METAHEURISTIC ALGORITHMS FOR OPTIMAL STRAIN DESIGN

Dissertation

for obtaining a doctorate degree

at the University of Natural Resources and Life Sciences Vienna

Submitted by

Govind Muralidhar Nair

Head of Department:

Ao.Univ.Prof. Dipl.-Ing. Dr.rer.nat. Reingard Grabherr

Advisor:

Univ.Prof. Dipl.-Ing. Dr.nat.techn. Diethard Mattanovich

Department of Biotechnology

Universität für Bodenkultur Wien
University of Natural Resources
and Life Sciences, Vienna

Vienna, March 2017

This work is dedicated to the preservation of biodiversity on our planet.

Abstract

Calculating metabolic intervention strategies that lead to desired outcomes is an important goal in metabolic modeling and analysis. Intervention strategies ensuring high growth-coupled product formation is of practical relevance in industrial biotechnology. The concept of constrained minimal cut sets (cMCS) is an valuable method in the arsenal of tools for manipulating metabolic networks. cMCS are minimal sets of reactions, the removal of which will force particular undesired fluxes to be blocked and some other desired ones to function. Two methods (viz., GAMCS and PSOMCS) have been developed in this work to calculate cMCS leading to optimal designs satisfying set objectives. GAMCS does this by partitioning the set of elementary flux modes (EFMs) of a network using a genetic algorithm (GA) and finding their corresponding cMCSs. PSOMCS depends on the concept of direct enumeration of cMCS and uses particle swarm optimization to find cMCSs corresponding to the optimal design. GAMCS was tested on *E. coli* metabolic networks of three different sizes. Three different engineering goals were set up and intervention strategies optimizing these goals were calculated. GAMCS had a superior performance against a method which enumerates all cMCS optimizing for a particular engineering objective. PSOMCS was also tested on a medium scale *E. coli* core metabolic network and shown to be have a performance orders of magnitude better than GAMCS. PSOMCS was also able to find intervention strategies leading to optimal designs in the iAF1260 genome-scale model of *E. coli* metabolism. Such designs were shown to be better than those produced by conventional strain design tools OptKnock and RobustKnock. Both methods found solutions comparable to previously published and experimentally verified results. GAMCS is capable of handling small and medium-scale metabolic networks within a reasonable time period. PSOMCS marks an improvement over this as it can handle genome-scale networks. Additionally, both methods are capable of handling complex design goals encoded by non-linear objective functions. The techniques presented here are capable of producing optimal designs satisfying multiple objectives. As the metabolic models of more industrially relevant organisms become available, these techniques will prove to be more useful.

Keywords: Systems biology, metabolic networks, elementary flux modes (EFMs), minimal cut sets (MCS), strain optimization, knockouts

Kurzfassung

Die rechnergestützte Vorhersage von genetischen Veränderungen zur Optimierung eines Produktionsstammes ist ein Hauptziel der metabolischen Modellierung und Analyse. Insbesondere die Vorhersage von Interventionsstrategien, die zu wachstumsgekoppelten Produktionsprozessen führen, ist von besonderer praktischer Relevanz für die industrielle Biotechnologie. Diesbezüglich erweist sich das Konzept der minimalen Schnittmengen unter Zwangsbedingen (constrained Minimal Cut Set, cMCS) als besonders wertvoll. cMCSs sind minimale Mengen von Reaktionen, die, wenn sie von einem Netzwerk entfernt werden, nur ungewünschte Funktionen unterdrücken, aber gewünschte Funktionalität erhalten. In dieser Arbeit wurden zwei Methoden (GAMCS und PSOMCS) entwickelt, die es erlauben optimale cMCSs vorherzusagen.

GAMCS verwendet einen genetischen Algorithmus zur Optimierung der Partitionierung der elementaren Flussmoden, aus denen im Anschluss die cMCSs bestimmt werden.

PSOMCS berechnet cMCSs direkt und verwendet danach Partikelschwarmoptimierung, um den besten cMCS zu finden.

Beide Algorithmen wurden an drei unterschiedlich großen metabolischen Netzwerken von E . coli mit drei unterschiedlichen Optimierungszielen getestet. Es konnte festgestellt werden, dass mit GAMCS eine leichte Performance-Verbesserung gegenüber bereits vorhandenen Methoden erreicht werden kann, die mit PSOMCS noch einmal deutlich gesteigert werden konnte. Im Gegensatz zu GAMCS ist PSOMCS sogar auf Genom-weite metabolische Netzwerke anwendbar. Insbesondere konnten mit PSOMCS erfolgreich optimale Interventionsstrategien in iAF1260, einem Genom-weiten metabolischen Netzwerk von E. coli, vorhersagt werden. Vergleichbare Ergebnisse konnten mit aktuellen state-of-the-art Methoden nicht erreicht werden. Auch im Vergleich mit Literaturdaten lieferten die beiden Methoden korrekte Vorhersagen.

Ein großer Vorteil der neu entwickelten Methoden liegt darin, dass auch komplexe, nicht-lineare Optimierungsziele einfach implementiert werden können.

Schlagwörter: Systembiologie, Stoffwechselnetze, Elementarflussmoden (EFMs), Minimal Cut Sets (MCS), Dehnungsoptimierung, Knockouts

Acknowledgements

A lot of people have helped me, guided me and provided support during my time as a graduate student. Their contribution has been invaluable in helping me achieve my goal and here I would like to thank all of them.

I will always be grateful to my advisor Prof. Diethard Mattanovich for agreeing to supervise my PhD thesis. A heartfelt thanks to Jürgen Zahghellini for accepting me into his group and letting me work on interesting projects. Because of you, I was introduced to the exciting area of stoichiometric analysis of metabolic networks. Your experience and guidance have been invaluable in getting my research published. I cannot thank Christian Jungreuthmayer enough for his patience and knowledge with which he answered many of my questions related to programming and mathematics. You taught me the importance of time and memory in a program. Thanks also to Michael Hanscho for always being there to answer my questions. I still remember times where without your insight, I would have remained stuck with a problem for much longer. Christian and Michi, you guys inspire me to be a better programmer. Thanks also to everyone in the Junior Group Metabolic Modeling, David Ruckerbauer, Matthias Gerstl, Sarah Galleguillos and David Pena Navarro for the many stimulating talks and discussions. You guys have improved my knowledge and have often times provided a different perspective on things.

I have to also thank all the people who volunteer their time to answer questions on online forums. I was almost always able to find answers to technical problems by just "googling" it. Special thanks to the contributors on Stackoverflow and the monks in the Monastery. Thanks for keeping the flame of knowledge burning bright!

Much of my work has been done using free software. To all the people working on open software, I applaud your selflessness in making such powerful and amazing tools freely available. You guys really make the world a better place!

The role played by art in our lives is often understated. My work was less stressful and evenings more enjoyable because of the amazing work of the Beatles. Thank you for the music!

A huge shout out to all my friends in Vienna who have made my life here over the past few years so much more "livable". Anton, Flo and Hansi, the value of your camaraderie cannot be put into words. Over this time I was also fortunate to meet many interesting people around Austria who were generous, kind and helpful. I hope that I was able to enrich your lives as much as you have enriched mine.

Many thanks to my parents Muralidharan Nair and Remadevi Nair, who have been supportive of all my endeavors.

Most of all, I would like to thank the love of my life, Kalindi, for being with me through thick and thin.

Govind Nair
Vienna, March 2017

If a machine is expected to be infallible, it cannot also be intelligent.

- A. M. Turing, , Lecture to The London Mathematical Society on 20 February 1947

Contents

Chapter 1

Introduction

1.1 Motivation

Models of metabolic networks are being increasingly used to predict manipulations in industrially relevant organisms. Such manipulations are aimed at optimizing the production of desired chemicals. A commonly used manipulation strategy is through gene/reaction knockouts. Current tools for calculating intervention strategies in metabolic networks suffer from some serious limitations. These include one or more of the following - inability to account for alternative optimal pathways, inability to guarantee that the predicted knockouts are minimal, inability to deal with larger networks in a short amount of time and the need to manually specify desired yields of metabolites of interest. Methods overcoming these limitations need to be developed to make knockout prediction tools practically relevant.

1.2 Background

Cellular production systems like bacteria, yeast as well as insect and mammalian cells are being used to produce a range of chemicals from electrons in biofuel cells [Liu *et al.* 2004], biofuels and pharmaceuticals [Causey *et al.* 2004, Misawa *et al.* 1991, Lee & Schmidt-Dannert 2002, Martin *et al.* 2003, Nakamura & Whited 2003, Báez-Viveros *et al.* 2004, Lee *et al.* 2008, Chen & Nielsen 2013, Stephanopoulos 2012], to simple/complex fine chemicals and many other molecules [Chotani *et al.* 2000]. This follows an increasing trend which aims to replace chemical synthesis with biotechnological production techniques owing to the benefits of environmentally friendly processing and sustainability of the latter. Metabolic engineering guided bio-based production has led to the industrial-scale production of artemisinin, omega-3 eicosapentaeonic acid, and 1,4-butanediol [Paddon & Keasling 2014, Xue *et al.* 2013, Yim *et al.* 2011].

Cellular production systems can thus be viewed as microbial factories. Unfortunately most of such naturally occurring systems are not designed to produce such chemicals at desired yields. This is because naturally occurring

organisms have evolved to survive and reproduce in their specific environments and not to satisfy human needs. Hence such naturally evolved systems have to be modified to support industrial chemical production needs. Such modifications have indeed come a long way since the days of random mutagenisis and screening, thanks mainly to the advent of genetic engineering and increasing knowledge about the working of biological systems. Engineering microorganisms for overproducing chemicals is an important challenge in biotechnology [Stephanopoulos *et al.* 1998]. This endeavor has been placed on a rational footing by systems biology [Kitano 2002] which combines biological experiments with mathematical modeling and computer simulations [Di Ventura *et al.* 2006]. Systems biology studies complex biological networks involving many components and their interactions at different biological levels. Methods from systems biology are also used in the field of synthetic biology to design and predict the behavior of assembled parts [Chandran *et al.* 2009, Ellis *et al.* 2009, Purnick & Weiss 2009, Smolke & Silver 2011]. Cellular biological networks can be roughly divided into signal transduction, gene regulatory and metabolic networks. The observed cellular behavior is a result of the interaction of these networks. Signal transduction networks allow cells to monitor and respond to changes in their environment like nutrient availability, presence of pathogens, damage to cellular components, etc. Information from various signals are detected, amplified, integrated and transmitted through a series of reactions, in response to which transcription factors regulating the expression of specific genes will be activated. Regulation of genes coding for enzymes will result in changes in the concentration of the corresponding enzymes. This change in enzyme concentrations may lead to changes in the rate at which metabolic reactions are catalyzed. Regulatory networks consist of the genes, regulatory proteins and their interactions. Metabolic networks are made up of all the metabolites and reactions involved in cellular metabolism. Since all biologically produced chemicals are the result of metabolic output, the ability to understand and manipulate metabolic networks is a very important need of biotechnology. Among the three biological network types, metabolic networks have been the most studied. The first genome-scale metabolic model produced was that of *Haemophilus influenzae* [Fleischmann *et al.* 1995]. Since then the metabolic networks of many different organisms have been mapped and experimentally verified [Feist *et al.* 2009, Broddrick 2017]. Detailed knowledge of signaling and regulatory networks is also important and its application to biotechnology has been steadily growing [Papin *et al.* 2005, Hecker *et al.* 2009]. This transition of biology from a descriptive to predictive science is also being enabled by the use of high-throughput technologies like next-generation sequencing [Behjati & Tarpey 2013] and novel analytical tools with incresing

data handling capabilities. Large amounts of data generated by such technologies and also from published literature have been integrated to produce metabolic pathways of many organisms [Feist *et al.* 2007, Förster *et al.* 2003, Schilling *et al.* 2002, Poolman *et al.* 2009, Henry *et al.* 2009].

Given a detailed knowledge of metabolites and their interacting reactions in a biological system, the first step in understanding its metabolic network is by modeling its behavior. Modeling of metabolic networks is done by employing a wide variety of analysis and simulation methods. Particularly successful have been the stoichiometric/structural analysis techniques which use only the reaction stoichiometry and other constraints to model cellular metabolism [Schuster & Hilgetag 1994, Schilling *et al.* 2000, Schuster *et al.* 2002, Price *et al.* 2003].

Metabolic engineering aims to improve chemical production yields in cellular systems. Supporting this aim is an important goal of metabolic modeling. Modeling is used to predict phenotypic outcomes of metabolic manipulations. Conversely, it is also used to predict manipulations resulting in desired behavior. OptKnock [Burgard *et al.* 2003], a *flux balance analysis* (FBA) based method, was one of the first methods proposed for predicting intervention strategies resulting in optimal product yield. This inspired the development of other methods [Tepper & Shlomi 2010, Kim & Reed 2010] aimed at overcoming the limitations of OptKnock as well as accounting for regulation. Many such applications have been reviewed in [Zomorrodi *et al.* 2012]. These methods however fail to account for alternate optimal solutions. The concept of *minimal cut sets* (MCS) and *constrained minimal cut sets* (cMCS) [Klamt & Gilles 2004, Klamt 2006, Hädicke & Klamt 2011], based on *elementary flux modes* (EFM), overcomes this limit but is unable to handle large metabolic networks. A method that will allow for the calculation of MCS/cMCS in large networks without using EFMs was proposed in [Ballerstein *et al.* 2012] and improved upon in [von Kamp & Klamt 2014, Mahadevan *et al.* 2015].

The large size of metabolic networks and the combinatorial nature of interventions means that the corresponding search space is very large, prompting the development of metaheuristic approaches which intelligently navigate through the space to find optimal solutions. These include approaches based on genetic algorithms [Patil *et al.* 2005], evolutionary algorithms and simulated annealing [Rocha *et al.* 2008] and a hybrid bees algorithm [Choon *et al.* 2014], all based on FBA.

There existed no metaheuristic strain design algorithms based on the concept EFMs or the direct enumeration of cMCS. GAMCS, based on EFMs, was designed to quickly find intervention strategies in small and medium-sized networks. PSOMCS, based on the direct enumeration of cMCS was developed to

find optimal intervention strategies in genome-scale metabolic networks.

Finally it should be mentioned that this sophistication in biological understanding has been brought about by the foundation technologies of DNA sequencing [Shendure & Ji 2008], recombinant DNA technology [Baneyx 1999], polymerase chain reaction (PCR) [Mullis *et al.* 1987], etc. The importance of the world wide web with its border-less nature, quick access to information and collaboration-enhancing structure cannot be understated in the ongoing transformation of biotechnology into an engineering-based, rational field.

Chapter 2

Metabolic network modeling and analysis

Cellular metabolism and indeed various other biological functions are the result of many interacting components which calls for a systems based analysis. This is challenging due the extreme complexity of biological systems. Various analysis methods have been developed to overcome this challenge. These include stoichiometric methods based on reaction stoichiometry and other constraints, kinetic modeling methods using detailed kinetics, and hybrid methods.

2.1 Metabolic networks

Metabolism, through a network of interconnected reactions produce energy and many organic compounds called metabolites from simple substrate molecules like sugars. Energy is stored in molecules like ATP and used to fuel various cellular functions. Metabolites may be used to build more complex molecules. Thus understanding metabolism is central to the understanding of cellular behavior. This is particularly important in the areas of industrial biotechnology and biotechnology, where the quality and quantity of many products depends on the underlying metabolism of the organism used. Since metabolism is made up of several interconnected reactions, it's behavior is determined by this network of reactions, including their regulatory aspects. Understanding metabolic networks is thus key to understanding and further manipulating metabolism. On-line repositories contain such network maps of reactions in various organisms or information on chemical molecules and enzymes. These include the BioCyc database collection which includes MetaCyc and BioCyc [Caspi *et al.* 2016], KEGG [Kanehisa *et al.* 2004] and Reactome [Fabregat *et al.* 2016], ERGO^{TM} [ERGO 2017], metaTIGER [metaTIGER 2017, Whitaker *et al.* 2009], ENZYME [ENZYME 2017], BRaunschweig ENzyme Database (BRENDA) [BRENDA 2017, Scheer *et al.* 2010], BioCarta [BioCarta 2017], PubChem [PubChem 2017], Universal Protein Resource (UniProt) [UniProt 2017], Chemical Entities of Biological Interest (CHEBI)

[ChEBI 2017, Degtyarenko *et al.* 2008], ExplorEnz [ExplorEnz 2017, McDonald *et al.* 2009], Integrated relational Enzayme database (IntEnz) [IntEnz 2017, Fleischmann *et al.* 2004], Protein ANalysis THrough Evolutionary Relationships (PANTHER) [PANTHER 2017, Mi *et al.* 2009], MetaNetX [Ganter *et al.* 2013, MetaNetX 2017], BioModels [Le Novere *et al.* 2006, BioModels 2017] and CellML [Lloyd *et al.* 2008, CellML 2017].

Metabolic networks are analyzed by using kinetic data, stoichiometric/structural modeling and hybrid modeling approaches [Tomar & De 2013].

2.2 Kinetic modeling

In kinetic modeling, biochemical reactions are modeled using ordinary differential equations (ODEs) or partial differential equations (PDEs). This requires detailed information on enzyme kinetics in the cellular environment. As such data is limited by the measurement techniques used, construction of kinetic models is not an easy task. Further complicating matters is the presence of large number of heterogenous parameters, complex interactions and its inherent non-linear nature. However, being a detailed representation of reaction dynamics, they are useful and kinetics models of small systems have been built. Kinetic models have been developed for glycolysis [Teusink *et al.* 2000, Smallbone *et al.* 2013], the central carbon metabolism in *E. coli* [Peskov *et al.* 2012, Chassagnole *et al.* 2002] and the central carbon metabolism in red blood cells [Joshi & Palsson 1989]. New methods are being developed with the aim of constructing thermodynamically feasible kinetic models of metabolic networks [Saa & Nielsen 2016]. However the construction of detailed kinetic models from *in vivo* data remains a challenge. The current challenges facing this area have been reviewed in [Vasilakou *et al.* 2016]. Various methods to overcome these challenges and kinetic modeling frameworks that could lead to genome-scale kinetic models have been reviewed in [Srinivasan *et al.* 2015].

2.3 Stoichiometric and structural modeling

Metabolic network function can also be modeled and understood based on the stoichiometry of its constituent reactions. Using a stoichiometric matrix, which contains the stoichiometric coefficients of the metabolites for all the reactions, the structural invariants of the network can be analyzed [Milner 1964]. Stoichiometric network analysis (SNA) [Clarke 1988], based on the concepts of convex geometry has been used to study the robustness of reaction networks.

Such an approach also avoids the problem of measuring reaction fluxes which is necessary for kinetic modeling.

2.3.1 Constraints based approaches

Given a stoichiometric model of a metabolic network, further constraints can be set which limit their behavior. This makes biological sense since cells are always subject to some constraints, like rate of nutrient uptake. Mathematically speaking, the flux space representing possible metabolic network behaviors is reduced by the addition of constraints. Consider the metabolic network shown in Figure 2.1 (redrawn from [Hädicke & Klamt 2011]). A cell boundary lets us distinguish between internal and external metabolites. Internal metabolites have variable concentration while the external ones which serve as source and sink points are assumed to have fixed concentrations. Let the number of internal metabolites be m and the number of reactions (including exchange reactions) be n. $\mathbf{M}$ and $\mathbf{R}$ are the vectors of metabolite and reaction names respectively. The mass conversion of metabolites within such a system can be described by

$$\frac{d}{dt}\mathbf{c} = \mathbf{N} \cdot \mathbf{r} \tag{2.1}$$

where $\mathbf{c}$ is the concentration vector of metabolites and $\mathbf{N}$ is the stoichiometric matrix in $\mathbb{R}^{m \times n}$. An element N_{ij} of $\mathbf{N}$ is the signed stoichiometric coefficient of metabolite i in reaction j. $\mathbf{r}$ is the vector of reaction fluxes in $\mathbb{R}^n$ where $\mathbf{r}_j$ gives the net rate of reaction j. Different biological components operate on different time scales, for example, metabolic reactions are fast compared to regulation. Realistically the concentrations of metabolites in a cell change over time and depend on various factors including cell phase. However, for the sake of simplicity, we will consider metabolite concentrations to remain constant in the long-term. Thus, we assume that the system is in steady-state, giving

$$\mathbf{N} \cdot \mathbf{r} = \mathbf{0}. \tag{2.2}$$

Further, irreversible reactions proceed in only one direction ($\mathbf{r}_{irr}$) while reversible reactions can go in both directions ($\mathbf{r}_{rev}$). Thermodynamic laws dictate that the rate of irreversible reactions be non-negative, that is $\mathbf{r}_{irr} \geq 0$. The space of feasible flux vectors can thus be specified by

$$C = \{\mathbf{r} \in \mathbb{R}^n \mid \mathbf{N} \cdot \mathbf{r} = \mathbf{0} \text{ and } \mathbf{r}_{irr} \geq 0\} \tag{2.3}$$

which is a convex polyhedral cone. All possible flux vectors of $\mathbf{N}$ will lie within this space.

