	6.78	16.73	18.82	19.41	15.30	6.18
Simulation Time			50 sec simulation need 211 sec of computational time						

The agent based simulation was able to reproduce the same results capturing both the experimental data and Chassagnole et al. (2002) model predictions. Chassagnole et al. (2002) discussed the reasons for the discrepancy between the model and the experimental observations and concluded that these occurred due to well known difficulties in assessing the actual kinetic phenomena that governs the dynamic behavior of the complex system like metabolic network. The mechanistic rate expressions used for the model developed based on limited understanding of the dynamics of the reactions. The discrepancy also demonstrates the sensitivity of the complex interacting system with respect to uncertainties in its detailed structure, which is very difficult to capture by parameter fitting.

4.5 Concluding remarks

In this chapter, the agent-based model of the metabolic network proposed in the previous chapter was also used for dynamic simulation. As case study, the central metabolic pathways of *E. coli* were selected. The agent-based framework was extended by incorporating kinetics in the Reaction agent. The scheduler agent was also extended by incorporating time factors. Experimental data from literature, where a glucose pulse had been injected into a steady state culture and dynamic response of metabolites recorded, served as the gold standard. The target was to re-create the experimental observations using dynamic simulation techniques. Simulation results confirmed the effectiveness the proposed modeling and simulation approach as it successfully captured the dynamics of the system and reproduced the same results as reported in literature. This study also shows the flexibility and extendibility of the agent based framework.

Chapter 5 Conclusions and Recommendations

Computational biology or in ***silico*** biology will be increasingly important as the scientific community is faced with the challenge of establishing the link between the genome scale model and the physiological functions of an organism. The computational analysis of genome sequence data is proving very useful; for example, 40 to 80% of the Open Reading Frame**s** identified in the fully sequenced organisms have been assigned a putative function. The next step is to derive thorough understanding of the genotype-phenotype relationship of the organism. When the results from genome sequencing projects are combined with bioinformatics analysis, a comprehensive metabolic model can be developed. The reconstruction and simulation of the overall cellular functions based on high throughput experimental data can pave the way to designing organisms to produce high-value metabolites. Current methods for reconstructing and simulating metabolic models are stymied by inconsistent and incomplete information of the metabolic network. A key challenge is to elucidate these inconsistencies and bridge them efficiently. In this work, a new agent based modeling and simulation approach has been proposed to analyze metabolic pathways.

In contrast to monolithic mathematical models of metabolism, the current work is centered around an individual based modeling paradigm. This paradigm, where the behavior of a complex system emerges from the interactions of simple individuals, each with its own resources and goals, has been successfully applied in other domains; to our knowledge this is the first such proposal in the metabolic engineering domain. To represent the metabolic activities of a single compartment organism, the proposed architecture uses three different classes of interacting agents, namely Cytoplasm agent, Scheduler agent and Reaction agents. Each model typically includes many instances of Reaction agents, each modeling the metabolite uptake and production of a metabolic reaction. Reaction agents interact with each other based on shared metabolites; the structural and the dynamic properties of the entire network emerges from these local interactions. We have shown that this distributed modeling architecture is specifically suited for indentifying network inconsistencies. A qualitative simulation based approach for identifying network gaps and a search based method for filling gaps have been proposed. Using the central metabolism of *E. coli* as a model system, the developed framework has demonstrated to effectively identify and fill gaps in both linear and branched networks. With minimum modification, the same framework has also been extended to emulate the dynamic behavior of metabolic networks using quantitative kinetic models of the reactions. This dynamic simulation has also been demonstrated on the central metabolism of *E. coli*. The results were to found to match well with those reported in literature.