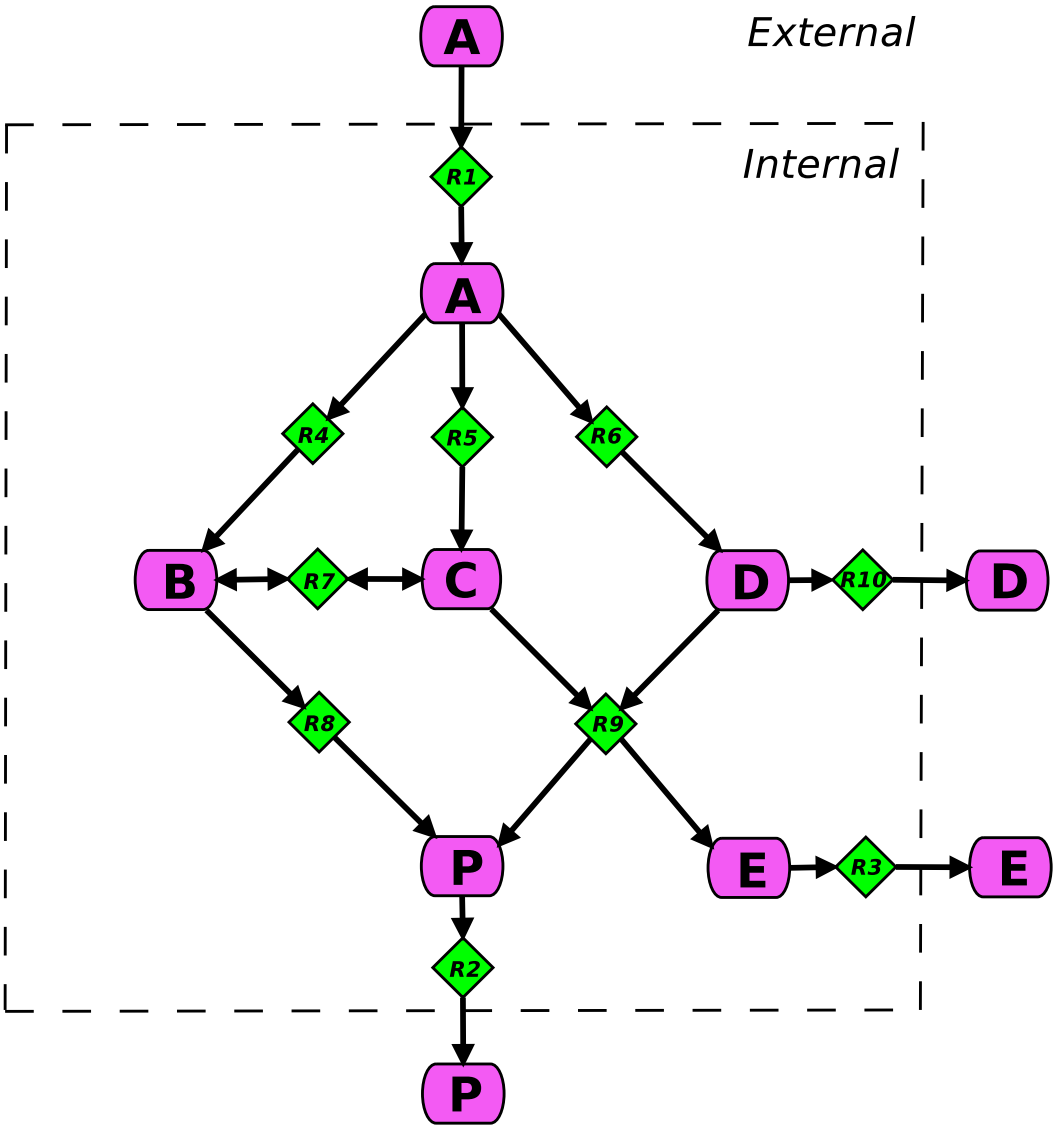


Figure 2.1: Toy network with one uptake reaction (R1) and three products (P, D and E).

Various methods have been developed to analyze this space. These methods can be divided into biased and unbiased methods. Biased methods search for particular solutions in the flux space without considering alternate solutions. Unbiased methods on the other hand take into account all possible solutions.

Many methods using constraint based methods have been developed, some of which are available in the Constraint-based reconstruction and analysis (COBRA) toolbox [Becker *et al.* 2007, Schellenberger *et al.* 2011]. Flux balance analysis (FBA) is the most popular biased approach. A variety of FBA algorithms are available under the Flux-balance Analysis based SIMUlations (FASIMU) [Hoppe *et al.* 2011]. Other popular constraint-based analysis methods include flux variability analysis (FVA) [Mahadevan & Schilling 2003], flux coupling analysis (FCA) [Burgard *et al.* 2004], minimization of metabolic adjustments (MOMA) [Segre *et al.* 2002] and regulatory on-off minimization (ROOM) [Shlomi *et al.* 2005].

It has been shown that network functionality is dependent on its topology [Stelling *et al.* 2002]. The flux space of the metabolic network can be analyzed using methods from convex analysis. A major unbiased approach which characterize the flux space using sets of convex vectors is elementary flux modes (EFM) analysis. Splitting up reversible reactions into two irreversible reactions gives a flux cone in the semi-positive orthant of the flux space, the edges of which were called extreme currents by Clarke [Clarke 1988]. Determining this flux cone without splitting of the irreversible reactions gives EFMs which are the representative flux vectors of this cone [Schuster & Schuster 1993].

The stoichiometric approaches used in this work are briefly explained below.

2.3.2 Flux balance analysis

Flux balance analysis (FBA) which was initially presented in [Papoutsakis 1984, Watson 1984, Watson 1986] and further developed in [Fell & Small 1986, Savinell & Palsson 1992, Varma & Palsson 1993] is a constraint-based method which finds particular solutions according to some optimality criteria assuming that the cellular system is subject to several governing constraints. Firstly, (2.2) provides a set of constraints linking together certain fluxes and restrict the space of possible flux distributions to a subspace of $\mathbb{R}^n$ with each axis representing flux through a single reaction. Irreversible fluxes are constrained to be non-negative, thus

$$r_j \geq 0 \; \forall \; j \in \mathbf{K}, \tag{2.4}$$

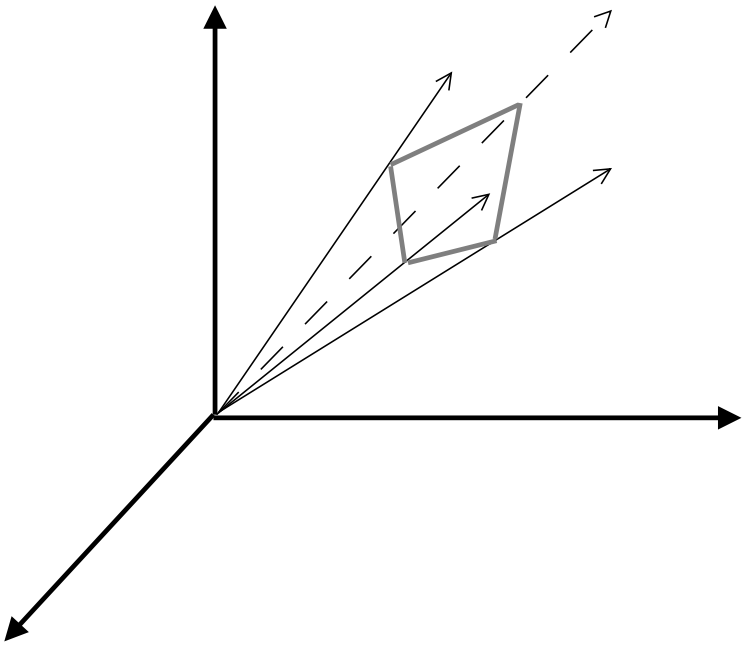


Figure 2.2: A convex polyhedral cone. EFMs are the minimum set of vectors describing this cone. Adapted from [Llaneras & Picó 2008]

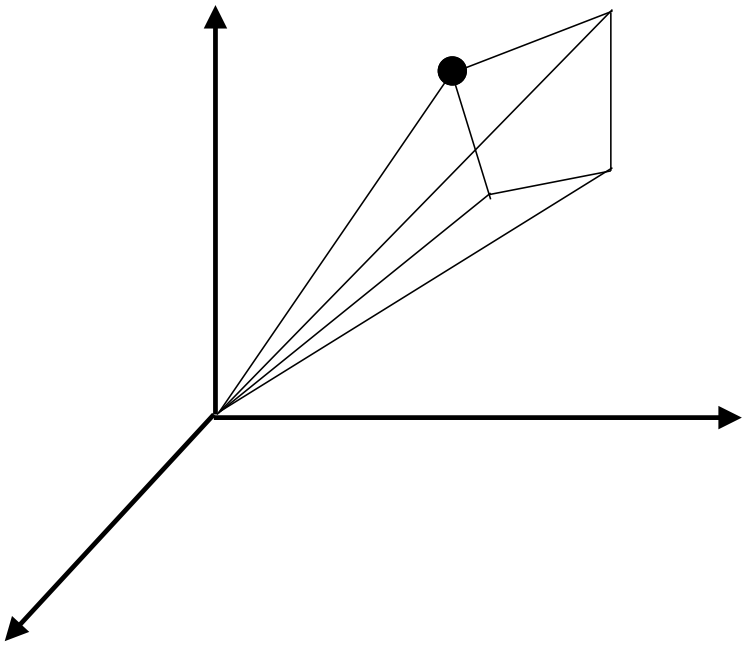


Figure 2.3: A polytope, which is a bounded convex polyhedral cone. The black circle could represent the result of flux maximization using FBA. Adapted from [Llaneras & Picó 2008]

where **K** is the index set of irreversible reactions. This space of flux distributions formed by equations (2.2) and (2.4) is a convex polyhedral cone, Figure 2.2. Additionally, all fluxes are limited by upper and lower bounds,

$$r_{min} \leq r_j \leq r_{max}, \tag{2.5}$$

thereby converting the convex polyhedral cone into a polytope, Figure 2.3. This is a bounded space where optimal solutions under a range of conditions exist. Such solutions can be encoded in an objective function **Z**.

$$\begin{gathered} \text{maximize } \mathbf{Z} = \mathbf{w}^T\mathbf{r} \\ \text{s.t. (2.2), (2.4) and(2.5)} \end{gathered} \tag{2.6}$$

where **w** is a weight vector applied to the flux vector **r**. Solving equation (2.6) will give the optimal flux distribution which will (if the method traverses the vertices of the polytope, see also Section 3.1) be a vertex of the polytope (Figure 2.3). This is a linear programming (LP) formulation which can be readily solved using software packages such as GAMS [Brooke *et al.* 1988], MATLAB [MATLAB 2017], CPLEX [CPLEX 2017], GLPK [GLPK 2017], or metabolic modeling packages like the constrained-based modeling and analysis (COBRA) toolbox [Becker *et al.* 2007, Schellenberger *et al.* 2011]. The objective functions generally used have been energy maximization, product maximization, growth maximization, etc [Palsson 2015]. FBA has used in the following strain engineering applications - ethanol overproduction in *Saccharomyces cerevisiae* [Bro *et al.* 2006, Hjersted *et al.* 2007] and *E. coli* overproducing succinic acid [Lee *et al.* 2005], lactic acid [Fong *et al.* 2005], lycopene [Alper *et al.* 2005a, Alper *et al.* 2005b], L-valine [Park *et al.* 2007], and L-threonine [Lee *et al.* 2007].

Consider the network in Figure 2.1. FBA can be used to answer questions like: given a particular uptake rate for metabolite A (e.g., 10 mmol/gDW/hr), what is the maximum rate at which product P can be produced? This is expressed by the following optimization problem

$$\begin{gathered} \text{maximize R2} \\ \text{s.t.} \\ 0 \leq \text{R1} \leq 10, \end{gathered}$$

solving which gives R2 = 10 mmol/gDW/hr.

A major shortcoming of FBA is its inability to account for alternate optimal solutions which may exist because flux may be redirected through other pathways not involved in the chemical of interest (D and E in Figure 2.1). Also, it is not easy to apply to signaling or gene regulatory networks.

2.3.3 Flux variability analysis

Flux variability analysis (FVA) [Mahadevan & Schilling 2003] was developed to overcome a limitation of FBA, namely its inability to account for alternate optimal solutions. FVA determines the range of flux variability by calculating the maximium as well as minimum of all fluxes for a given cellular objective. Linear programming is used to solve the following two cases.

Case 1:

$$\begin{gathered} \text{maximize } r_i \\ \text{s.t.} \\ \mathbf{0} \leq \mathbf{r} < \mathbf{r}_{max} \\ \forall\, i = 1 \ldots n \\ \mathbf{N} \cdot \mathbf{r} = \mathbf{0} \\ \mathbf{w}^T \mathbf{r} = \mathbf{Z}_{obj}. \end{gathered} \tag{2.7}$$

Case 2:

$$\begin{gathered} \text{minimize } r_i \\ \text{s.t.} \\ \mathbf{0} \leq \mathbf{r} < \mathbf{r}_{max} \\ \forall\, i = 1 \ldots n \\ \mathbf{N} \cdot \mathbf{r} = \mathbf{0} \\ \mathbf{w}^T \mathbf{r} = \mathbf{Z}_{obj}. \end{gathered} \tag{2.8}$$

where $\mathbf{Z}_{obj}$ is the value of the objective function obtained in (2.6). Solving these $2n$ LP problems gives the maximum and minimum flux bounds for each flux r_i under a constant value for the original objective function.

FVA can be used to find the maximum and minimum flux bounds for each reaction in Figure 2.1 as shown in Table 2.1.

2.3.4 Elementary flux modes

As stated before, the space of flux distributions formed by equations (2.2) and (2.4) is a convex polyhedral cone 2.3. Convex analysis shows that this infinite set of steady-state flux distributions can be represented by a finite set of generating vectors. Elementary flux modes (EFMs) are obtained by extending the concept of generating vectors to include irreversible reactions [Schuster & Hilgetag 1994, Schuster *et al.* 2002]. For a flux vector $\mathbf{e}$ to be an EFM, the following three conditions must be satisfied.

i) steady-state constraints, $\mathbf{N} \cdot \mathbf{e} = \mathbf{0}$

ii) thermodynamic constraints, $e_j \geq 0 \; \forall \; j \in \mathbf{K}$

Table 2.1: **Flux variability for the toy network**

Reaction	Minimum	Maximum
R1	9.5	10
R2	9.5	9.5
R3	0	0.5
R4	0	9.5
R5	0	9.5
R6	0	0.5
R7	-9.5	0.5
R8	0	9.5
R9	0	0.5
R10	0	0.5

The maximum and minimum values for all the reactions of the network 2.1, when flux through R2 is constrained at 0.95 of the maximum.

iii) minimality, $supp(\mathbf{r}) \not\subset supp(\mathbf{e})$ for any $\mathbf{r}$ which satisfies i) and ii).

Or, a flux vector is an EFM only if it operates at a steady state, has non-negative fluxes and there exists no other flux vector with a subset of non-zero reaction rates fulfilling these conditions. Biologically speaking, EFMs represent unique non-decomposable pathways in a metabolic network connecting inputs to outputs. The removal of a single reaction from such a pathway will render the entire pathway non-functional. EFMs can also be thought of as minimal sets of enzymes which need to be expressed for the functioning of a particular pathway at a steady state.

EFMs are the generating vectors of the flux cone (2.3). Hence, the convex combination of EFMs spans the entire flux space of a metabolic network [Schuster *et al.* 2002]. Specifically, any feasible flux $\mathbf{r}$ in (2.3) can be represented as a non-negative linear combination of its EFMs.

$$\mathbf{r} = \sum_i \alpha_i \mathbf{e}_i \quad (\alpha_i \geq 0) \tag{2.9}$$

Most algorithms used for calculating EFMs are variants of the double description method [Fukuda & Prodon 1996], with further algorithmic improvements being introduced in [Gagneur & Klamt 2004, Klamt *et al.* 2005, Terzer & Stelling 2008, Von Kamp & Schuster 2006, Terzer & Stelling 2006, Urbanczik & Wagner 2005]. The double description method calculates new EFMs from combining existing EFMs in a pairwise fashion and later verifying that the new EFM candidate has not been previously identi-

fied. This is a time and memory consuming procedure and is further complicated by the observation that the number of EFMs explode with metabolic network size [Klamt & Stelling 2002]. Hence new methods are being developed to overcome this problem. One strategy is to divide the metabolic network into subnetworks by enforcing and suppressing reaction fluxes [Jevremović *et al.* 2011, Hunt *et al.* 2014]. Using regulatory information can also place limits on the number of EFMs calculated [Jungreuthmayer *et al.* 2013c]. tEFMA is a method which calculates only thermodynamically feasible EFMs [Gerstl *et al.* 2015a, Gerstl *et al.* 2015b]. Approaches have also been developed to calculate a subset of EFMs [De Figueiredo *et al.* 2009, Kaleta *et al.* 2009, Pey & Planes 2014, Rezola *et al.* 2011, Quek & Nielsen 2014, Pey *et al.* 2014].

A method to use EFMs for calculating reaction knockouts leading to efficient biomass and energy producing cells was proposed by Carlson and Srienc [Carlson & Srienc 2004, Carlson & Srienc 2004, Trinh *et al.* 2006]. This method has subsequently been used for the overproduction of chemicals of interest [Trinh & Srienc 2009, Unrean *et al.* 2010, Trinh *et al.* 2008, Trinh *et al.* 2011]. This method however uses an iterative procedure where the effect of subsequent knockouts on the desirable properties of the network are calculated. The knockouts so obtained cannot be guaranteed to be minimal.

The EFMs of the network in Figure 2.1, calculated using *efmtool* are shown in Table 2.2 and Figure 2.4. EFMs are alternatively called *elementary modes* (EMs) or just *modes*. In this text, both EFMs and modes will be used depending on the convenience and intelligibility.

Table 2.2: **Elementary flux modes of the toy network**

	R1	**R2**	**R3**	**R4**	**R5**	**R6**	**R7**	**R8**	**R9**	**R10**
$\mathbf{e}_1$	1.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	1.0
$\mathbf{e}_2$	1.0	1.0	0.0	0.0	1.0	0.0	-1.0	1.0	0.0	0.0
$\mathbf{e}_3$	2.0	1.0	1.0	1.0	0.0	1.0	1.0	0.0	1.0	0.0
$\mathbf{e}_4$	2.0	1.0	1.0	0.0	1.0	1.0	0.0	0.0	1.0	0.0
$\mathbf{e}_5$	1.0	1.0	0.0	1.0	0.0	0.0	0.0	1.0	0.0	0.0

The EFMs of the network in Figure 2.1, the corresponding pathways through the network are displayed in Figure 2.4.

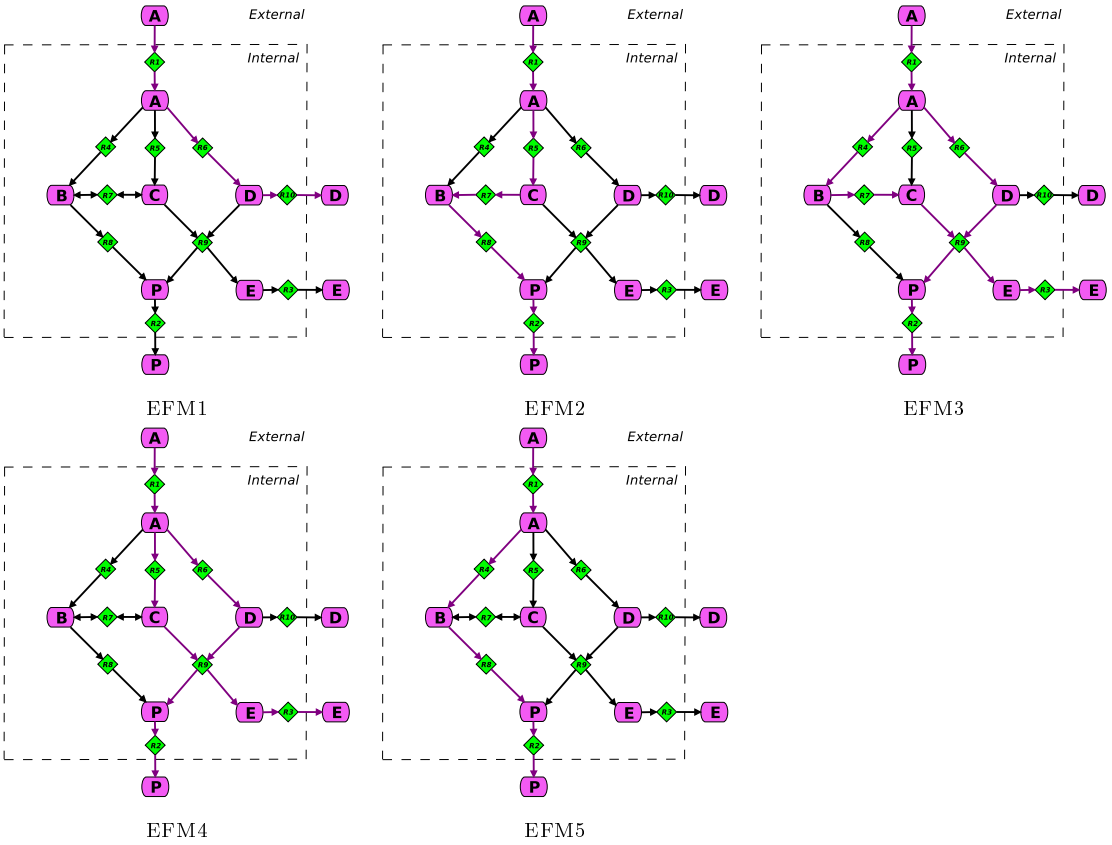


Figure 2.4: EFMs of the network in Figure 2.1 highlighted in purple. These correspond to the ones in Table 2.2.