While developing the agent based approach for metabolic network model, several attractive features of the modeling paradigm become apparent. Firstly, ABMS has the ability to capture emergent properties of highly interactive systems. The local interactions of individuals in highly networked systems gives rise to global consequences, which cannot be attributed to any single individual in the system. This ‘‘emergent property’’, a characteristic of the system as a whole with no significance at the individual level, distinguishes a complex system from an ordinary one. Secondly, the key aspect of an agent-based modeling framework is the interaction of an agent with other fellow agents and its immediate environment, which is very much appropriate to model the metabolic activities inside the cell cytoplasm. Thirdly, the natural modularity as evident in this framework helps one to exploit all the advantages characteristic of an object-oriented paradigm. Any new component, activities, or experimental arrangement can be modeled as an individual agent or object with unique functionality, without affecting the basic architecture of the system. For example, during the dynamic simulation of *E. coli*, the injection pulse experimental technique was easily designed as an additional agent and effectively implemented into the main structure.

The current work is, to our knowledge, the first agent-based model of metabolic networks. It can be further extended in several ways. The data structure used in the proposed framework is not directly suited for use with SBML (System Biology Markup Language) supported databases. SBML is now become the global data format for representing models of biochemical reaction networks, like metabolic network, cell signaling networks, etc. Currently, manual conversion of online available databases into the compatible format for the proposed structure is required. This limitation could be resolved by developing a parser module, which is able to access the online databases and convert them to a compatible version for the proposed agent framework. Furthermore, for the dynamic simulation, the operation is sequential and iterative, hence comparatively slow in generating results compared to traditional differential equation solvers. Some effort is required to come up with an efficient algorithm for distributed dynamic simulation.

We see agent based modeling approach as a new and potential tool for modeling the complex organization in the cell. Though the developed agent based framework is designed for a single compartment (where all reactions occur in cytoplasm) prokaryotic organism modeling, the same approach could be extended to model eukaryotes with multiple compartments. The agent-based approach is very suitable for capturing the effect of spatial arrangement of intercellular compartment. Agent-based model can potentially be applied to capture the emergent properties of eukaryotes arising from intra-organelle interactions. We hope that agent-based modeling and simulation (ABMS) along with other artificial intelligence techniques will help to reveal the complexities of intra and inter cellular processes.

References

Albe, K., Wright, B., (1992), Systems analysis of the tricarboxylic acid cycle in *Dictyostelium discoideum*: II. Control analysis, *Journal of Biological Chemistry*, 267, 3106–3114.

Alper, H., and Stephanopoulos, G., (2004), Metabolic engineering challenges in the post-genomic era, *Chemical Engineering Science,* 59, 5009-5017.

Alur, R., Belta, C., Kumar, V., Mintz, M., Pappas, G. J., Rubin, H. and Schug, J., (2002), Modeling And Analyzing Biomolecular networks, *Computing in Science & Engineering, 4 (1),* 20, database: IEEE Xplore.

Aristidou, A.A., San, K.-Y., and Bennet, G.N., (1990), Improvement of biomass yield and recombinant gene expression in *Escherichia coli* by using fructose as the primary carbon source, *Biotechnology Progress,* 15, 140-145.

Aristidou, A.A., San, K.-Y., and Bennet, G.N., (1995), Metabolic Engineering of *Escherichia coli* to enhance recombinant protein production through acetate reduction, *Biotechnology Progress,*11, 475-478.

Bailey, J.E., (1991), Towards a science of metabolic engineering. *Science* 252, 1668-1675.

Bodik, T.A., and Rasche, M.E., (2001), Identification of the human methylmalonyl-CoA recemase gene based on the analysis of prokaryotic gene arrangements. Implications for decoding the human genome, Journal of Biological chemistry, 276, 37194-37198.

Burleigh, I., Suen, G. and Jacob, C., (2003), DNA in Action! A 3D Swarm-based Model of a Gene Regulatory System, Proceedings of the First Australian Conference on Artificial Life, Canberra, Australia.

Cameron, D. C., and Tong, I. T., (1993), Cellular and Metabolic Engineering, *Applied Biochemistry and Biotechnology,* 38, 105-140.

Chassagnole, C., Rizzi, N. N., Schmid, W., Mauch, K., and Reuss, M., (2002), Dynamic modeling of the central metabolism of *Escherichia coli*, *Biotechnology and Bioengineering,* 79(1), 53-73.