2.3.5 Minimal cut sets

Since the complete set of EFMs represent all possible phenotypes of an organism, it can be used to identify particular characteristics. For example, a set of EFMs would be linked to the production of a metabolite. Identifying and knocking out the set of reactions common to the set of such EFMs will necessarily block the production of that particular metabolite. A method for finding the minimal set of knockouts, called minimal cut sets (MCS) was initially proposed by Klamt and Gilles [Klamt & Gilles 2004], with a generalized version in [Klamt 2006]. Suppose the entire set of EFMs of a metabolic network are represented by $\mathbf{E}$. Consider a set of $\mathbf{T} \subset \mathbf{E}$ modes representing undesired characteristics, also called target modes which we want to remove from the network. Then, a cutset C is a set of reactions, the removal of which ensures the removal of all $\mathbf{T}$.

$$\forall\, T \in \mathbf{T} : C \cap T \neq \varnothing \tag{2.10}$$

Further, a cutset C is a *minimal cut set* if no proper subset of C is also a cut set. The concept of MCS was initially used to find reactions that bock all flux through a particular reaction. In engineering organisms, there often arises situations where in addition to removing certain characteristics, certain other desirable characteristics need to be preserved, i.e., prevented from being removed from the network. Hädicke and Klamt introduced the concept of *constrained minimal cut sets* (cMCS) which is a generalization of MCS. cMCS account for the necessity to preserve certain desired EFMs while killing others. Given a set of EFMs with desired characteristics $\mathbf{D}$, a cMCS will hit all the target EFMs $\mathbf{T}$ while ensuring survival of at least n EFMs of $\mathbf{D}$. The set of EFMs not hit by a MCS C can be designated by $\mathbf{D}^C$, a subset of $\mathbf{D}$.

$$\mathbf{D}^C = \{D \in \mathbf{D} \mid C \cap D = \varnothing\} \tag{2.11}$$

We also need a minimum number n of $\mathbf{D}$ to survive, hence

$$|\mathbf{D}^C| \geq n \tag{2.12}$$

A cMCS satisfies (2.10), 2.11 and (2.12). There can of course be EFMs which do not fall into either of the desired or target categories. Hence, the sum of $\mathbf{D}$ and $\mathbf{T}$ might not give $\mathbf{E}$.

cMCSs have been calculated using a modified Berge algorithm [Hädicke & Klamt 2011] and also using binary integer program (BIP) [Jungreuthmayer & Zanghellini 2012]. It was shown that the modified Berge algorithm has a better performance compared to BIP [Jungreuthmayer *et al.* 2013b]. The complexity of calculating a MCS was

shown to be NP-hard in [Acuna *et al.* 2009], where also was presented a polynomial approximation algorithm for finding MCS and another algorithm for checking if a given set of reactions constitutes an MCS. A method to find MCS under a boolean model using a integer programming and feedback vector sets (FVS) was proposed in [Tamura *et al.* 2012].

Coming back to the toy network in Figure 2.1, suppose the production of metabolites D and E needs to be blocked. From table 2.2 it can be seen that $\mathbf{e}_1, \mathbf{e}_3$ and $\mathbf{e}_4$ are the modes producing D and E. Killing these modes will block the production of metabolites D and E, i.e., $\mathbf{T} = \{\mathbf{e}_1, \mathbf{e}_3, \mathbf{e}_4\}$. The following MCSs which will achieve this end:

Table 2.3: **MCSs in the toy network**

	MCS
MCS1	R1
MCS2	R6
MCS3	R9 R10
MCS4	R2 R10
MCS5	R3 R10
MCS6	R4 R5 R10
MCS7	R5 R7 R10

Removal of these sets of reactions will completely block the production of metabolites D and E.

Notice that some of the cutsets in Table 2.3 will also lead to the blocking of production of metabolite P. If it is desired to preserve the production of P, the problem becomes one of calculation of cMCS where $\mathbf{T} = \{\mathbf{e}_1, \mathbf{e}_3, \mathbf{e}_4\}$ and $\mathbf{D} = \{\mathbf{e}_2, \mathbf{e}_5\}$ out of which at least one will need to survive. *mhsCalculator* [Jungreuthmayer *et al.* 2013a] returns the following output

Table 2.4: **cMCSs in the toy network**

	MCS
MCS1	R6
MCS2	R9 R10
MCS3	R3 R10
MCS4	R5 R7 R10

Removal of these sets of reactions will completely block the production of metabolites D and E, while ensuring that the production of P is not affected.

2.3.6 Direct enumeration of minimal cut sets

The similarity and interdependency between EFMs and MCSs was noted in [Klamt 2006]. Here it was also shown that given an MCS, the corresponding target EFMs can be obtained. Thus EFMs and MCSs can be converted into each other. An approach based on the Joint-Generation Algorithm [Fredman & Khachiyan 1996] was used to simultaneously calculate both the EFMs as well as the MCS. The insights from these works was brought together leading to the conceptual breakthrough in [Ballerstein *et al.* 2012] where an algorithm to directly calculate MCS without the use of EFMs was presented. This is of great advantage as the extra overhead of generating and using EFMs is overcome . Also to be considered is the explosion in the number of EFMs with network size [Klamt & Stelling 2002]. Ballerstein et. al., use the concept of a *dual network* where MCS can be directly computed given the stoichiometric matrix [Ballerstein *et al.* 2012] and the sets of desired and target fluxes. The methodological breakthrough came by using the system developed in [Ballerstein *et al.* 2012] to calculate shortest MCSs in genome-scale metabolic networks [von Kamp & Klamt 2014].

The set of undesired fluxes for t reactions can be defined by

$$\mathbf{Tr} \leq \mathbf{t} \tag{2.13}$$

where $\mathbf{T} \in \mathbb{R}^{t \times n}$ and $\mathbf{t} \in \mathbb{R}^{t \times 1}$. Similarly, the set of desired fluxes for d reactions can be defined by

$$\mathbf{Dr} \leq \mathbf{d} \tag{2.14}$$

with $\mathbf{D} \in \mathbb{R}^{d \times n}$ and $\mathbf{d} \in \mathbb{R}^{d \times 1}$.

MCS are minimal sets of reactions which when set to zero will satisfy (2.2) and (2.13), i.e., block the target fluxes. cMCS are MCS which additionally satisfy (2.14).

Ballerstein et. al., showed that MCS of the system (2.2) and (2.13) corresponds to the *irreducible inconsistent subsets* **IIS** of the inconsistent system made by combining (2.2) , (2.13) and the equality constraints

$$\mathbf{Ir} = \mathbf{0} \tag{2.15}$$

where $\mathbf{I}$ is an $n \times n$ identity matrix. Such a system is inconsistent because the equality constraints in (2.15) contradict the inequality constraints in (2.13). The IIS of such a system is an inconsistent subsystem which cannot be further reduced into proper inconsistent subsystems. Thus, for finding MCS, all we need to do is to find the IIS of the given inconsistent system. The following system is inconsistent,

$$\begin{pmatrix} \mathbf{N} \\ \mathbf{I} \\ -\bar{\mathbf{I}}_{irr} \\ \mathbf{T} \end{pmatrix} \mathbf{r} \leq \begin{pmatrix} 0 \\ 0 \\ 0 \\ \mathbf{t} \end{pmatrix} \tag{2.16}$$

where the identity matrix $\mathbf{I} \in \mathbb{R}^{n \times n}$ and $\bar{\mathbf{I}}_{irr} \in \mathbb{R}^{n \times |irr|}$ having $\mathbf{0}$ rows corresponding to all $\mathbf{r}_{rev}$.

It was shown by Gleeson and Ryan that the IIS of an inconsistent system can be calculated by enumerating the extreme rays of a particular polyhedron [Gleeson & Ryan 1990]. This polyhedron is obtained by applying Farkas Lemma [Farkas 1902] to the inconsistent system (2.16) to obtain a dual system, which is ensured to be consistent.

Lemma 1 (Farkas Lemma) Given a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ and $\mathbf{b} \in \mathbb{R}^m$, exactly one of the following two statements is true

i) There exists $\mathbf{x}$ such that $\mathbf{Ax} \leq \mathbf{b}$.

ii) There exists $\mathbf{y} \geq 0$ such that $\mathbf{y}^T\mathbf{A} = 0$ and $\mathbf{y}^T\mathbf{b} < 0$.

Which means that given a point $\mathbf{b}$ and a cone $\mathbf{P}$, either $\mathbf{b}$ lies inside $\mathbf{P}$ or there exists a hyperplane passing through the origin separating $\mathbf{b}$ from $\mathbf{P}$. Since (2.16) doesn't exist, we know from Farkas Lemma that (ii) above is true. Hence, the following system is consistent

$$\begin{gathered} \left(\mathbf{N}^T\ \mathbf{I}\ -\bar{\mathbf{I}}_{irr}\ \mathbf{T}^T\right)\mathbf{y} = 0 \\ \left(0\ 0\ 0\ \mathbf{t}^T\right)\mathbf{y} \leq -c \\ \mathbf{y} \geq 0 \\ c > 0. \end{gathered} \tag{2.17}$$

According to Gleeson and Ryan [Gleeson & Ryan 1990], the minimal infeasible subsystems of (2.16) correspond to the vertices of the polyhedron given by (2.17). Ballerstein et. al., show that all MCSs of the system given by (2.2) and (2.13) relate to distinct IISs of (2.16). Hence MCSs can be found by enumerating the vertices of (2.17). Splitting the vector $\mathbf{y}$ into a $\mathbf{u}$ corresponding to $\mathbf{N}^T$, a $\mathbf{v}$ associated with $\mathbf{I}$, $\mathbf{z}$ associated with $\bar{\mathbf{I}}_{irr}$ and $\mathbf{w}$ associated with $\mathbf{T}^T$, we get

$$\begin{gathered} \left(\mathbf{N}^T\ \mathbf{I}\ -\bar{\mathbf{I}}_{irr}\ \mathbf{T}^T\right)\begin{pmatrix} \mathbf{u} \\ \mathbf{v} \\ \mathbf{z} \\ \mathbf{w} \end{pmatrix} = 0 \\ \mathbf{t}^T\mathbf{w} \leq -c \\ \mathbf{u} \in \mathbb{R}^m, \mathbf{v} \in \mathbb{R}^n, \mathbf{w} \in \mathbb{R}^t, \mathbf{z} \in \mathbb{R}^{|irr|}, c \in \mathbb{R} \\ \mathbf{w} \geq 0, \mathbf{z} \geq 0,\ c > 0. \end{gathered} \tag{2.18}$$

IISs and hence MCSs correspond to those vertices of this system with minimal support in $\mathbf{v}$. von Kamp and Klamt [von Kamp & Klamt 2014] restated (2.18) as a mixed integer system. They split the $\mathbf{v}$ variables into positive and negative values, $\mathbf{vp}$ and $\mathbf{vn}$ and attach indicator variables $\mathbf{zp}$ and $\mathbf{zn}$ respectively to these. Also the fact that $\mathbf{z} \geq 0$ is directly incorporated into the system giving

$$\begin{gathered}\begin{pmatrix} \mathbf{N}_{rev}^T & \mathbf{I}_{rev} & -\mathbf{I}_{rev} & \mathbf{T}_{rev}^T & 0 \\ \mathbf{N}_{irr}^T & \mathbf{I}_{irr} & -\mathbf{I}_{irr} & \mathbf{T}_{irr}^T & 0 \end{pmatrix} \times \begin{pmatrix} \mathbf{u} \\ \mathbf{vp} \\ \mathbf{vn} \\ \mathbf{w} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \\ \geq \\ \mathbf{t}^T\mathbf{w} \leq -c \\ \mathbf{u} \in \mathbb{R}^m, \mathbf{vp}, \mathbf{vn} \in \mathbb{R}^n, \mathbf{d} \in \mathbb{R}^d, \mathbf{vp}, \mathbf{vn}, \mathbf{w}, c > 0. \end{gathered} \tag{2.19}$$

The matrices $\mathbf{N}$ and $\mathbf{T}$ have been split into reversible (subscript rev) and irreversible parts (subscript irr). Similarly, for the identity matrix giving $\mathbf{I}_{rev}$ and $\mathbf{I}_{irr}$. cMCS are directly calculated by finding solutions with minimum number of non-zero entries in $\mathbf{vp}, \mathbf{vn}$. This system ((2.19)) was augmented by Mahadevan and Klamt [Mahadevan *et al.* 2015] to include the desired fluxes defined by (2.14), resulting in the following system

$$\begin{gathered}\begin{pmatrix} \mathbf{N}_{rev}^T & \mathbf{I}_{rev} & -\mathbf{I}_{rev} & \mathbf{T}_{rev}^T & 0 \\ \mathbf{N}_{irr}^T & \mathbf{I}_{irr} & -\mathbf{I}_{irr} & \mathbf{T}_{irr}^T & 0 \\ 0 & 0 & 0 & 0 & \mathbf{N} \\ 0 & 0 & 0 & 0 & \mathbf{D} \end{pmatrix} \times \begin{pmatrix} \mathbf{u} \\ \mathbf{vp} \\ \mathbf{vn} \\ \mathbf{w} \\ \mathbf{r} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \mathbf{d} \end{pmatrix} \\ \geq \\ = \\ \leq \\ \mathbf{t}^T\mathbf{w} \leq -c \\ \mathbf{u} \in \mathbb{R}^m, \mathbf{vp}, \mathbf{vn} \in \mathbb{R}^n, \mathbf{d} \in \mathbb{R}^d, \mathbf{vp}, \mathbf{vn}, \mathbf{w}, \mathbf{r}_{irr} \geq 0, c > 0 \end{gathered} \tag{2.20}$$

The binary *indicator* variables $\mathbf{zp}$ and $\mathbf{zn}$ are introduced such that $zp_i = 0$ if $vp_i = 0$ and $zp_i = 1$ *if* $vp_i > 0$ and $zn_i = 0$ if $vn_i = 0$ and $zn_i = 1$ if $vn_i > 0$. Only one of vp_i and vn_i can be active since v_i can be active in only one direction,

$$zp_i + zn_i \leq 1. \tag{2.21}$$

cMCS of the system given by (2.2), (2.13) and (2.14) are the reactions corresponding to positive zp_i,zn_i values obtained after solving the following optimization problem

$$\begin{gathered}\text{minimize} \sum_{i=1}^{n} (zp_i + zn_i) \\ \text{s.t. } (2.20), (2.21) \end{gathered} \tag{2.22}$$

with the additional constraint that the flux through a reaction is turned off if it is part of a cMCS, i.e., $r_i = 0$ if $zp_i = 1 \,||\, zn_i = 1$.

$$\begin{gathered} \{\text{cMCS}\} = \{R_i\} \\ \forall\, i \;\; \text{s.t.} \;\; zp_i = 1 \,||\, zn_i = 1 \\ \text{s.t.} \;\; (2.22) \end{gathered} \tag{2.23}$$

Note that EFMs and MCSs are similar, viz, both are the vertices of polyhedrons. Hence, the same methods could be used for calculating them. Metatool [Von Kamp & Schuster 2006] - a double-description algorithm based method originally developed for calculating EFMs was used in [Ballerstein *et al.* 2012] to find growth-disabling MCSs in a model of *E. coli* metabolism. Following a similar line of thought, the K-shortest algorithm developed for enumerating the shortest EFMs [De Figueiredo *et al.* 2009] was adapted for enumerating MCSs [von Kamp & Klamt 2014]. The dual representation is used in [Tobalina *et al.* 2016] to find MCS involving a specific reaction. They also state that not every vertex of the dual system corresponds to a valid MCS of th primal system. The direct enumeration of cMCS been used in the metabolic engineering of *E. coli* for high yield itaconic acid production [Harder *et al.* 2016].

Coming back to the toy network in Figure 2.1, given an uptake on metabolite given by R1 $\leq$ 10 mmol/gDW/hr, we can specify a design with the following target and desired fluxes,

Desired fluxes

$$0.6\text{R1} - \text{R2} \leq 0.$$

Target fluxes

$$-0.6\text{R1} + \text{R2} \leq 0.$$

which constrains the yield of P to ≥ 0.6. Integrating this into the system (2.20) and solving (2.22) returns the cutsets shown in Table 2.5.

Table 2.5: **Direct enumeration of cMCSs in the toy network**

	MCS
MCS1	R6
MCS2	R9 R10
MCS3	R3 R10
MCS4	R5 R7 R10

Removal of these sets of reactions will ensure that P is produced at a yield ≥ 0.6.

Chapter 3

Mathematical optimization

In metabolic engineering, we seek optimal behavior, like maximum product yield, maximum growth rate, minimal nutrient uptake, etc and also seek optimal ways of achieving such goals. The first step in this is to mathematically model the phenomenon under consideration. In the previous chapter, we model flux in a metabolic network using a system of linear equations at steady state. After having set up the required constraints, the next step is to find the best solution as specified by some well-defined optimization criteria. Mathematical optimization provides the tools and techniques for finding such solutions. This in itself is a huge subject spanning different areas of mathematics. Mathematical optimization problems have the following general form

$$\begin{aligned} &\text{minimize } f_0(\mathbf{x}) \\ &\text{subjext to } f_i(\mathbf{x}) \leq a_i \;\; \forall\, i = 0, \ldots, k \end{aligned} \tag{3.1}$$

where $\mathbf{x} \in \mathbb{R}^n$ is the variable being optimized using the function $f_0 : \mathbb{R}^n \mapsto \mathbb{R}$, also called the *objective function*. The optimization is subject to k constraints specified by the *constraint functions* f_i and the associated bounds b_i for all $i = 0, \ldots, k$. A solution $\hat{\mathbf{x}}$ is the optimal solution if $f_0(\hat{\mathbf{x}}) \leq f_0(\mathbf{y})$ for any $\mathbf{y} \in \mathbb{R}^n$. Mathematical optimization can be divided into different categories depending on the nature of the functions $f_0, \ldots, f_k$. If all the functions are linear, the program is called a *linear program*, otherwise it is a *nonlinear program*.

In the present work, we use the techniques of *linear programming* **LP**, *mixed integer linear programming* **MILP**, *genetic algorithm* **GA** and *particle swarm optimization* **PSO** for which we provide a brief introductions here.

3.1 Linear programming

This is an important subclass of mathematical optimization programs where the objective as well as the constraint functions are linear.

$$\begin{aligned} &\text{maximize } \mathbf{c}^T\mathbf{x} \\ &\text{s. t. } \mathbf{A}\mathbf{x} \leq \mathbf{b} \\ &\mathbf{x} \geq \mathbf{0} \end{aligned} \tag{3.2}$$

where $\mathbf{c}$ and $\mathbf{b}$ are vectors of known constants and $\mathbf{A}$ is a matrix of coefficients. Here the objective function is $\mathbf{c}^T\mathbf{x}$ and the constraints are specified by $\mathbf{Ax} \leq \mathbf{b}$ and $\mathbf{x} \geq \mathbf{0}$.

Many algorithms have been proposed to solve LPs. These include basis exchange algorithms like the *simplex algorithm* by George Dantzig and the *criss-cross algorithm* which find the solution by visiting the vertices of the polytope formed by the feasible region. Interior point methods move through the interior of the feasible region. These include the landmark *ellipsoid algorithm* by Leonid Khachiyan, *projective algorithm* of Karmarkar, *affine scaling* and *path-following algorithms*. Although these algorithms may have similar efficiency in solving general LP problems, some algorithms may be better suited for particular problems. Nowadays large LP problems with thousands of variables can be easily solved on common computers. Many commercial and freely available solvers exist for solving LPs. LPs spring up in diverse situations, from simple ones like scheduling classes for teachers to complex ones like optimizing the national budget. In this work for example, FBA and FVA are LPs. Although a lot of work has been done to improve the efficiency of solving LPs, there exist many open problems which when solved could enhance our ability to solve large LPs. Such open problems relating to strongly polynomial-time solvability of LPs are also of fundamental importance in mathematics [Smale 1998].