Chen, L. and Vitkup, D. (2006), Predicting genes for orphan metabolites activities using phylogenetic profiles, *Genome Biology* (7), R 17.

Covert, M. W.; Palsson, B. O., (2002), Transcriptional regulation in constraints-based metabolic models of *Escherichia coli. J. Biol.Chem. 277* (31), 28058-28064.

Covert, M. W.; Palsson, B. O., (2003), Constraints-based models: Regulation of gene expression reduces the steady-state solution space. *Journal of Theoretical Biology. 221* (3), 309-325.

Edwards, J., Palsson, B., (1998), How will bioinformatics influence metabolic engineering?, *Biotechnology and Bioengineering* 58, 162–169.

Gadkar, K. G.; Doyle, F. J., III; Edwards, J. S.; Mahadevan, R., (2005), Estimating optimal profiles of genetic alterations using constraint based models. *Biotechnology Bioengineering, 89* (2), 243-251.

Goldbeter A (1996) From ultradian biochemical oscilations to circadian rhythms. *Membranes and Circadian Rhythms.* T. Vanden Driessche (Ed.), Springer, Berlin, 67-93.

Gonzalez, P. P., Cardenas, M., Camacho, D., Franyuti, A., Rosas, O., and Lagunez-Otero, J., (2003), Cellulat: an agent-based intracellular signaling model, *Biosystems,68 (2-3),* 171-185.

Green, M. L., and Karp, P. D., (2004), A Bayesian method for identifying missing enzymes in predicted metabolic pathway databases, *BMC Bioinformatics*, 5, 76.

Heinrich, R., Rapaport, S. M., and Rapaport, T.A., (1977), Metabolic regulation and Mathematical models, *Progress in Biophysics and Molecular Biology. 32,* 1-82.

Heinrich, R., Schuster, S., (1996), The Regulation of Cellular Systems. Chapman & Hall, New York.

Henriksen, C. M., Chritensen, L. H., Nielsen, L., and Villadsen, J., (1996), Growth energetics and metabolism fluxes in continuous culture of *Penicillium chrysogenum, Journal of Biotechnology,* 45, 149-164.

Ikeda, M., and Katsumata, R., (1994), Transport of Aromatic Amino Acids and Its Influence on Overproduction of the Amino Acids in *Corynebacterium glutamicum, Journal of Fermentation and Bioengineering,* 78, 420-425.

Ikeda, M., Nakanisho, K., Kino, K., and Katsumata, R., (1994), Fermentative production of tryptophan by a stable recombinant strain of *Corynebacterium glutamicum* with a modified serine-biosynthetic pathway, *Bioscience Biotechnology and Biochemistry,* 58, 674-678.

Ishii, N., Robert, M., Nakayama, Y., Kanai, A., Tomita, M., (2004), Towards large scale modeling of microbial cell for computer simulation, *Journal of Biotechnology,* 113, 281-294.

Jackson, D. A., Symons, R. H., and Berg, P., (1972), Biochemical methods for inserting new genetic information into DNA of Sirnian Virus 40: circular SV40 DNA molecules containing lambda phage genes and the galactose operon of *Escherichia coli., Proceedings of the National Academy of Sciences of the United States of America, 69,* 2904-2909.

Kacser, H., and Burns, J. A., (1973), The control of flux, *Symposium for the Society of Experimental Biology.* 27, 65-104.

Katare, S. and Venkatasubramanian, V., (2001), An agent-based learning framework for modeling microbial growth, *Engineering Applications of Artificial Intelligence, 14 (6),* 715-726.

Keasling lab, UCB, Lawrence Berkeley National Laboratory, http://keaslinglab.lbl.gov/wiki/index.php/Main_Page, last accessed on 11 Dec, 2007.

Kellogg, S.T., Chatterjee, D.K. and Charkrabarty, A.M., (1981), Plasmid-assisted molecular breeding: new technique for enhanced biodegradation of persistent toxic chemicals. *Science* 214, 1133-1135.