3.2 Mixed integer linear programming

If some of the variables in (3.2) are required to have integer values, then the resulting problem is called a mixed integer linear program. In (3.3) C is a set of indices corresponding to the continuous variables $\mathbf{x}_C$ in $\mathbf{x}$ and I is the set of indices on x corresponding to integer variables $\mathbf{x}_I$. $\mathbf{c}_C$ and $\mathbf{c}_I$ are the respective objective coeffecients for the continuous and integer variables. Similarly, the constraint coeffecients are split into $\mathbf{A}_C$ and $\mathbf{A}_I$. If all the variables are required to have integer values ($\mathbf{c}_C, \mathbf{A}$ and $\mathbf{x}_C$ do not exist) then it is an *integer program*. Additionally if the integer values take only $0-1$ values ($\mathbf{x}_I$ is binary), then it is a *binary problem*. But if $\mathbf{c}_C, \mathbf{A}$ and $\mathbf{x}_C$ exist and $\mathbf{x}_I$ is binary, it is a *mixed binary problem*.

$$\begin{aligned} &\text{minimize } \mathbf{c}_C^T\mathbf{x}_C + \mathbf{c}_I^T\mathbf{x}_I \\ &\text{s. t. } \mathbf{A}_C\mathbf{x}_C + \mathbf{A}_I\mathbf{x}_I \leq \mathbf{b} \\ &\mathbf{x}_C, \mathbf{x}_I \geq 0,\ \mathbf{x}_I \text{ integer} \end{aligned} \tag{3.3}$$

These problems are generally NP-hard. Algorithms used for solving problems represented by (3.3) are *branch-and-bound*, *branch-and-cut*, *branch-and-*

price, *cutting-plane-method* and *column generation*. All modern MILP solvers use the *branch-and-bound* (B&B) algorithm or variations of it. Various techniques like tuning of the algorithm parameters to additional processing tools depending on the problem structure have been employed alongside B&B to make NP-hard combinatorial optimization problems more tractable. The state of the art in MILP solvers have been covered in the following review [Lima & Grossmann 2011]. A practical set of ideas for solving difficult MILPs has been presented in [Klotz & Newman 2013]. In this work a MILP is used for the direct enumeration of cMCS in section 7.2.1.

3.3 Metaheuristics

A mathematical optimization problem could also be solved by i) guessing potential solutions and ii) evaluating these solutions based on some measurement criteria. Thus one can sample the solution space till a desirable solution is found.

3.3.1 Genetic algorithms

One intelligent way of sampling the solution space is by using a genetic algorithm (GA). GAs are population-based metaheuristics based on the principles of Darwinian evolution. GAs have been used to solve hard optimization problems, especially where the objective is non-linear or where the solution space is very large. In a GA candidate solutions are encoded in a *chromosome*. A chromosome can be imagined as a string of connected units where each unit corresponds to the value of a particular problem variable. In the most common representation used, the units take a value of 0 or 1 which relates to the corresponding variable being turned off or on.

A GA follows the following basic steps

1. **Initialization** A population of candidate solutions is randomly generated.

2. **Selection** Each candidate solution is evaluated based on some criteria, usually a fitness function which will assign it a *fitness*. Fitness indicates the quality of the solution. The higher or better quality solutions are then selected. Selection can be done using a variety of methods - roulette wheel selection, tournament selection, etc. The basic idea is to make more copies of the better solutions while discarding/eliminating the worse ones. This is akin to the "survival of the fittest" concept.

3. **Variation** This step is intended to change the candidate solutions so that a new area of the solution space is explored. It is hoped that by modifying the best candidate solutions obtained in the previous step, better solutions may be obtained. The commonly used variation operators are *crossover* and *mutation* which as their names indicate are biologically inspired.

4. **Replacement** The newly generated candidate solutions from the previous step become the current population and are subsequently evaluated for their fitness.

5. **Termination** The steps of selection, variation and replacement are iterated until a termination criteria is met. Each iteration is called a *generation*. The termination criteria used can be solutions matching some criteria, fixed number of generations, stability of the best solution over generations or a combination of these.

By following these steps, a GA is expected to evolve towards the optimal solution. However, by design a GA is not guaranteed to find the optimum. Generally, if the objective function is specified well, GAs find near optimal or good enough solutions. GAs may also get stuck at a local optimum. Another issue facing GAs is the number of parameters which need to be adjusted for the GA to function properly. GAs also generally need to run over many hundreds of generations before finding an optimum. Nevertheless they are inherently parallelizable making them suitable for running on multi-cored CPUs and computer clusters. Part of the reason these problems exist is because of a lack of good theoretical understanding of GA behavior. Attempts to theoretically analyze GAs were first attempted using the *schema theorem* in [Holland 1975]. The *building block hypothesis* was proposed in [Golberg 1989] to overcome the limitations of schema theorem and continues to be used today albeit with criticisms . GAs have been successfully used in a number of applications covering a range of fields. For better understanding of the working and implementation of GAs and applications, the interested reader should refer to [Mitchell 1998, Whitley 1994]. A detailed explanation of the GA used in this work can be found in Chapter 6 under 6.2.2.

3.3.2 Particle swarm optimization

It has been observed that naturally occurring swarms, like flocks of birds, schools of fishes, colonies of ants and swarms of bees have a collective behavior and achieve objectives like finding food sources and responding to predators. It is also known that there is no central authority in such natural groups.

Each member of the swarm follows certain simple rules which taken together translate into the observed complex behavior of the swarm. The modeling of such swarm behaviors led to the development of particle swarm optimization (PSO), a powerful optimization method. The basic idea is that particles distributed in the search space of a problem behave like a swarm and collectively move towards the global optimum. The first PSO algorithm was developed by Kennedy and Eberhart [Kennedy & Eberhart 1995] which has since been improved upon by many other workers.

A typical particle is made up of the following three components,

- **Current position** This is a set of coordinates representing a point in the search space. This point will have a corresponding fitness.
- **Previous best position** This is the set of coordinates corresponding to the point with the highest fitness encountered by the particle during its movement. This functions as the particle's memory.
- **Velocity** This component directs the particle to a new position. Understandably its value is a function of the positions of other particles in the swarm, i.e., the knowledge of the swarm.

Additionally, there exists a **Global best position** which is the the set of coordinates corresponding to the point with the highest fitness encountered by the entire swarm. Depending on implementation, this value may be restricted to particles in its immediate neighborhood and not the entire swarm. This leads to the important concept of **swarm topology** which specifies the information a particle has about other particles. In a natural swarm, an entity will be aware of the positions of only its immediate neighbors. Many different topologies have been proposed and have been found to greatly affect the behavior of the swarm and consequently the optimization performance [Poli *et al.* 2007].

A typical PSO algorithm consists of the following steps.

1. **Initialization** The particles are randomly initialized such that they are uniformly distributed throughout the search space.
2. **Evaluation and movement** Particles are evaluated based on a fitness function. Particle **velocity** is next calculated based on the **previous best position** and the **global best position**. The new particle position is a sum of the **current position** and velocity. This step is iterated till a termination condition is met.
3. **Termination** The algorithm stops when a specified termination condition is met, e.g., number of iterations, acceptable fitness value, etc.

In a PSO, the swarm moves cooperatively towards the global optima. The relatively few parameters involved and their simplicity make PSOs easy to implement. However, like metaheuristic methods in general, PSOs are not guaranteed to find the optimum and the swarm may get trapped at a local optima. There is also a deficit in its theoretical understanding. Building a mathematical model of PSO has proven challenging because of the presence of interacting particles which are endowed with memory and the ability to decide the direction of movement, and the stochastic nature of the forces governing its behavior [Poli *et al.* 2007]. It is also critically affected by the structure of the fitness function [Poli *et al.* 2007]. An excellent review of the current theoretical understanding of PSOs can be found in [Bonyadi & Michalewicz 2016]. PSO is a relatively new optimization technique and hence improvements and new applications are being continuously proposed. A survey of PSO applications can be found in [Zhang *et al.* 2015]. These excellent reviews [Poli *et al.* 2007, Banks *et al.* 2007, Banks *et al.* 2008], cover all the basics of particle swarm optimization and should be referred to by anyone interested in it. Working and implementation of the PSO used in this work can be found in Chapter 7 under 7.2.2.

Chapter 4

Results

The aim of this work has been to develop methods for finding metabolic intervention strategies resulting in high product yield. Two metaheuristic algorithms were developed to tackle this problem. GAMCS is a genetic algorithm which explores partitions in the EFM space to find one corresponding to the optimal design. PSOMCS is a particle swarm optimization algorithm which searches the feasible flux space and uses the method of direct enumeration of cMCSs to find the optimal design. Additionally, recognizing the impracticality of a large number of knockouts, both methods place an emphasis on finding smaller sized intervention strategies.

4.1 Prediction of intervention strategies using GAMCS

Here the concepts of EFMs and cMCSs were used for the rational identification of optimal engineering strategies. EFMs are minimal functional building blocks in a metabolic network. EFMs allow one to identify all desired and undesired network states in an organism. Based on this classification, minimal intervention strategies (cMCSs) can be calculated that eliminate all undesired states from the organism but keep at least some of the desired properties. However, even for small sized networks, the possible partitions of EFMs explode combinatorially, making it computationally extremely challenging to find a classification of EFMs which optimizes the engineering objective yet minimizes the number of knockouts required. This problem of sifting through all possible EFM partitions is addressed by applying a genetic algorithm (GA) to quickly find optimal EFM partitions and their corresponding cMCSs. The GAMCS algorithm was implemented in Perl and various tests were carried out to validate its correctness, comparative performance against other strain design methods and its ability to produce good designs. Three different design criteria were considered using models of three different sizes - M1, 5010 EFMs, M2, 38001 EFMs and M3, 429275 EFMs, all obtained from the model used in [Trinh *et al.* 2008].

GAMCS solutions were compared against those obtained using the automatic partitioning method (APM) [Ruckerbauer *et al.* 2014]. AMP enumer-

ates all cMCS leading to an optimal design and hence is designed to find the global optimum. While maximizing for efficiency, GAMCS was able to find the global optimum for M1 and near optimal solutions for M2 and M3, Table 4.1. While maximizing for ethanol yield, GAMCS found the global optima for all the three models, Table 4.2. GAMCS also retrieved many other non optimal solutions during its search path. It was observed that 100% of the lower cardinality cMCSs were retrieved under both cases. This is attributed to the design of the objective function which favored lower cardinality cut sets. In terms of performance, GAMCS found near optimal solutions in 25% and 2.5% of the time taken by APM to find the same solutions for M2 and M3. However, in the smaller model M1, APM outperformed GAMCS. When used to maximize for ethanol yield, GAMCS found the globally optimal solution for all three models. It was also much faster than APM every time. Again, all lower cardinality cMCSs were retrieved, owing possibly to the objective function design. Also, APM is more resource intensive compared to GAMCS.

A fitness function was designed to find EFM partitions which included both the maximum ethanol producing mode plus modes with high efficiency. Its application to all three models produced similar designs, all with a cMCS cardinality of 5. The time taken to find the solutions ranged from a few minutes for M1 and M2 to a few hours for M3. These designs were similar to the ones used in the experimentally verified [Trinh *et al.* 2008].

In conclusion, it can be said that in large metabolic networks GAMCS outperforms alternative approaches by orders of magnitude in terms of runtime. It is naturally and easily parallelized to gain further runtime gains on current computing infrastructure or computer clusters. Moreover, it has no restrictions on the form of the engineering objective, which for the prediction of cMCSs was previously restricted only to linear functions.

As the method deals with the complete set of EFMs of a network, it is limited by the number of EFMs. GAMCS can handle small and medium scale networks. Genome-scale networks are however outside of its scope. Like all metaheuristics, GAMCS cannot guarantee that the global optimum will be found, a fact reflected by its inability to find the most efficient design for M2 and M3. A more detailed analysis of the above can be found in Chapter 6, Section 6.3.

4.2 Prediction of intervention strategies using PSOMCS

Calculation of intervention strategies based on EFMs are limited to small and medium scale networks. On the other hand, strategies based on other

Table 4.1: **Efficiency values GAMCS**

Model	Maximum efficiency	Design efficiency
M1	0.1340	0.1340
M2	0.1542	0.1497
M3	0.1521	0.1374

The designed strains attain efficiency very close to the maximum possible efficiency.

Table 4.2: **Ethanol yield values GAMCS**

Model	Maximum ethanol	Design ethanol
M1	2.00	2.00
M2	2.00	2.00
M3	2.00	2.00

Designs for all the models are capable of producing ethanol at the highest possible yield.

methods like FBA can not guarantee the minimality of the interventions and do not account for alternative optima. This problem was solved by the rigorous approach of directly calculating cMCSs given a specific design [von Kamp & Klamt 2014] which made possible the calculation of intervention strategies even in genome-scale metabolic networks. However, there was a need for a method which can not only find the optimal intervention strategy for a given design but also the best possible design. PSOMCS was developed to address this need. PSOMCS uses particle swarm optimization (PSO) to find optimal cMCSs satisfying multiple design objectives. PSOMCS was written in Perl and the IBM ILOG CPLEX Optimization studio was used to solve the LPs and MILPs. The tests were done on model M3 and the iAF1260 genome-scale model of *E. coli* metabolism. The iAF1260 model was reduced by removing reactions and metabolites so that it was capable of growing anaerobically on glucose as the sole carbon source. The aim was to find designs guaranteeing a high ethanol yield with the additional requirement that the ethanol production be growth-coupled.

Intervention strategies optimizing for the same objective function were calculated using both PSOMCS and GAMCS. Both methods reached the same maximum fitness and returned the same optimal design with the same minimal

ethanol yield and growth rate (see Figure 7.4 in Chapter 7). This design of 5 reaction knockouts is similar to the design presented in [Trinh *et al.* 2008] which has been experimentally verified. Since both methods generally return multiple solutions, multiple cMCSs resulting in the same design were obtained. In terms of performance PSOMCS was over 23 times faster than GAMCS at reaching this optimum.

PSOMCS was used to optimize ethanol production in the genome-scale iAF1260 model. This model has 1413 reactions and 971 metabolites. Two routinely used strain design algorithms, OptKnock [Burgard *et al.* 2003] and RobustKnock [Tepper & Shlomi 2010] were used for comparison. PSOMCS produced a better design than the ones produced by OptKnock and RobustKnock (see Figure 7.5 in Chapter 7). Although the designs produced by OptKnock and RobustKnock allowed for high ethanol yield, the minimum possible yield was 0. In contrast, the design output by PSOMCS guarantees a minimal yield of 0.9. The maximum possible ethanol yield is 2 for all the three methods. In the PSOMCS design ethanol production was also more strongly growth-coupled compared to the other designs. PSOMCS runtime for finding this design was 74 hours. OptKnock and RobustKnock took only a few minutes and just over an hour respectively for their respective designs. Both of these methods require a minimal biomass production to be manually set before running the program. Although OptKnock took a few minutes for all biomass levels tested, RobustKnock ran for over 90 hours with a biomass level of 0.001 and for over 24 hours with a biomass level of 0.005 before being manually terminated. For the purposes of this test, the minimally required biomass production was set to 0.006 for both OptKnock and RobustKnock. PSOMCS on the other hand does not require any manual intervention and finds the optimum solely depending on the fitness function.

This was the first successful demonstration of PSO for strain design. We showed that PSOMCS is orders of magnitude faster than other comparable methods and can calculate optimal designs even in genome scale metabolic networks. Metaheuristic strategies for finding optimal designs using flux balance analysis (FBA) and elementary flux modes (EFMs) exist but this was the first attempt at using the concept of direct calculation of cMCSs to do so.

Although the direct enumeration of cMCS gets around the problem of enumerating EFMs, it is not immune to the issue of memory requirements imposed by larger network sizes. Consequently, this is a limiting factor for PSOMCS too. This memory issue comes into play while solving the MILP represented by (7.5). The search tree constructed by the CPLEX Branch and Cut algorithm can quickly consume a large amount of memory. For example, with a knockout size limit of 6, the search tree produced while solving one of the MILPs exceeded 130 GB.

4.3 Comparison of GAMCS and PSOMCS

Both the GAMCS and PSOMCS algorithms were developed to achieve the same end result. Albeit both being metaheuristic methods, the approach followed by each is completely different from the other. The other similarities between them is that both are network based approaches capable of parallel computational execution and with the ability to produce designs guaranteed minimal product yield. It has been shown that PSOMCS outperforms GAMCS (see Section 7.3 in Chapter 7). The other major differences between the two are highlighted in Table 4.3.

Table 4.3: **Differences between the two approaches presented here**

GAMCS	**PSOMCS**
Optimization done using a genetic algorithm.	Particle swarm algorithm is used for optimization.
Based on the concept of EFMs.	Based on the concept of direct enumeration of cMCS.
Works only on small and medium-scale networks.	Capable of handling all network sizes including genome-scale networks.
Slower than PSOMCS.	Orders of magnitude faster than GAMCS.
Multiple solutions (cMCS) are retrieved for each individual problem solved.	Each individual problem solved returns only a single solution.
Has 14 parameters.	Has 3 parameters.
Parameter adjustment is difficult and slight changes in them affects the performance of the algorithm.	Algorithm is robust in the face of small changes in parameter values.

PSOMCS was developed with the aim of improving upon the performance of GAMCS and it clearly delivers on this aim.

Chapter 5

Conclusion and outlook

The goal of this work was to develop tools for calculating intervention strategies in metabolic networks leading to optimal production of the chemical of interest. The desired characteristics of the system resulting from such intervention strategies were i) a guaranteed minimal yield of the product ii) growth-coupled product formation iii) achieving this through a minimum number of knockouts. Two metaheuristic algorithms were developed towards fulfilling this goal. The large search space of metabolic intervention strategies along with the combinatorial nature of the problem make metaheuristics an ideal tool for attacking this problem. GAMCS is a genetic algorithm based tool and PSOMCS is based on particle swarm optimization. Using the metabolic network model of *E. coli* and ethanol as the chemical of interest, both GAMCS and PSOMCS were shown to satisfy the aforementioned goal. Additionally, their performance was compared to other methods aiming for similar outcomes. GAMCS was shown to be much faster that APM for larger models. It also offers more flexibility in terms of design. PSOMCS was shown to be orders of magnitude faster than GAMCS while producing better designs compared to standard strain design tools, OptKnock and RobustKnock. PSOMCS also represents an advancement in the size of networks that can be handled.

Knowledge gained from the increasing number of genome-scale sequences and the availability of high-throughput data sets have made it possible to integrate biological knowledge beyond the identification of genes in the genome. Bioinformatic tools and databases can be used to produce protein-protein interaction data, regulatory data and metabolic network reconstructions from the sequence data [Reed *et al.* 2006]. These capabilities are predicted to grow. Not only are metabolic networks of more and more organisms becoming available but the quality and size of existing networks are improving. This means more organisms will be available for industrial exploitation. Tools for calculating intervention strategies will remain important not only for industrial purposes but for enhancing basic understanding of cellular functions. Ideally tools must be scalable in the face of such increasing demands. The time taken for calculating optimal intervention strategies in metabolic networks can be considerably shortened using the methods presented in this work. These methods are also capable of searching for complex designs encoded by flexible/non-linear objective functions.

The main limitation of GAMCS is its dependence on the complete set of EFMs of a metabolic network. One can get around this by reducing the set of EFMs to a relevant subset. Many techniques have been proposed for calculating a smaller set of EFMs [De Figueiredo *et al.* 2009, David & Bockmayr 2014, Jungreuthmayer *et al.* 2013c, Machado *et al.* 2012, Kaleta *et al.* 2009, Gerstl *et al.* 2015a]. Working with a smaller set of EFMs will certainly reduce the time taken by GAMCS to find intervention strategies. It will also make it easier to be applied to larger networks. However the validity of such results have yet to be tested. PSOMCS is mainly limited by the MILP used to calculate cMCS. Particularly, having many variables with possible integer values leads to the MILP solver constructing a large solution tree which not only takes memory but time to parse. Hence, a straightforward approach to improving PSOMCS performance is by allowing only a few reactions to be knocked out i.e., take integer values. in [Mahadevan *et al.* 2015] FAV was used to identify 10 knockable reactions and knockouts up to size 6 were calculated from these. GA based techniques could also be used to automatically find reactions not to be considered for knockouts, thereby reducing the size of the knockable reaction set. PSOMCS may also be extended to incorporate regulatory interventions using the approach presented in [Mahadevan *et al.* 2015].

Most tools for finding metabolic intervention strategies do so given a particular design. This in turn introduces the problem of having to find the best design. The methods presented in this work solve both of these using a single tool. That is, they are capable of not only finding minimal knockouts for a particular design but also of finding the best design. Production at zero-growth is an important development garnering increasing interest [Lange *et al.* 2016, Rebnegger *et al.* 2016]. This lets all the carbon source, barring some for maintenance, to be converted to the product of interest. The tools presented in this work already allow for such designs by considering the entire range of product formation and growth. Furthermore the designs presented ensure a high minimal product yield even at zero-growth.

The calculation of knockouts is based on a strong theoretical foundation of constraint based modeling. Metabolic engineering however also relies on gene over expression. Predicting over expression targets requires understanding of correlations among genes, mRNAs, transcriptional or translational regulations, proteins, and metabolic fluxes [Park *et al.* 2012]. This makes over expression target prediction much more difficult than calculation of knockouts. It also highlights the limited scope of constrained based methods in modeling cell behavior. Although some methods have been proposed for identifying over expression targets [Kim & Reed 2010, Ranganathan *et al.* 2010, Park *et al.* 2012, Jian *et al.* 2016], the theoretical foundations on which these methods are based are not strong enough. Critical breakthroughs are needed

in this area to make computational modeling approaches more useful in metabolic engineering.