Kharchenko, P., Vitkup, D., Church, G. M., (2004), Filling gaps in a metabolic network using expression information. *Bioinformatics* (20), I 1178- I 1185.

Kharchenko P., Chen L, Freund Y, Vitkup D, Church G. M., (2006), Identifying metabolic enzymes with multiple types of association evidence, BMC Bioinformatics, 7, 177.

Lee, S.Y. and Papoutsakis, E.T., Metabolic Engineering. Marcel Dekker, (1999) New York.

Liao, J.C., Hou, S., and Chao, Y., (1996), Pathway analysis, engineering and physiological considerations for redirecting central metabolism. *Biotechnology and bioengineering,* 52, 129-140.

MacQuitty, J.J., (1988), Impact of biotechnology on the chemical industry, ACS *Symposium series* 362, 11-25.

Mahadevan, R.; Edwards, J. S.; Doyle, F. J., III, (2002), Dynamic flux balance analysis of diauxic growth in *Escherichia coli*, *Biophysics Journal, 83* (3), 1331-1340.

Martin, V. JJ., Pitera, D. J., Withers, S. T., Newman, J. D., and Keasling, J. D., (2003), Engineering a mevalonate pathway in *Escherichia coli* for production of terpenoids, *Nature Biotechnology,* 21 (7), 796-802.

Neidhardt, F. C., Curtiss, R III., Ingraham, J. L., Lin, E. C. C., Low, K. B., Magasanik, B., Reznikoff, W. S., Riley, M., Schaechter, M., and Umbarger, H. E., (1996), *Escherichia coli* and *Salmonella:* Cellular and Molecular Biology, ASM press, Washington DC.

Nerem, R.M., (1991), Cellular Engineering, *Annals of Biomedical Engineering* 19, 529-545.

Notebaart, R.A., van Enckvort, F.H., Francke, C., Siezen, R.J., Teusink, B., (2006), Accelerating the reconstruction of genome-scale metabolic networks, *BMC Bioinformatics*, 7, 296.

Ohta, K., Beall, D. S., Mehia, J. P., Shanmugam, K. T., and Ingram, L. O., (1991), Metabolic engineering of klebsiella oxytoca M5A1 for ethanol production from xylose and glucose, *Applied and Environmental Microbiology,* 57, 2810-2815.

Osterman, A. and Overbeek, R., (2003), Missing genes in metabolic pathways: a comparative genomic approach, *Current opinion in Chemical Biology* (7), 238-251.

Papin J.A., Stelling, J., Price, N.D., Klamt, S., Schuster, S., Palsson, B.O.,(2004). Comparison of network-based pathway analysis methods. Trends Biotechnol. 22(8), 400-405.

Park, Y. S., Ohtake, H., Fukaya, M., Okumura, H., Kawamura, Y., and Toda, K., (1989), Enhancement of acetic acid production in a high Cell-density culture of *Acetobacter aceti, Journal of Fermentation and Bioengineering,* 68, 315-319.

Parunak, H.V.D., Savit, R., Riolo, R.L.,and Clark, S., “Dynamical Analysis of Supply Chains”, http://www.erim.org/cec/projects/dasch.htm, ERIM (1998). Available at http://www.erim.org/cec/projects/dasch.htm.

Paton, R. C., (1993), Some computational models at the cellular level, *Biosystems,29 (2-3),* 63-75.

Pellegrini, M., Thompson, M., Fierro, J., Bowers, P.,(2001), Computational method to assign microbial genes to pathways, *Journal od cellular Biochemistry,* Suppl 37, 106-109.

Pogson, M., Smallwood, R., Qwarnstrom, E., and Holcombe, M., (2006), Formal agent-based modeling of intracellular chemical interactions, *BioSystems 85,* 37-45.

Price, N.D., Reed, J.L., Palsson, B.O., (2004). Genome-scale models of microbial cells: evaluating the consequences of constraints. Nat. Rev. Microbiol. 2(11), 886-897.

Reed J.L, Famili I, Thiele I, Palsson B.O., (2006) Towards multidimensional genome annotation. *Nature Review Genetics,* 7, 130–141.

Rizzi, M., Baltes, M., Theoblad, U., and Reuss, M., (1997), In vivo analysis of metabolic dynamics in Saccharomyces cerevisiae: II. Mathematical model, *Biotechnology and Bioengineering,* 55, 592-608.

Russell, S., and Norvig, P., (1995), Artificial Intelligence— A Modern Approach. Prentice Hall, New Jersy.

Sauer, U., Hatzimanikatis, V., Bailey, J., Hochuli, M., Szyperski, T., and Wuthrich, K., (1997). Metabolic fluxes in riboflavin-producing *Bacillis subtilis, Nature Biotechnology,* 15, 448-452.

Savageau, M.A., (1969), Biochemical systems analysis. II. The steady state solutions for an n-pool system using a power-law approximation, *Journal of Theoretical Biology, 25*, 370-379.

Schilling, C.H., Schuster, S., Palsson, B.O., and Heinrich, R., (1999), Metabolic pathway analysis: Basic concepts and scientific applications in the post-genomic era, *Biotechnology Progress,* 15, 296-303.

Shoham, Y., and Tennenholtz, M., (1997), On the emergence of social conventions: modeling, analysis and simulations, *Artificial Intelligence, 94(1-2)*, 139-166.

Slater, S., Gallaher, T., and Dennis, D., (1992), Production of poly-(3-hydroxybutyrate-co-3-hydroxyvalerate) in a recombinant *Escherichia coli* strain, *Applied Environmental Microbiology*. 58, 1089-1094.

Stephanopoulos, G., (1999), Metabolic Fluxes and Metabolic Engineering, *Metabolic Engineering*, 1, 1-10.

Stephanopoulos, G., Aristidou, A.A., Nielsen, J.,(1998) Metabolic Engineering—Principles and Methodologies. Academic Press, New York.

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

Takors, R., Wiechert, W., Weuster-Botz, D., (1997). Experimental design for the identification of macrokinetic models and model discrimination, *Biotechnology and Bioengineering*. 56, 564–576.

Timmis, K.N., Rojo, F. & Ramos, J.L, (1988), In Environmental Biotechnology, Edited by Omenn, G.S. New York, NY:plenum press. 61.

Tong,I-T., Liao, H. H. & Cameron, D.C., (1991), 1,3-Propanediol production by Escherchia coli expression genes from the klebsiella puenumoniac dha regulation, *Applied and Environmental Microbiology* 57, 3541-3546.

Varma, A., and Palsson, B.O., (1994). Metabolic flux balancing: basic concepts scientific and practical use, *Biotechnology,* 12, 994-998.

VSIS project web site, University of Hamburg; http://vsis-www.informatik.uni-hamburg.de/projects/jadex/. Last accessed on Dec, 2007.

Weber, J. M., Leung, J. O., Swanson, S. J., Idler, K. B., and McAlpine, J. B., (1991), An erythromycin derivative produced by targeted gene disruption in *Saccharopolyspora erythraea, Science,* 252, 114-117.

Winter, R. B., Yen, K M., and Ensley, B. D., (1989), Efficent degradation of trichloroethylene by a recombinant *Escherichia coli, Bio-Technology,* 7, 282-285.

Wood, B. E., and Ingram, L. O., (1992), Ethanol production from cellobiose, amorphous cellulose, and crystalline cellulose by recombinant *Klebsiella oxytoca* containing chromosomally integrated *zymomonas mobilis* genes for ethanol production and plasmids expressing thermostable cellulose genes from *Clostridium thermocellum, Applied and Environmental Microbiology,* 58, 2103-2110.

Wooldridge, M., (1998), Agent-based computing, *Interoperable Communication Networks* 1(1), 71-97.

Yang, Y.-T., Aristidou, A. A., San, K.-Y., and Bennet, G. N., (1999), Metabolic Flux Analysis of *Escherichia coli* Deficient in the Aceate Production Pathway and Expressing the *Bacillus subtilis* Acetolactate Synthase, *Metabolic Engineering,* 1, 26-34.