Chapter 6

Designing minimal microbial strains of desired functionality using a genetic algorithm

---

This chapter was publish by Govind Nair, Christian Jungreuthmayer, Michael Hanscho and Jürgen Zanghellini in *Algorithms for Molecular Biology* 10:29, 2015, DOI: 10.1186/s13015-015-0060-6. [1]

- **Background** The rational, *in silico* prediction of gene-knockouts to turn organisms into efficient cell factories is an essential and computationally challenging task in metabolic engineering. Elementary flux mode analysis in combination with constraint minimal cut sets is a particularly powerful method to identify optimal engineering targets, which will force an organism into the desired metabolic state. Given an engineering objective, it is theoretically possible, although computationally impractical, to find the best minimal intervention strategies.

- **Results** We developed a genetic algorithm (GA-MCS) to quickly find many (near) optimal intervention strategies while overcoming the above mentioned computational burden. We tested our algorithm on *E. coli* metabolic networks of three different sizes to find intervention strategies satisfying three different engineering objectives.

- **Conclusions** We show that GA-MCS finds all practically relevant targets for any (non)-linear engineering objective. Our algorithm also found solutions comparable to previously published results. We show that for large networks optimal solutions are found within a fraction of the time used for a complete enumeration.

[1] JZ and CJ conceived and designed the study. GN, MH, JZ and CJ designed the algorithm. GN implemented the algorithm, ran the analysis and validated the results. All authors were involved in the analysis of the results and read, reviewed and approved the manuscript.

Background

The availability of high amount of biological data has led to the reconstruction of genome-scale metabolic networks for many organisms [Covert *et al.* 2001, Durot *et al.* 2009, Henry *et al.* 2010, Thiele & Palsson 2010] which can be analysed and probed using mathematical and computational methods [Oberhardt *et al.* 2009, Tenazinha & Vinga 2011]. Prominent among these are constraint based modelling approaches which depend on the stoichiometry of the reactions. These include methods like flux balance analysis, **FBA**, [Orth *et al.* 2010] and elementary flux mode analysis, **EFMA** [Schuster & Hilgetag 1994, Schuster *et al.* 2000]. The major difference between these approaches is that FBA seeks particular flux solutions whereas EFMA seeks to describe the entire flux space by enumerating all its elementary and balanced pathways which are called elementary flux modes, **EFMs**. Thus, the complete set of EFMs describes all possible cellular states. The disadvantage is that enumerating all the EFMs of a metabolic network is computationally very demanding as the number of EFMs explodes with network size [Klamt & Stelling 2002]. However, the ability to enumerate EFMs has been steadily improving [Gagneur & Klamt 2004, Terzer & Stelling 2008, Jungreuthmayer *et al.* 2013c, David & Bockmayr 2014].

An important application of an EFMA is the prediction of gene knockouts to turn wild-type organisms into efficient minimal cell factories [**?**]. The design of efficient cell factories is based on the concept of networks of minimal functionality. These are derived from wildtype metabolic networks by keeping typically very few, specifically selected metabolic functions, e.g., EFMs with high yields of products of interest, while diminishing all other unwanted (wildtype) functionality by appropriately selected gene/reaction knockouts. These interventions channel the available carbon flux towards the product of interest. Based on EFMA the concept of constrained minimal cut sets, **cMCS** can be used to redirect cellular resources towards the product of interest [Hädicke & Klamt 2011]. cMCS are minimal (reaction) knock-out strategies, that disable unwanted EFMs (e.g., low product yield/growth) while the desired EFMs (e.g., high product yield) are preserved. In particular, cMCSs of minimal cardinality are important as these solutions minimize the experimental effort when knockouts are actually implemented *in vivo*. Several methods for the computation of cMCS based on a given EFM spectrum are known [Hädicke & Klamt 2011, Jungreuthmayer & Zanghellini 2012, Jungreuthmayer *et al.* 2013b]. Alternatively, cMCS can also be calculated directly without first calculating EFMs [Ballerstein *et al.* 2012, von Kamp & Klamt 2014, Mahadevan *et al.* 2015]. However, in all these methods, explicit design criteria must be used (e.g.

by providing boundaries for the desired minimal product yield). This is problematic in so far as a slight change in the design criteria might lead to large changes in the minimal cardinality of the cMCSs, i.e. the minimal number of required knockouts. For example, Trinh *et al.* [Trinh *et al.* 2008] optimized *E. Coli* for ethanol production with seven reaction knockouts. Jungreuthmayer *et al.* [Jungreuthmayer *et al.* 2013d] on the other hand, were able to design a strain with identical key features and almost identical overall functionality, which required only five reaction knockouts.

If the EFMs are known it is theoretically possible but generally impractical to find all optimal partitions of EFMs and their corresponding cMCSs (of minimal cardinality). In a recent work Ruckerbauer *et al.* [Ruckerbauer *et al.* 2014] approach this problem by first finding the smallest possible cMCS which contributes towards the engineering objective. Then cMCSs of higher cardinality are successively enumerated such that the engineering objective value is greater than or equal to that of the previous smaller cMCS. This circumvents the problem of large number of binning possibilities but will work, in a reasonable amount of time, only for small scale networks.

Here we present a novel approach which uses a genetic algorithm, **GA** to "evolve" near optimal solutions from starting sets of randomly partitioned modes. This results in minimal strains such that only that fraction of the total EFMs which contribute towards the design objective are active after deletion of the predicted cMCSs. This approach combines the simplicity of a GA with the power of EFMA and cMCS. The GA not only circumvents the manual partitioning of EFMs but also finds increasingly better solutions in a relatively short amount of time. This method can be used to satisfy not only traditional design objectives like product yield and growth but can also incorporate more complex design objectives like high growth-coupled product yield using minimal number of knockouts or even non-linear objectives.

6.1 Preliminaries

6.1.1 Elementary flux modes, EFMs.

The material balances in a metabolic network with m internal metabolites and r reactions in steady state can be represented by

$$\mathbf{N} \cdot \mathbf{v} = 0. \tag{6.1}$$

where $\mathbf{N}$ is the $m \times r$ stoichiometric matrix and $\mathbf{v}$ is a flux vector containing the fluxes through the network and $\mathbf{v} \in \mathbb{R}^r$, i.e., $\mathbf{v} = (v_1, \ldots, v_r)^T$. The set of reactions can be partitioned based on thermodynamic constraints into sets

of reversible and irreversible reactions. If *Irrev* is the index set of irreversible reactions,

$$v_j \geq 0 \quad \forall\, j \in \mathit{Irrev}. \tag{6.2}$$

The support of the flux vector $\mathbf{v}$ can be defined as $\text{supp}(\mathbf{v}) = \{j | v_j \neq 0\}$, which is the set of reaction indices in $\mathbf{v}$ with non-zero flux values. An EFM, $\mathbf{e}$, is a flux vector $\mathbf{v} \neq \mathbf{0}$ which satisfies (6.1), (6.2) and a non-decomposability condition which states that, there is no non-trivial flux vector $\mathbf{w}$ satisfying (6.1), (6.2) and whose support is a proper subset of $\mathbf{e}$, i.e., $\text{supp}(\mathbf{w}) \subset \text{supp}(\mathbf{e})$. The non-decomposability condition means that the removal of any supporting reaction in an EFM will block a steady state flux through it. The set of all EFMs of a network completely describes the entire metabolic capabilities of the network. Every possible flux through the network can be expressed as a non-negative weighted combination of EFMs without cancellation. This means that if the flux through a reaction is 0, then all the contributing EFMs necessarily will have 0 flux through that reaction. For more information on EFMs, see [Zanghellini *et al.* 2013].

We will use the following notation henceforth, $E = \text{supp}(\mathbf{e})$. Let $\mathbf{E} = \{E_1, \ldots, E_n\}$ represent the full set of all n EFMs in support notation.

6.1.2 Constrained minimal cutsets, cMCSs.

Suppose there are certain network states which need to be suppressed. These states can be represented by a set of EFMs $\mathbf{T}$, where $\mathbf{T} \subset \mathbf{E}$. The problem then becomes one of "killing" all the EFMs in $\mathbf{T}$. This can be done by "knocking-out" a cutset C of reactions which will "hit" all of $\mathbf{T}$. That is,

$$\forall\, T \in \mathbf{T},\ C \cap T \neq \emptyset, \tag{6.3}$$

C will be a minimal cut set, **MCS**, if there is no proper subset $B \subset C$ which satisfies (6.3) [Klamt & Gilles 2004].

Suppose that in addition to network states which need to be suppressed, there are certain states which we need to preserve when knockouts are applied (e.g. biomass production and product formation). This can be done using the concept of cMCS [Hädicke & Klamt 2011]. The set of desired EFMs $\mathbf{D}$ corresponds to the network states to be preserved. Since in general it cannot be expected that an MCS will not hit any of $\mathbf{D}$, we will say that we would like to have at least k EFMs untouched by an MCS where $k \leq |\ \mathbf{D}\ |$. Given an MCS C, let the set of EFMs $\mathbf{D}^C$ represent $D \in \mathbf{D}$ which survive after applying C,

$$\mathbf{D}^C = \{D \in \mathbf{D} \mid C \cap D = \emptyset\}. \tag{6.4}$$

An MCS which satisfies (6.3) and the following constraint is a cMCS

$$|\mathbf{D}^C| \geq k. \tag{6.5}$$

Thus an intervention problem

$$I = I(\mathbf{T}, \mathbf{D}, k) \tag{6.6}$$

is defined by a set of target EFMs $\mathbf{T}$ which need to be "killed" and a set of desired modes $\mathbf{D}$ of which at least k have to be "kept". Several methods to solve (6.6) are available [Hädicke & Klamt 2011, Jungreuthmayer & Zanghellini 2012, Jungreuthmayer *et al.* 2013b]. Note that $\mathbf{D} \cup \mathbf{T}$ does not necessarily unite to the full set of EFMs since there could be EFMs which we do not want to either kill or keep but instead have a "don't care" status. However, we do not need to specify such an association since we will not operate on these EFMs. We will operate only on the EFMs we are interested in ($\mathbf{D}$ & $\mathbf{T}$) and do not bother with what happens to the EFMs with "don't care" status because by definition it wouldn't matter to us if these EFMs survive or are killed.

In the following we describe a GA to solve the intervention problem (6). For simplicity our implementation partitions the complete set of EFMs into $\mathbf{D}$ and $\mathbf{T}$ and does not make use of the "don't care" option.

6.2 Methods

6.2.1 The EFM kill/keep problem.

Equation (6.6) allows to search for cMCS which keep certain EFMs and kill others. However, it is not intuitive which EFMs to keep and which to kill in order to minimize the cardinality of the cMCSs. Thus the question arises: What is the best partitioning of EFMs in order to reach a specific engineering objective? Even in a modest sized network, the possible combination of EFMs to keep or kill is very large. For example, in a small scale network with 5,000 EFMs, the number of possible kill/keep combinations is $2^{5,000}$. It is practically impossible to explore all points in such a large solution space. Therefore, it makes sense to utilize a program that finds the best set of EFMs to keep, and the corresponding cMCSs which will achieve this for a given an engineering objective [Ruckerbauer *et al.* 2014]. We do this using a GA, the working of which is described below.

6.2.2 The Genetic algorithm, GA.

GAs are heuristics inspired by the theory of evolution, generally used when the extreme of the function cannot be analytically established or when it is impractical to search the whole solution space. GAs work on problems by encoding possible solutions into a population of individuals. These individuals are chromosome like data structures which are iteratively refined to "evolve" better solutions by applying strategies inspired by Darwinian evolution [Whitley 1994, Beasley *et al.* 1993, Li & Yunfei 2002, Mitchell 1998]. In our implementation each individual represents an intervention problem (6.6).

Given a population size p, we randomly generate individuals $S_i = \{s_i^1, \ldots, s_i^n\}$, $1 \leq i \leq p$, where each element s_i^j of S_i indicates if the EFM E_j is present ($s_i^j = 1$) in the individual S_i or not ($s_i^j = 0$). Thus each individual S_i codes an intervention problem (6.6) with

$$\begin{aligned} &I_i = I_i[\mathbf{T}(S_i), \mathbf{D}(S_i), k(S_i)], \\ &\text{with } \mathbf{T}(S_i) = \{E_j | s_i^j = 0\}, \\ &\qquad \mathbf{D}(S_i) = \{E_j | s_i^j = 1\}, \\ &\qquad k(S_i) = w_k |\mathbf{D}(S_i)| \end{aligned} \qquad 1 \leq i \leq p, 1 \leq j \leq n \tag{6.7}$$

where $w_k \in [0, 1]$ is a freely adjustable GA parameter. s_i^j-values are assigned randomly but we provided for the possibility to pre-process EFMs such that EFMs with desirable characteristics have a higher chance of being 1. For example, suppose a cell is described by the following set of EFMs $\{E_1, \ldots, E_7\}$, where only E_1, E_3 and E_7 support product formation. If we want to optimize for product formation, we clearly do not want to keep the non-producer. So we choose s_i^j such that undesirable states never get selected. In our example possible randomly selected individuals could look like $S_1 = \{1, 0, 1, 0, 0, 0, 1\}$, $S_2 = \{1, 0, 1, 0, 0, 0, 0\}$, etc. while $\{1, 1, 1, 0, 0, 0, 1\}$ would not be generated because it includes E_2 which we want to eliminate. This leads to a significant reduction in the search space. Finally, for each individual S_i, cMCS are calculated using the MHScalculator [Jungreuthmayer *et al.* 2013a].

GAs aim to proceed towards better solutions by evaluating each individual S_i against a fitness function F and selecting the top-performers for procreation. The fitness function reflects the design objective since those are the traits we want to improve. In our implementation individuals are selected for mating using a fitness proportionate selection [Goldberg & Deb 1991]. In addition, we use the concept of "elitism" where a pre-specified percentage of top-performers will propagate into the next generation without any modification as shown in Figure 6.2C. This guarantees that the population's maximum fitness does not decrease. We use crossover, mutation [Beasley *et al.* 1993, Whitley 1994], and random selection based on previous

information about surviving EFMs to produce a new generation of individuals. These mechanisms are explained below.

6.2.2.1 Crossover.

We take two parent individuals, S_1 and S_2, and randomly exchange their elements to create two new offspring S_3 and S_4. We implemented the following three standard types of crossovers. For **1point** crossover, generate a random integer r_c, $1 \leq r_c < n$ for each pair of parents, then the offspring of crossover are $S_3 = \{s_1^1, ...s_1^{r_c}, s_2^{r_c+1}, ...s_2^n\}$ and $S_4 = \{s_2^1, ...s_2^{r_c}, s_1^{r_c+1}, ...s_1^n\}$, (see Figure 6.2A). In **2point** crossover, two random integers r_{c1}, r_{c2}, $1 \leq r_{c1} < r_{c2} < n$ are generated for each pair of parents. The offspring in this scenario are $S_3 = \{s_1^1, ...s_1^{r_{c1}}, s_2^{r_{c1}+1}, ...s_2^{r_{c2}}, s_1^{r_{c2}+1}, ...s_1^n\}$ and $S_4 = \{s_2^1, ...s_2^{r_{c1}}, s_1^{r_{c1}+1}, ...s_1^{r_{c2}}, s_2^{r_{c2}+1}, ...s_2^n\}$. In **uniform** crossover, for each EFM a random number $0 \leq r_u^j < 1$ is generated and the offspring are $S_3 = \{s_1^j$ if $r_u^j < 0.5$ else $s_2^j\}$ and $S_4 = \{s_2^j$ if $r_u^j < 0.5$ else $s_1^j\}$.

6.2.2.2 Mutation.

Given an individual S_1 and a random integer r, $1 \leq r < n$, the mutated individual is $S_2 = \{s_1^i$if $i \neq r$, else $1 - s_1^i\}$. The absolute number of such random integers generated for each individual is given by ρr_m, where r_m is a freely adjustable GA parameter, the mutation rate, $0 \leq r_m < 1$ and ρ the maximum number of EFMs with desirable characteristics, $\rho \leq n$ (see Figure 6.2A).

6.2.2.3 Pattern-based individual generation.

In addition to mutation and crossover we create new individuals based on the fittest patterns. For each individual S, whose corresponding intervention problem has solution(/s), we generate a "design pattern", which contains only the surviving EFMs,

$$P = \{p^j \mid p^j = 1 \text{ if } E_j \in \mathbf{D}^C \text{ else } p^j = 0\}. \tag{6.8}$$

Given a binary individual $S = 1010001$, if only EFM 3 and 7 survive the intervention, the resulting pattern will be 0010001. Thus a pattern is a specific strain design for an intervention problem. A solvable intervention problem typically produces more than one solution. Therefore, one individual will usually have more than one pattern associated with it. Since the fitness depends on the surviving EFMs, each pattern will have its own fitness value. Thus one individual may be associated with more than one fitness value. Here, the fitness of an individual S is defined as the fitness of the fittest pattern P.

To create the new individuals, we start by weighting each EFM proportional to the number of times the EFM survived in all previous patterns. Let $\boldsymbol{P}_t$ represent the entire set of patterns found until a given generation t. The weight w_t^i for an EFM E_i is calculated by

$$w_t^i = \sum_{j=1}^{|\boldsymbol{P}_t|} (\boldsymbol{P}_t)_j^i. \tag{6.9}$$

Next we generate a set of desired candidate EFMs by randomly selecting a random number of EFMs with non-vanishing w_t^i. Out of these desired candidate EFMs new individuals were composed by including those candidate EFMs for which a randomly selected number r_i was not larger than the weight of the corresponding candidate EFM, $0 \leq r_i < \max w_t$ and $\max w_t$ is the maximum of all such weights (see Figure 6.2B),

$$S_{\text{new}} = \{s_{new}^i | \text{ if } w_t^i \geq r_i,\ s_{new}^i = 1 \text{ else } s_{new}^i = 0\}. \tag{6.10}$$

The number of individuals generated by this method can be controlled by the GA parameter 'new_S', Table 6.1. It is a way to consider all good solutions obtained so far and ensures that more EFMs with desirable properties find their way into the set of desired EFMs. This helps the GA to reach the optimum faster.

The GA stops after reaching a pre-specified number of generations or when the maximum fitness doesn't improve for a given number of generations, outputting all MCSs of minimal cardinality associated with each desired pattern. The schematic of the GA implemented and used here is shown in Figure 6.1 along with a small illustrative example in Figure 6.2.

6.2.3 Implementation.

The GA was implemented in Perl `http://www.perl.org/`. cMCSs were calculated with MHScalculator which is an open source C-program that is freely available [Jungreuthmayer *et al.* 2013a]. EFMs were calculated using the *regEfmtool* [Jungreuthmayer *et al.* 2013c]. All runs were performed on a machine with the following specifications - 2 CPUs, 12 cores, Intel Xeon X5650 2.67 GHz and an Ubuntu 14.04 LTS operating system, allowing the used programs to utilise 10 threads in parallel. Caching in form of look-up tables is employed to store previously obtained MCS, patterns and corresponding fitnesses, to avoid repetition of calculation. We also use *tmpfs*, a temporary file storage created on the RAM, for faster i/o on intermediate files. A general description of the parameters used for controlling the GA are shown in Table 6.1. Specific parameter values for the individual runs are shown in Table 6.2.

6.2.4 Validation.

We ran the GA on an *E. coli* core model, M3, [Trinh *et al.* 2008] and two smaller models, M1 and M2, which were derived from the parent model, M3, by removing several reactions. M3 describes the central carbon metabolism of *E. coli* including the uptake and utilization of several hexose and pentose sugars. Compared to M3, M2 is restricted to model only glucose utilization (all other carbon uptake relations were removed). Finally, M1, the smallest model of the three, describes glucose utilization under anaerobic conditions. The main topological properties of the three models are summarized in Table 6.3.

6.3 Results

Our aim is to design optimised *E. coli* strains for ethanol production. The optimization objectives considered in this study were ethanol yield (Y_{Etoh}), substrate specific productivity which is the product of normalised specific ethanol production and normalised biomass production [Feist *et al.* 2010] also called "efficiency" ($\eta_{Etoh} = Y_{Etoh} \times Y_{Biomass}$), and an objective which considers both the yield and efficiency together. In all objectives, we favor solutions with low cardinalities (for details see Table 6.4).

6.3.1 Benchmarking

We tested the performance of the GA against the automatic partitioning method, **APM** developed by Ruckerbauer *et al.* [Ruckerbauer *et al.* 2014] using the models M1, M2 and M3. The APM was selected for comparison, as for any given, linear engineering objective APM enumerates all optimal knockout strategies without requiring any manual interference. We tested for maximum efficiency and ethanol production using the fitness function F_1 and F_2, respectively as given in Table 6.4. For the three models used we listed the main characteristics of the optimal solutions with respect to the fitness functions in Table 6.3. All simulations were run five times. In the following we reported averages over these five runs, unless otherwise stated.

6.3.1.1 Maximizing for efficiency

We used the fitness function F_2 (Table 6.4) with the parameters shown in Table 6.2 to optimize for efficiency. The GA was terminated when the fitness function remained unchanged for 15 generations.

The GA found all optimal solutions for the small model M1 (see Figure 6.3a). In the bigger models M2 and M3 the GA did not find the best solutions but got within 3% and 1.2% of the maximum fitness, respectively.

In M2 and within the selected runtime, the GA mostly found near optimal solutions (see Figure 6.3b), and rarely converged to the optimal solution. In the case of M3 the GA got stuck in a local optimum (see Figure 6.3c).

While the GA does not necessarily identify the absolute best solutions, it generally finds near-optimal solutions extremely quickly. In M2 and M3 near-optimal solutions are found in about 25% and 2.5% of the time taken by the APM, respectively (see Figure 6.3). Only in the small-scale model M1, which is easy to enumerated fully, the GA is slower than the APM.

Comparing the MCSs obtained with the GA to the ones obtained with the APM, as shown in Figure 6.4 a, b and c, reveals that our algorithm retrieves 100% of all low cardinality MCS. The number drops with increasing MCS' cardinality. This behavior is expected as our fitness functions favors low cardinality solutions. Thus it is very unlikely that the GA will identify many high cardinality solutions. In fact, this explains the non-monotonic behavior of the line of maximum efficiency in Figure 6.3c. Because the fitness function F_2 allows for a trade off between cardinality and maximum efficiency, the efficiency might decrease. Yet the fitness function still increases.

6.3.1.2 Maximizing for ethanol production

We used the fitness function F_1 (Table 6.4) with the parameters shown in Table 6.2 to maximise for ethanol yield. The GA was terminated when the fitness function remained unchanged for 15 generations.

Unlike the previous case, here our algorithm found all optimal solutions for all models. Also, we were faster than the APM in reaching the optimum for all models (see Figure 6.3d, e and f).

Again, like in the case of maximising for efficiency, the GA retrieves 100% of lower cardinality MCSs (Figure 6.4 d, e and f), and not many of the higher cardinality solutions, when compared to the solutions obtained using APM. This is a result of the fitness function, F_1, which favors towards lower cardinality MCSs. The effect of this can be observed in Figures 6.3d and e where the GA first finds higher cardinality solutions for the optimal ethanol yield and settles down to the lowest possible cardinality in subsequent generations.

6.3.2 Optimizing for a complex design

Although maximising for ethanol yield and efficiency, produces sub-optimal to optimal designs, these designs may not be the best to implement *in vivo*.

For example, the EFMs which result in the maximum ethanol yield do not support growth. However, two of these EFMs provide maintenance energy. On the other hand, designs with maximum efficiency do not include maximum ethanol producing EFMs. It would be preferable to have a design which combines these features. We used the fitness function F_3 (Table 6.4) with the parameters shown in Table 6.2 to find optimal designs.

A similar problem was looked at in [Ruckerbauer *et al.* 2014] where the authors optimised M1 for efficiency while ensuring that at least one of the maximal ethanol producing EFMs survive in the final design. Their design included the most efficient ethanol producing EFMs as well as EFMs with maximum ethanol yield, achieved with an MCS cardinality of 6. A similar design was used by Trinh *et al.* [Trinh *et al.* 2008] using 7 reaction knockouts. Our algorithm produces designs of similar functionality with MCSs of cardinality 5, Figure 6.5b. Similar results were obtained for M2, and M3 as shown in Figure 6.5d and f respectively, both with MCS cardinalities of 5. Also, our algorithm was very quick in finding these designs, taking a few minutes for M1 and M2 and a few hours for M3.

6.4 Conclusion

We have presented a method for the design of minimal microbial strains of desired functionality. The designs are minimal in the sense that only a few of the total number of pathways (EFMs) are active after deletion of the predicted cMCSs. Our GA uses the MHScalculator [Jungreuthmayer *et al.* 2013a] to find cMCSs for a given set of desired and target EFMs. However, the optimal selection of such sets is non-intuitive. Hence, the aim was finding the best possible set of pathways which maximize a given engineering objective.

Another GA, called the OptGene method has been previously reported which finds reaction cuts to achieve a design objective [Patil *et al.* 2005]. This algorithm works by testing different combinations of reaction knockouts. In contrast, we test partitions of EFMs. Thus our search space is by orders of magnitude larger than theirs. OptGene finds many solutions too, but it cannot be guaranteed that these are minimal. Also, the knockout cardinalities are restricted to 1 - 10. Our approach is based on the concept of EFMs which enumerate all possible network states. OptGene however uses methods like FBA, MOMA [Segre *et al.* 2002], etc. to calculate the fitness which, unlike EFMs do not account for alternative pathways. Although methods which use FBA and MOMA predict optimal solutions, there is no guarantee that the predicted optimum will be achieved. In a similar vein, the method presented here has advantages over other methods which use a biased biological objective

like OptKnock [Burgard *et al.* 2003], RobustKnock [Tepper & Shlomi 2010] and tilting of the objective function [Feist *et al.* 2010].

Boghigian *et al.* [Boghigian *et al.* 2010] also use a GA and EFMs to design strains with higher product yields. Their approach however differs from the method presented in this paper in a few major ways. First, the aim of the GA presented in [Boghigian *et al.* 2010] is to only improve product yields without considering the minimality of the knockouts. Hence, in contrast to us their predicted knockouts are not guaranteed to be minimal. Second, the basic problems considered by both methods are different, although the final aim is the same, namely strain improvement. Boghigian *et al.* look for reaction knockouts which will improve product yields. Our GA not only maximizes the product yield but also simultaneously searches for optimal partitions in the set of EFMs. Finally, we deal with networks where the number of EFMs are one order of magnitude larger than that used in [Boghigian *et al.* 2010].

Tools which use EFMs to find intervention strategies include the MHScalculator [Jungreuthmayer & Zanghellini 2012, Jungreuthmayer *et al.* 2013a] and a tool to calculate cMCSs as part of the *CellNetAnalyzer*, a MATLAB package providing comprehensive structural and functional analysis of biochemical networks [Klamt *et al.* 2007]. These methods use EFMs and hence consider the entire metabolic landscape of the organism. The limitation of these methods is that the EFMs which must survive or be killed by an intervention have to be manually partitioned.

A recent method (APM [Ruckerbauer *et al.* 2014]) overcomes this issue by calculating all partitions of EFMs for MCSs of increasing cardinality such that the objective is higher than that corresponding to the previous smaller MCS size. This is an exhaustive and exact method for finding intervention strategies in metabolic networks. However, this method is impractical for large networks given current computational capabilities. Although the GA is not faster than the APM at very small network sizes like M1, its comparative performance improves with increasing network sizes, Figure 6.3. Also, when optimizing for efficiency, the GA does not reach the global optimum when APM does, Figure 6.3 b,c. Note that however, an exact comparison to APM is not possible since APM tries to find all MCS whereas the GA tries to find the best cut set for a particular objective. Our method also incorporates the freedom to encode complex design criteria, which is not possible with the APM. Also, since the APM is based on linear programming, it is limited to linear objective functions whereas we can implement non-linear objective functions as well.

An important new approach initially proposed by Ballerstein *et al.* [Ballerstein *et al.* 2012] with further improvements in [von Kamp & Klamt 2014, Mahadevan *et al.* 2015] is able to directly find MCSs without first needing to calculate the EFMs by using the concept

of hypergraph dualisation. This gets rid of the problem of explosion in the number of EFMs with increasing network sizes, allowing for prediction of intervention strategies in genome-scale metabolic networks. However, these methods have to specify design criteria like minimal product yield [von Kamp & Klamt 2014]. This is a limitation in that slight changes in the value of the specified design criteria may lead to different MCSs. In contrast, our algorithm tries to automatically find the best design criteria.

The GA implemented here is able to predict numerous good solutions to problems of product maximization which are comparable to experimentally verified designs [Trinh *et al.* 2008]. One advantage of this method is the short time taken while dealing with bigger systems. The biggest advantage though is the flexibility in the selection of the design criteria using the fitness function. The fitness function can be arbitrarily complex to accurately reflect the design criteria. Here it has allowed us to produce good designs without knowing the specific properties of EFMs which need to survive.

Since our approach mainly relies on a GA, it may be affected by inherent limitations of GAs, including the possibility of getting stuck at a local optimum. This may be overcome by employing multiple runs or changing the GA parameters. Note that we have considered reaction knockouts here but this can be easily translated into gene knockouts using gene-reaction associations.

Finally, we provide a brief description of the parameter values used. The mutation rate was set such that only two to four positions in an individual are affected, an increase in this number resulted in the GA not producing any good solutions. Decreasing this number resulted in a slower rate of improvement in fitness (data not shown). It is also possible to completely turn off mutation by setting r_m to 0. In any case the performance of the GA can be improved with pattern-based individual generation rather than relying solely on mutation and crossover. The number of such individuals can be adjusted with the 'new_S' parameter. However, too high 'new_S' values led to a comparatively worse GA performance (data not shown). The parameter w_k specifies the minimum number of EFMs which should survive an intervention. The lesser this value, the higher the probability of finding better solutions - because typically, optimal solutions have very few surviving EFMs, Table 6.3. However, small w_k also produces more solutions which in turn takes more time for pattern and fitness calculations. In order to reach the optimum with as few solutions as possible, we found that in general, w_k can be large for small models (e.g., M1) and must decrease for growing models (e.g., M2 and M3) (for exact values see 6.2). 'min_1s' determines the minimum number of possible good EFMs that will end up in the set of desired EFMs **D** in the initial population. Because the EFMs are randomly selected to be in **D**, not all individuals will generate viable solutions. Also, it is important

that the union of **D**s in the whole population nearly covers the set of good EFMs. The EFMs which are not covered must otherwise rely on mutation to be transferred from **T** to **D**. The probability of this happening decreases with increasing individual size. Hence, 'min_1s' was set to a high value of 0.9 for all of the runs. A future direction of this work would be to study the effect of these parameters in detail. This will help get rid of the empirical setting of parameters in our GA and allow for the implementation of a protocol to automatically determine these values during the running of the GA.

In summary, our algorithm is able to quickly find (near) optimal intervention strategies satisfying non-linear engineering objectives in large metabolic networks. However, EFMs are still necessary for our method which is a significant bottleneck when it comes to genome-scale networks. We expect that combining the dual method [Ballerstein *et al.* 2012, von Kamp & Klamt 2014, Mahadevan *et al.* 2015], which will allow for the calculation of cMCS directly from the stoichiometric matrix, with a GA will overcome this hurdle.

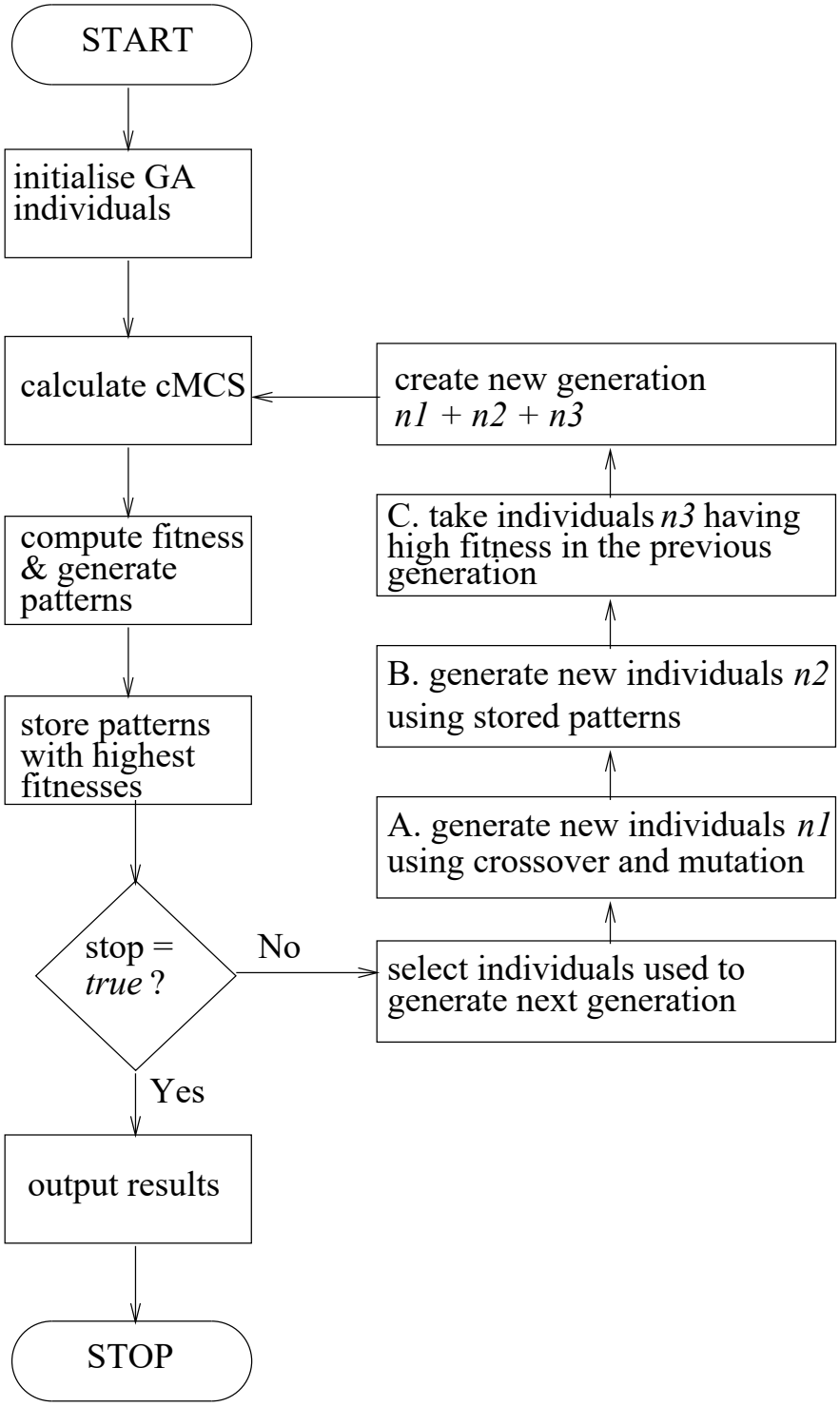
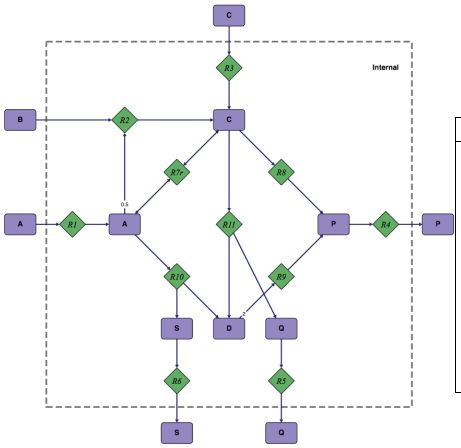


Figure 6.1: Flowchart of the GA. The GA stops when a stopping condition is met, which here is if the number of generations reaches a pre-specified maximum or if the maximum fitness remains unchanged for a pre-specified number of generations (Table 6.1).



Complete EFMs of the network

	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11
E_1	0.0	1.0	0.0	0.25	0.0	0.5	1.0	0.0	0.25	0.5	0.0
E_2	0.0	1.0	0.0	0.5	0.0	0.0	0.5	0.5	0.0	0.0	0.0
E_3	0.0	0.0	1.0	0.5	0.0	1.0	1.0	0.0	0.5	1.0	0.0
E_4	0.5	1.0	0.0	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
E_5	1.0	0.0	0.0	1.0	0.0	0.0	-1.0	1.0	0.0	0.0	0.0
E_6	0.0	0.0	1.0	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
E_7	0.0	1.0	0.0	0.25	0.5	0.0	0.5	0.0	0.25	0.0	0.5
E_8	0.5	1.0	0.0	0.5	1.0	0.0	0.0	0.0	0.5	0.0	1.0
E_9	0.0	0.0	1.0	0.5	1.0	0.0	0.0	0.0	0.5	0.0	1.0
E_{10}	1.0	0.0	0.0	0.5	1.0	0.0	-1.0	0.0	0.5	0.0	1.0
E_{11}	1.0	0.0	0.0	0.5	0.0	1.0	0.0	0.0	0.5	1.0	0.0

Initial population and results

i	$E_i \in \mathbf{D}$	S_i	C_i	P_i	$Y_{R4,i}$	$\|C_i\|$	Fit_i
1	E_2, E_5, E_7, E_{11}	01001010001	R2 R3 R9	00001000000	1	3	**1.73**
2	E_4, E_5, E_8, E_9	00011001100	R3 R7 R9	00010000000	1	3	**1.73**
3	E_2, E_3, E_4, E_5, E_8, E_9, E_{11}	01111001101	R3 R7	00010001001	0.5	2	**1.32**
4	E_2, E_3, E_4, E_5, E_6, E_9, E_{11}	01111100101	R9	01011100000	0.5	1	**1.4**

EFMs are randomly selected encoding GA individuals S_i such that a 1 & 0 indicates inclusion of the corresponding EFM in **D** & **T** respectively. Searching for cMCS such that at least one EFM of **D** survives results in patterns P_i. $Y_{R4,i}$ is the least value corresponding to $R4$ in the surviving EFMs. $Fit_i = Y_{R4,i} + 1 - (|C_i|/|n|)$.

Creating second generation individuals

A	RWS		crossover	mutation	$n1$
	S_1	01001010001	0100101**1100**	00**1**1**1**001100	00111001100 S_{1new}
	S_2	00011001100	**00011000**0001	0**1**011**1**0**1**100	01011001100 S_{2new}
B	patterns				$n2$
	00001000000				
	00010000000				
	00010001001				
	01011100000				
	w_t	01032101001			00011100000 S_{3new}
C	Fittest individuals				$n3$
	01001010001				01001010001 S_{4new}
	00011001100				

$n1$ is generated by randomly selecting from S_i based on F and subjecting these to GA operations. $n2$ is generated by randomly selecting EFMs based on w_t, which represents survival of corresponding EFMs in the previous generations. $n3$ is for elitism. **A**, **B** and **C** correspond to sections in the flowchart in Figure 1 with the same names.

Second generation and results

i	$E_i \in \mathbf{D}$	S_i	C_i	P_i	$Y_{R4,i}$	$\|C_i\|$	Fit_i
1*new*	E_3, E_4, E_5, E_8, E_9	00111001100	R3 R7 R9	00010000000	1	3	**1.73**
2*new*	E_2, E_4, E_5, E_8, E_9	01011001100	R3 R9	01011000000	0.5	2	**1.32**
3*new*	E_4, E_5, E_6	00011100000	R7 R9	00010100000	1	2	**1.81**
4*new*	E_2, E_5, E_7, E_{11}	01001010001	R2 R3 R9	00001000000	1	3	**1.73**

Figure 6.2: GA example. Running the GA on the given toy network of 11 EFMs with the aim of maximizing production of P. The initial individuals S_i and the effect of applying the mutation, crossover and elitism operators to generate new individuals are shown. Here the GA finds the best solution with a fitness of 1.81 and yield (Y_{R4}) of 1 in the second generation.

Table 6.1: **The GA parameters**

No:	GA parameter	Description
1	t	This parameter is used to specify the number of generations for which the GA will run.
2	p	This parameter is used to specify the number of individuals S present in one generation of the GA.
3	r_m	This parameter is used to set the mutation rate which specifies the number of bits in an individual S that will be flipped from 0 to 1 or vice versa.
4	cross	This parameter is used to select among the three types of crossover operations possible here: 1point, 2point and uniform.
5	elit	This parameter is used to specify the fraction of the number of total individuals from the previous generation which will be retained in the subsequent generation.
6	new_S	This parameter specifies the number of new individuals which will be generated in each generation, based upon information from previous generations.
7	t_stop	This parameter is used to set the maximum number of generations after which the GA terminates if the maximum fitness remains unchanging.
8	min_1s	This parameter specifies the fraction of maximum number of possible good modes which must be present in the initial population.
8	w_k	This parameter is used by the MHSCalculator to specify the minimum number of EFMs which have to survive in given a set of desired modes **D** (provided as fraction of the number of EFMs in **D**).
9	threads	This parameter specifies the maximum number of threads to be used by the program.

These parameters are used to control the running of the GA and also to get more specific results.

Table 6.2: **GA parameters for different runs**

GA parameter	**M1 ethanol**	**M1 efficiency**	**M1 complex**	**M2 ethanol**	**M2 efficiency**	**M2 complex**	**M3 ethanol**	**M3 efficiency**	**M3 complex**
w_1	1	0	1	1	0	1	2	0	1
w_2	0	50	50	0	50	50	0	10	50
w_3	1	1	1	1	1	1	1	1	1
w_4	1	1	1	1	1	1	1	1	1
t	100	100	100	100	100	100	100	100	100
p	50	50	50	50	50	50	50	50	50
r_m	0.00025	0.00025	0.00025	0.00025	0.00025	0.00025	0.000025	0.000025	0.000025
cross	1point	1point	1point	1point	1point	1point	1point	1point	1point
elit	0.025	0.025	0.025	0.025	0.025	0.025	0.025	0.025	0.025
w_k	0.03	0.017	0.04	0.025	0.01	0.03	0.01	0.0075	0.03
new_S	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
t_stop	15	15	15	15	15	15	15	15	15
min_1s	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9
threads	10	10	10	10	10	10	10	10	10

Parameters used in the various runs.

Table 6.3: **Features of models used**

Model	**M1**	**M2**	**M3**
model source	[Trinh *et al.* 2008]	[Trinh *et al.* 2008]	[Trinh *et al.* 2008]
growth conditions	anaerobic, glucose + minimal media	aerobic, glucose + minimal media	aerobic, xylose, arabinose, glucose, galactose and mannose + minimal media
no: reactions	59	60	71
no: metabolites	47	49	68
total no: EFMs	5010	38001	429275
F_1	1.6170	1.6103	2.2770
- max Y_{Etoh}	0.6667	0.6667	0.6667
- MCS cardinality	3	4	4
- number of MCSs	22	82	76
- number of EFMs	14	28	62
F_2	7.7860	8.5283	2.3169
- max η_{Etoh}	0.1390	0.1542	0.1542
- MCS cardinality	10	13	16
- number of MCSs	240	240	2880
- number of EFMs	4	2	6

Features of the networks on which the GA was tested. The maximum possible values for ethanol yield, Y_{Etoh} and efficiency, η_{Etoh} are presented. The minimal cardinality of MCSs which will force the network into these optimal values are also shown along with the total number of such MCSs and the number of EFMs which will survive after application of these MCSs. The corresponding fitness values, F_i have been obtained using the fitness functions presented in Table 6.4.

Table 6.4: **Fitness functions used**

i	**Design objective**	**Fitness function** F_i
1	Ethanol production with minimal MCS size	$w_1 \min Y_{Etoh} + w_3(1 - \|C\|/n)$
2	Substrate specific productivity with minimal MCS size	$w_2 \min \eta_{Etoh} + w_3(1 - \|C\|/n)$
3	Growth coupled product yield with minimal MCS size and maximum number of surviving modes	$w_1 \min Y_{Etoh} \times w_2 \max \eta_{Etoh} + w_3(1 - \|C\|/n) + w_4\|\mathbf{D}^C\|/\|\mathbf{E}\|$

Fitness functions used, where, w_1, w_2, w_3 and w_4 are weights associated with ethanol yield (Y_{Etoh}), ethanol efficiency (η_{Etoh}), MCS cardinality ($|C|$) and number of surviving modes ($|\mathbf{D}^C|$) respectively. These weights are used primarily to ensure desired contribution of the different variables towards the fitness function. They can also be used to give higher preference to a particular variable. C is the MCS, n the total number of reactions and $\mathbf{E}$ the set of all EFMs in a network. All fitness functions were maximised.

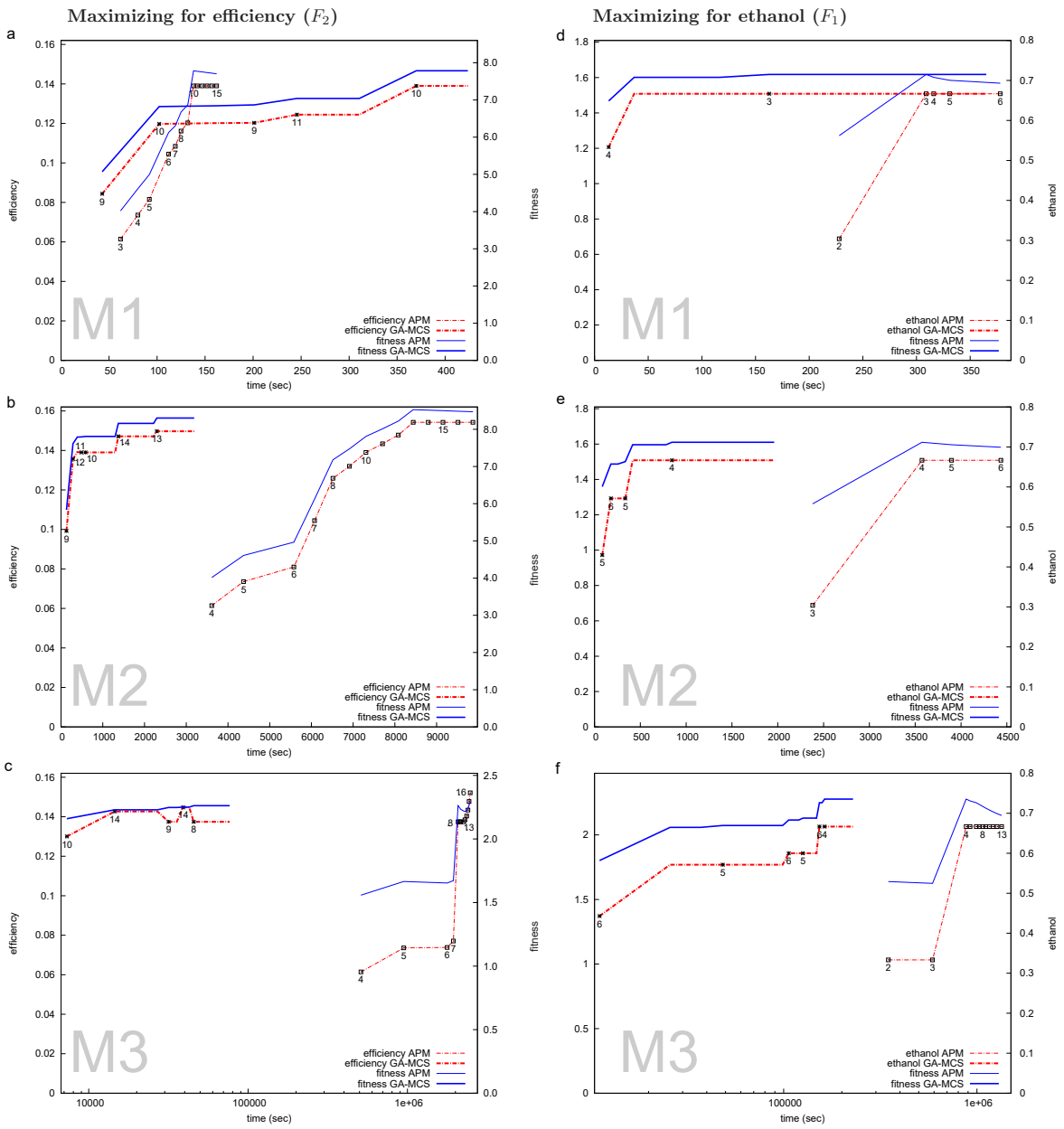


Figure 6.3: GA performance comparison. Comparing the performance of GA-MCS against APM [Ruckerbauer *et al.* 2014] using a single representative run for each model. The best solution in each generation was used to represent the performance of the GA. The numbers under the lines represent the cardinality of MCS corresponding to the objective value plotted. The time axes in c) and f) is in logarithmic scale. The fitness functions used are given in Table 6.4. Note that for the same objective with a lesser cMCS cardinality, the fitness will be higher.

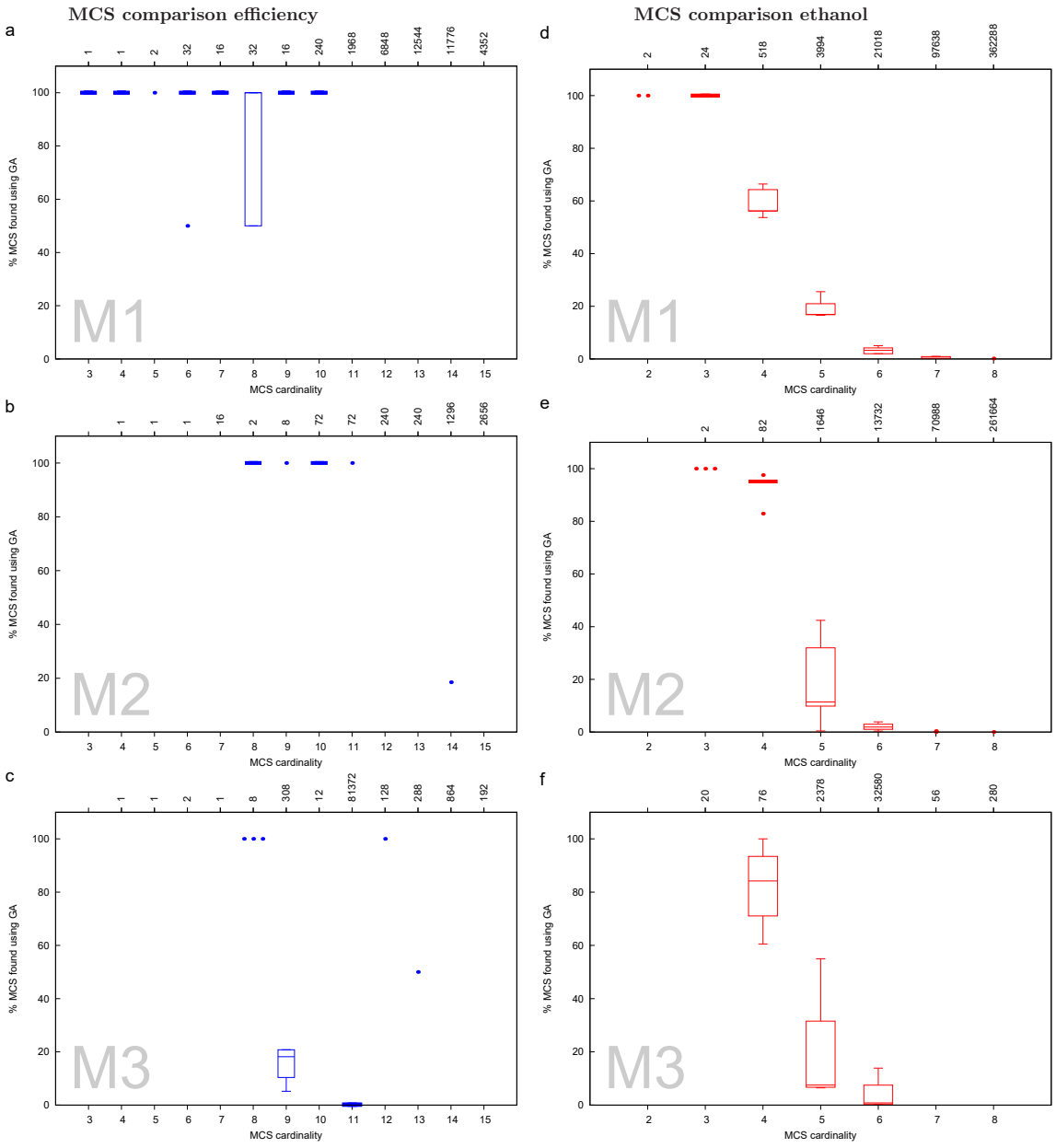


Figure 6.4: Number of solutions retrieved by the GA. Boxplots representing the number of matches between MCS retrieved using GA-MCS and APM broken down by cutset cardinality across five runs. Boxes have been drawn around the first and third quartile values, with the median being represented by the horizontal line within the box. Points represent outliers or data with three or lesser number of points. The numbers shown at the top of each plot indicate the total number of MCSs of the given cardinality as found by APM.

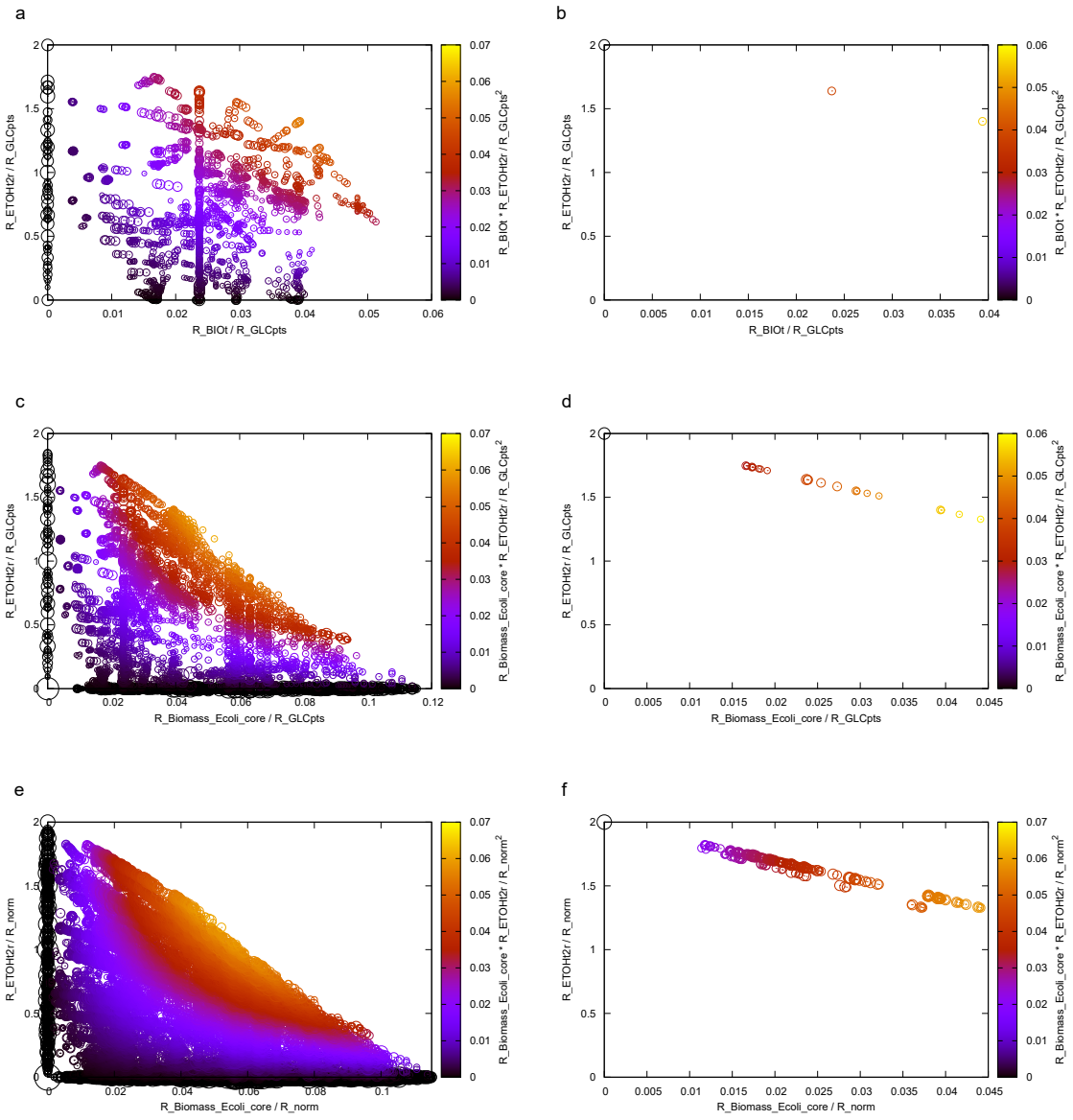


Figure 6.5: Complex designs optimized by the GA.Figures (a), (c) and (e) show the complete set of EFMs of the M1, M2 and M3 models respectively and (b), (d) and (f) represent corresponding solutions obtained using the GA which were obtained in 22 minutes, 28 minutes and 8 hours, 48 minutes respectively. EFMs are represented as a function of ethanol and biomass production. Each circle represents a set of EFMs with the same yield and efficiency. The diameter of the circle reflects the number of EFMs represented. The colour of the EFMs indicates their efficiency as specified by the index on the right hand side of each graph. R_ETOHt2r, R_BIOt and R_GLCpts represent the ethanol secretion, biomass and glucose uptake reactions in the model. In (b), the cutset corresponding to the solution is ({R_G6PDH2r R_FRD7 R_LDH_D R_ACt2r R_SUCCt3}), in (d), the modes represented are the ones which survive after applying the cutset ({R_GND R_FUM R_ACt2r R_D_LACt2 R_SUCCt3}) and in (f) the cutset corresponding to the solution is ({R_GND R_SUCOAS R_MALS R_ACt2r R_D_LACt2}). In e) and f) R_norm = R_GLCpts + R_MAN1 + R_TRA8 + R_TRA9 + R_TRA10.

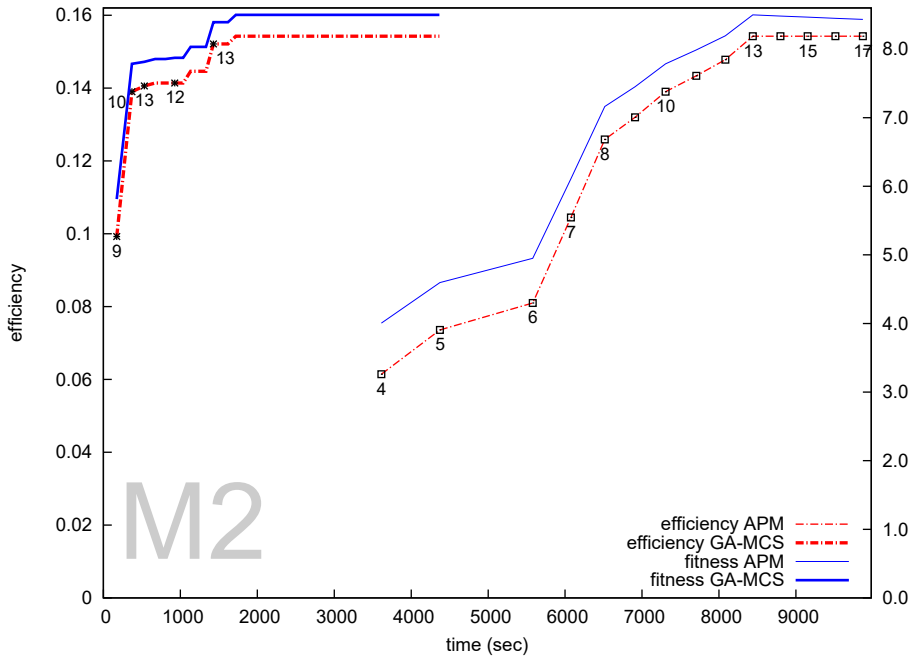


Figure 6.6: M2 maximum efficiency. Figure showing an instance where GA-MCS is able to find the global optimum for M2 efficiency by changing only a single parameter value in Table 6.2.

Chapter 7

Optimal knockout strategies in genome-scale metabolic networks using particle swarm optimization

---

This chapter was publish by Govind Nair, Christian Jungreuthmayer and Jürgen Zanghellini in *BMC Bioinformatics* 18:78, 2017, DOI: 10.1186/s12859-017-1483-5. [1]

- **Background** Knockout strategies, particularly the concept of constrained minimal cut sets (cMCSs), are an important part of the arsenal of tools used in manipulating metabolic networks. Given a specific design, cMCSs can be calculated even in genome-scale networks. We would however like to find not only the optimal intervention strategy for a given design but the best possible design too. Our solution (PSOMCS) is to use particle swarm optimization (PSO) along with the direct calculation of cMCSs from the stoichiometric matrix to obtain optimal designs satisfying multiple objectives.

- **Results** To illustrate the working of PSOMCS, we apply it to a toy network. Next we show its superiority by comparing its performance against other comparable methods on a medium sized *E. coli* core metabolic network. PSOMCS not only finds solutions comparable to previously published results but also it is orders of magnitude faster. Finally, we use PSOMCS to predict knockouts satisfying multiple objectives in a genome-scale metabolic model of *E. coli* and compare it with OptKnock and RobustKnock.

- **Conclusions** PSOMCS finds competitive knockout strategies and designs compared to other current methods and is in some cases significantly faster. It can be used in identifying knockouts which will force optimal desired behaviors in large and genome scale metabolic networks.

[1] JZ and GN conceived and designed the study. CJ and GN implemented the algorithm. GN designed the algorithm, ran the analysis and validated the results. All authors were involved in the analysis of the results and read, reviewed and approved the manuscript.

It will be even more useful as larger metabolic models of industrially relevant organisms become available.

Availability: https://github.com/gogothegreen/PSOMCS

7.1 Background

Metabolic engineering aims to improve product yields in cellular systems by applying a variety of tools. Constraint based methods which use only the stoichiometry of metabolic reactions have been particularly successful in the development of strategies towards fulfilling this aim [Stelling *et al.* 2002]. One important application is the prediction of knockouts to enforce desired metabolic behaviors in an organism. A method that allows one to predict efficient intervention strategies using the concept of minimal cut sets **MCSs**, was developed by Klamt and Gilles [Klamt & Gilles 2004]. This was generalized to constrained minimal cut sets **cMCS**, where in addition to blocking undesired fluxes, survival of some desired fluxes is possible [Hädicke & Klamt 2011, Jungreuthmayer *et al.* 2013b]. The automatic partitioning method **APM** uses an objective function to specify the design objectives and the partitioning of fluxes into desired/undesired is done automatically to find successively larger cMCS till a global optimum is reached [Ruckerbauer *et al.* 2014]. Previously we showed that a genetic algorithm could reach the global optimum faster than than APM [Nair *et al.* 2015]. However, all these methods are applicable only to small and medium-scale metabolic networks.

In a recent work by Ballerstein et al, it was shown that cMCS can be directly calculated from the stoichiometric matrix [Ballerstein *et al.* 2012]. Using this method, it is possible to calculate intervention strategies even in genome-scale metabolic networks [von Kamp & Klamt 2014]. Another work extended this concept to include regulation [Mahadevan *et al.* 2015]. A limitation of this method is that the desired flux or flux ratio of a metabolite has to be manually specified to get corresponding cMCS.

There exist other constraint based methods for predicting intervention strategies. OptKnock solves a bi-level optimization problem, to predict knockouts leading to maximal product formation at maximal growth [Burgard *et al.* 2003]. A three-level optimization problem is used to maximize minimal product formation in RobustKnock [Tepper & Shlomi 2010]. OptGene uses a genetic algorithm to predict knockouts [Patil *et al.* 2005]. Similarly, evolutionary algorithms and simulated annealing have been used in [Rocha *et al.* 2008]. Another metaheuristic approach was using a hybrid of bees algorithm with flux balance analysis **FBA** [Choon *et al.* 2014]. While

these methods optimize for design goals, doing so with a minimal number of knockouts is not necessarily guaranteed.

From an engineering perspective, we would like the organism to have a guaranteed high yield for the product of interest. Given that even in the face of genetic perturbations microorganisms redirect metabolic flux towards maximizing cellular growth [Ibarra *et al.* 2002], this high yield must be maintained at high growth rates. Additionally, the number of knockouts should be as small as possible to facilitate easy implementation in the laboratory.

Here we present a new method, PSOMCS, which uses particle swarm optimization **PSO** along with the method developed in [Ballerstein *et al.* 2012, von Kamp & Klamt 2014, Mahadevan *et al.* 2015] to calculate cMCS while overcoming the mentioned limitations of other methods. Our basic motivation is to combine the computational rigour of cMCS with the flexibility of the optimization-based approaches in order to solve (non-linear) intervention problems efficiently. We aim to find not only the optimal intervention strategy for a given design but also the best possible design. In addition, we show that PSOMCS is also faster than other methods which try to find cMCS leading to optimal design objectives.

7.2 Methods

7.2.1 Calculating cMCS

A metabolic network of m internal metabolites connected by n reactions in steady state is represented by the set of linear equations

$$\mathbf{Nr} = \mathbf{0} \tag{7.1}$$

where $\mathbf{N}$ is a $m \times n$ matrix consisting of stoichiometric coefficients of all participating reactions such that each column represents one reaction. $\mathbf{r}$ is a vector of reaction fluxes. Reactions can be both reversible (Rev) and irreversible ($Irrev$), thereby imposing the constraint

$$r_i \geq 0 \;\; \forall\, i \in Irrev. \tag{7.2}$$

(7.1) and (7.2) define a flux space. Depending on the desired outcome, an intervention problem can be set up dividing this space into desired and undesired fluxes. The set of undesired fluxes for t reactions can be defined by

$$\mathbf{Tr} \leq \mathbf{t} \tag{7.3}$$

where $\mathbf{T} \in \mathbb{R}^{t \times n}$ and $\mathbf{t} \in \mathbb{R}^{t \times 1}$. Likewise, the set of desired fluxes for d reactions can be defined by

$$\mathbf{Dr} \leq \mathbf{d} \tag{7.4}$$

with $\mathbf{D} \in \mathbb{R}^{d \times n}$ and $\mathbf{d} \in \mathbb{R}^{d \times 1}$.

In [von Kamp & Klamt 2014], cMCS are calculated by first solving a series of mixed integer linear programming **MILP** problems representing (7.1) and (7.3) and then filtering those solutions which also satisfy (7.4). In [Mahadevan *et al.* 2015], this is combined into a single system represented as (cf. equation (5) in [Mahadevan *et al.* 2015])

$$\begin{pmatrix} \mathbf{N}_{rev}^T & \mathbf{I}_{rev} & -\mathbf{I}_{rev} & \mathbf{T}_{rev}^T & 0 \\ \mathbf{N}_{irr}^T & \mathbf{I}_{irr} & -\mathbf{I}_{irr} & \mathbf{T}_{irr}^T & 0 \\ 0 & 0 & 0 & 0 & \mathbf{N} \\ 0 & 0 & 0 & 0 & \mathbf{D} \end{pmatrix} \times \begin{pmatrix} \mathbf{u} \\ \mathbf{vp} \\ \mathbf{vn} \\ \mathbf{w} \\ \mathbf{r} \end{pmatrix} \begin{matrix} = \\ \geq \\ = \\ \leq \end{matrix} \begin{pmatrix} 0 \\ 0 \\ 0 \\ \mathbf{d} \end{pmatrix} \tag{7.5}$$

$$\mathbf{t}^T\mathbf{w} \leq -c$$

$$\mathbf{u} \in \mathbb{R}^m, \mathbf{vp}, \mathbf{vn} \in \mathbb{R}^n, \mathbf{d} \in \mathbb{R}^d, \mathbf{vp}, \mathbf{vn}, \mathbf{w}, \mathbf{r_{irr}} \geq 0, c > 0.$$

Note that the $\mathbf{N}$ and $\mathbf{T}$ matrices have been split into reversible (subscript *rev*) and irreversible submatrices (subscript *irr*). Similarly, identity submatrices for reversible and irreversible reactions are represented by the matrices $\mathbf{I}_{rev}$ and $\mathbf{I}_{irr}$ respectively. cMCS are directly calculated by finding solutions with minimum number of non-zero entries in $\mathbf{vp}, \mathbf{vn}$. Additionally binary *indicator* variables $\mathbf{zp}$ and $\mathbf{zn}$ are introduced such that $zp_i = 0$ if $vp_i = 0$ and $zp_i = 1$ if $vp_i > 0$ and similarly for zn, vn. Only one direction of $\mathbf{v}$ (either vp_i or vn_i) can be active, hence

$$zp_i + zn_i \leq 1. \tag{7.6}$$

We set up the following optimization problem

$$\begin{matrix} \text{minimize} \sum_{i=1}^{n} (zp_i + zn_i) \\ \text{s.t. } (7.5), (7.6) \end{matrix} \tag{7.7}$$

with the additional constraint that the flux through a reaction is turned off if it is part of a cMCS, i.e., $r_i = 0$ if $zp_i = 1 \mid\mid zn_i = 1$.

With this system it is possible to find cMCS which will result in designs satisfying constraints on yields/fluxes specified by (7.3), (7.4). However, we would like to have a method which given some design objectives (e.g., high product yield even at high growth rates) calculates cMCS corresponding to optimal values for the design objectives. Since any design can be represented as a function of $\mathbf{T}, \mathbf{D}, \mathbf{t}$ and $\mathbf{d}$, the optimization problem can be stated as

$$\begin{matrix} \max f(\mathbf{T}, \mathbf{D}, \mathbf{t}, \mathbf{d}) \\ \text{s.t. } (7.7). \end{matrix} \tag{7.8}$$

In other words, the problem is to find optimal combinations of {target/desired} yields for all reactions to be optimized. This is not easy for a few reasons. In general, this is a non-linear optimization problem. Non-linear optimization is known to be inherently complex with general deterministic solutions being impossible to find. Secondly, slight adjustments in (7.3), (7.4) could result in completely different cMCS with different cardinalities. Finally, not all such combinations will result in cMCS. These issues become acute when the search space is more dense with many possible combinations, as in large and genome-scale metabolic networks. We attack this problem using PSO as it has been successfully used to find solutions to complex non-linear optimization problems in other fields [Poli *et al.* 2007, Banks *et al.* 2008, Del Valle *et al.* 2008].

7.2.2 Particle swarm optimization

PSO is a metaheuristic inspired by the flocking behavior of birds [Kennedy & Eberhart 1995]. In PSO, particles distributed within a multi-dimensional space collectively move towards an optimum guided by a fitness function. Particle fitness is determined by its position in the search space. The motion of a particle is influenced by its neighbours and the currently known fittest particle. More information on PSO can be found in [Poli *et al.* 2007, Banks *et al.* 2008, Del Valle *et al.* 2008, Banks *et al.* 2007].

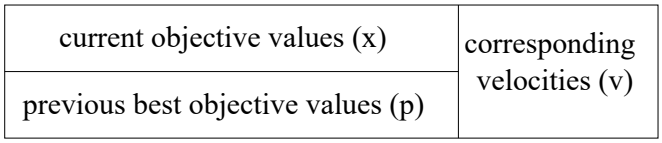


Figure 7.1: **Schematic of the PSO particle.** A particle stores three types of information: the current values, values corresponding to its own previous fitness and velocities corresponding to each objective.

Typically, a particle is made up of three j-dimensional vectors, where j is the dimensionality of the search space. These represent the current position x, its previous best position p which is the position corresponding to the highest fitness achieved by the particle and the velocity v, Figure 7.1. Particle motion is guided by the following equations,

$$v_i(t+1) = \chi\{v_i(t) + \varphi_1\,\beta_1\,[p_i(t) - x_i(t)] + \varphi_2\,\beta_2\,[g_i(t) - x_i(t)]\} \tag{7.9}$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \tag{7.10}$$

$$i \in \{1..j\}.$$

g is the position corresponding to the global best fitness of the entire swarm till the current t. $\varphi_{1,2}$ are called "acceleration constants" and determine the relative influence of the particle's own knowledge and that of the group, both of which are commonly set to 2 [Del Valle *et al.* 2008, Banks *et al.* 2007]. $\beta_{1,2}$ are uniformly generated random numbers within the range $(0, 1]$ for each i,t. χ is the constriction coefficient first introduced in [Clerc & Kennedy 2002] and generally has a value of 0.7298 in the literature [Poli *et al.* 2007, Del Valle *et al.* 2008]. This dampens the dynamics of the particles, preventing the velocity from rapidly increasing beyond the problem bounds. The amount of information available to a particle depends on its access to information of other particles. Access to a limited number of other particles is closer to the behaviour of natural swarms. In our implementation each particle is connected to four other particles, which has a comparatively better performance than other choices[Kennedy & Mendes 2002]. Additionally, we borrow a concept from [Gong & Zhang 2013], where in addition to its fixed neighbours, a particle also establishes connection with another randomly selected particle.

The MILP given by (7.7) needs constraints specified by (7.3), (7.4) to calculate corresponding cMCS. For example, consider a network which has, among other reactions, a substrate uptake reaction R_S, a reaction for the product secretion R_P and one for biomass R_{Bio}. An optimal design could be stated as having $R_P/R_S \geq x_1$ and also that biomass fluxes of $R_{Bio}/R_S \geq x_2$ exist. However, we don't know the combinations of x_1, x_2 resulting in optimal design. This is where a PSO can be useful. After initializing $\mathbf{x}, \mathbf{v}$, the set of positions and velocities for all particles, within the range of values for $\{x_1, x_2\}$ on some constant R_S, the PSO iteratively finds increasingly better solutions for (7.8) using (7.9) and (7.10) and moves towards the global optimum. The PSOMCS flowchart is shown in Figure 7.2.

The fitness function will depend on the nature of the desired optimum. Considering that our objective is to have a design with high yields and minimal knockouts, the following fitness function was used,

$$F(x) = \left(1 - \frac{|cMCS|}{n}\right) \cdot \prod_i \frac{x_i}{x_i(max)}. \tag{7.11}$$

7.3 Results

To clarify the working of PSOMCS, we first apply our method to a small toy network, optimizing for only a single reaction. Next, to confirm the accuracy of our predictions, we compare our method against another method based on a genetic algorithm (GAMCS) which we had previously developed [Nair *et al.* 2015]. The model used is the medium-scale *E. coli* core model presented in [Trinh *et al.* 2008]. Finally we find optimal intervention strategies for maximizing the minimal product yield in a genome-scale metabolic network. FBA was used to calculate the range of yields [min:max] for each objective and particles were initialised within this range. Only one solution is calculated for a MILP. The parameters used are shown in Table 7.1. Implementation of PSOMCS was done using Perl `http://www.perl.org/`. For the performance critical parts of the program, i.e., solving the MILP and also the LP, the IBM ILOG CPLEX Optimization Studio - a commercial optimization package - was used through the Math::CPLEX Perl module. Also, our algorithm is designed to make use of modern CPU architectures and can be run in parallel on multiple cores.

Table 7.1: **PSOMCS parameters**

Model	**No: particles**	**No: iterations**
toy network	4	2
E. coli core	10	40
iAF1260	10	40

Details of parameters used for the different models.

Consider the network given in Figure 7.3. We wish to find minimal knockouts which will ensure the highest possible yield for reaction R4. In the first iteration, cMCS corresponding to low yields are found. In the second iteration, all particles move towards higher yields. One particle, on the solution of its dual system gives the cMCS of 'R2 R9'. Removal of R2 and R9 from the network blocks all flux through R5 and R6, thus redirecting the network flux through R4. This corresponds to the highest minimal yield of 1 for R4.

We apply PSOMCS to generate designs in an *E. coli* core network which will ensure high yield of ethanol even in the face of high growth. This network was previously used to design a high yield ethanol producing strain in [Trinh *et al.* 2008]. This model has 71 reactions and 68 metabolites. We had previously used this model to predict optimal intervention strategies using a genetic algorithm (GAMCS), which we had shown to be faster than other

current approaches [Nair *et al.* 2015], particularly compared to APM, which is guaranteed to find the optimal solution [Ruckerbauer *et al.* 2014]. Here we compare our approach with GAMCS in terms of speed and accuracy of results. The machine used had the following specifications − 2 CPUs, 12 cores, Intel Xeon X5650 2.67 GHz, running an Ubuntu 14.4 LTS operating system. The time taken for a typical PSOMCS and GAMCS run is plotted in Figure 7.4a. The superiority of our method in terms of speed can be clearly observed. GAMCS takes 34,857 seconds to reach the maximum fitness. PSOMCS takes only 1,493 seconds for the same. This is an over 23 fold improvement in performance. In comparison, APM would not only require that the desired EFMs be assigned weights, but also the time taken by it would have been outside the boundaries of this plot. The cMCS corresponding to the optimum obtained by both GAMCS and PSOMCS are exactly the same. Figure 7.4b is one of the designs corresponding to a high fitness. This design was in the solution pool of both the PSO and GA methods. In this design, a minimum ethanol yield of 1.33 is guaranteed even when the growth rate is 0.044. Also, as can be expected, production of competing by-products: acetate, lactate and succinate is blocked. Additionally, flux through the oxidative part of the pentose phosphate pathway is blocked and so is the pyruvate-malate cycling. Multiple cMCS resulting in similar design characteristics were returned by our method.

To test the capabilities of our method we applied it to the genome-scale model of *E. coli* presented in [Feist *et al.* 2007]. Our aim was to find cMCS that result in an scenario of growth-coupled ethanol yield. A few strategies were used in [von Kamp & Klamt 2014, Mahadevan *et al.* 2015] to reduce the network size. These strategies are aimed at reducing the network size and improving computational efficiency, which takes real growth conditions into account and removing all superfluous components. First, the network was reduced to grow anaerobically on glucose as the only carbon source. The resulting network has 1413 reactions and 971 metabolites. Network compression was done by combining reactions operating at fixed ratios into reaction subsets. Exchange reactions, spontaneous reactions and reactions essential for the ethanol and biomass production were excluded from participating in cMCS by setting their corresponding zp, zn variables to zero. The machine we used for this test had 24 CPUs, 396GB RAM, Intel Xeon E5-2667 2.90 GHz processor, running on Ubuntu 14.4 LTS. The cMCS cardinality was limited to 5. With 4 particles being processed in parallel, the program was run for 40 iterations. It took 14 iterations ($\sim$ 74 hours) to find the optimal design. One of the designs is shown in Figure 7.5 along with designs obtained using OptKnock and RobustKnock on the same machine. The envelope of the strain specific phenotypic solution space was calculated with flux variability analysis **FVA** [Mahadevan & Schilling 2003] of the iAF1260 network while considering

the respective knockouts predicted by each method. The minimally required biomass production was set at 0.006 and both were limited by unit glucose uptake and a maximum knockout size of 5. OptKnock took 4 minutes to run while RobustKnock ran for 71 minutes. The minimal ethanol yields were 0 in both cases. As can be observed, PSOMCS offers a better design with the ethanol production being strongly coupled to biomass production and at no point falls below a yield of 0.9.

7.4 Discussion

Here we have presented a method, PSOMCS, to design strains with high minimal product yield using knockouts of minimal possible size. To do this, we employ a PSO together with the direct enumeration of cMCS developed in [Ballerstein *et al.* 2012, von Kamp & Klamt 2014, Mahadevan *et al.* 2015]. This method has made it possible to find cMCS in large and genome-scale networks. However, it is not designed to optimize engineering goals. That is, we we would like to find not only the optimal intervention strategy for a given design but the best possible design too. Finding intervention strategies that achieve this is an important goal of metabolic engineering, especially in the production of industrially important chemicals. We deliver on this goal by using a PSO built on top of the base provided by the direct enumeration of cMCS. Our method thus expands the utility of this method. Additionally we would like to point out that in the case of optimizing for a single reaction, solving (7.5) with continuous values within the [min:max] range for that reaction would suffice. However, in the presence of multiple objectives this task becomes computationally exhaustive and infeasible, thereby justifying the use of a metaheuristic approach such as the one used here.

There have been other methods with a similar strategy as ours, which is the use of a metaheuristic in combination with another method like linear programming. Most methods have relied on genetic algorithms [Patil *et al.* 2005, Nair *et al.* 2015, Boghigian *et al.* 2010], evolutionary algorithms and simulated annealing [Rocha *et al.* 2008] and also an artificial bees algorithm [Choon *et al.* 2014]. Ours is the first attempt at using the dual method in a similar fashion, along with the use of a PSO.

As shown by the comparison with OptKnock and RobustKnock in Figure 7.5, although all designs have the same highest ethanol yield of 2, PSOMCS provides a design with the highest guaranteed minimal ethanol yield. RobustKnock was developed to overcome the 'too-optimistic' nature of OptKnock and this is reflected in the nature of their respective designs. Also of note is the fact that both OptKnock and RobustKnock need a minimal level of

biomass production to be manually specified while PSOMCS does not. In fact, if we reduce the minimal biomass production requirement to 0.001 (in order to mimic the PSOMCS settings), RobustKnock runs for over 90 hours without finding the optimum. Running OptKnock and RobustKnock multiple times with different biomass levels will result in different solutions, some of which will be better than others. PSOMCS eliminates this need to manually set reaction fluxes and searches the entire feasible space of biomass yields to find the optimal one. Growth-coupling is a key principle in metabolic engineering. It requires that growth should only be feasible if a desired compound, like ethanol, is mandatorily produced as by-product. It can be seen in Figure 7.5 that PSOMCS achieves this with a growth rate about one third of the wild-type. However, growth-coupling does not enforce nor require that the maximal product yield is attained at a non-zero growth rate. In fact Figure 7.5 illustrates the rule rather than the exception, as typically the maximum product yield is achieved at zero growth [Campodonico *et al.* 2014, Klamt & Mahadevan 2015]. Furthermore, an ideal production state will be characterized by zero growth, where all available resources are used for product formation. In this senses, biomass production can be seen as an "unwanted" by-product. Recent advances in fermentation processes employ zero-growth approaches [Lange *et al.* 2016, Rebnegger *et al.* 2016]. However, these approaches are associated with many challenges which go far beyond the scope of the presented work. Nevertheless, Figure 7.5 indicates that the presented designs retain their wild-type behavior to be operated as optimal zero-growth factories.

In heuristic search algorithms, performance comes at the cost of being too specific to the problem being solved [Wolpert & Macready 1997]. By virtue of having few parameters, PSOs are less affected by this problem. In our implementation, we have used parameter values as found in the general PSO literature without the need to adjust them. The only parameters that we adjusted were the number of particles and the number of iterations. We clearly use fewer particles than is typical. This is because we found a population size of 10 to be sufficient for our needs (see Figure 7.6). Although we have sampled the entire solution space, particles can easily be forced to explore a subspace. Certain reactions can be excluded from being considered for knockouts by forcing their corresponding indicator variables in the dual system to be 0. Our fitness function is specific to our target design, however new fitness functions can be thought of depending on the desired final objective. Our method produces cMCS leading to designs with similar characteristics as the one used in [Trinh *et al.* 2008]. Our method also returns multiple solutions. The limiting factor in our method is the MILP for the dual system.

MILPs are more difficult to solve than LPs and may consume large

amounts of time as well as memory [Cornuéjols *et al.* 2006]. During our runs, the search tree generated by CPLEX's Branch and Cut algorithm for a single MILP grew to consume over 130 GB of memory when limited to a knockout size of 6. This memory consumption grows quickly with increasing knockout size, thereby limiting the ability of PSOMCS to find the optimal solution.

Improvements in run time can be made by forcing PSOMCS to explore only a part of the flux space leading to a smaller solution space to be explored. For instance lets consider the design in Figure 7.5, with a minimal biomass yield of 0.01, the optimal design presented here was found within 24 hours. Further improvements to performance could be obtained by following the strategies outlined in [Klotz & Newman 2013]. Also, algorithmic improvements in solving MILPs could be useful in this regard.

Here we have dealt only with knockout strategies to design better strains. It can easily be extended to include the concept of regulatory MCS introduced in [Mahadevan *et al.* 2015] which combine reaction up/downregulation with knockouts. There are other constraint based methods dealing with intervention strategies like gene knock-ins and up/downregulation. PSOs and swarm intelligence algorithms in general may be used to compliment these methods.

7.5 Conclusion

PSOMCS finds the best possible design in metabolic networks given multiple objectives with the corresponding cMCS. We have demonstrated its capability in finding optimal knockouts and designs in genome-scale metabolic networks. It finds competitive designs compared to standard tools and is orders of magnitude faster than EFM based tools in finding the optimal solution. PSOMCS could be used to predict minimal knockouts resulting in optimal yields in industrially important microorganisms. As the size and quality of metabolic models increase, methods like the one presented here will be even more useful.

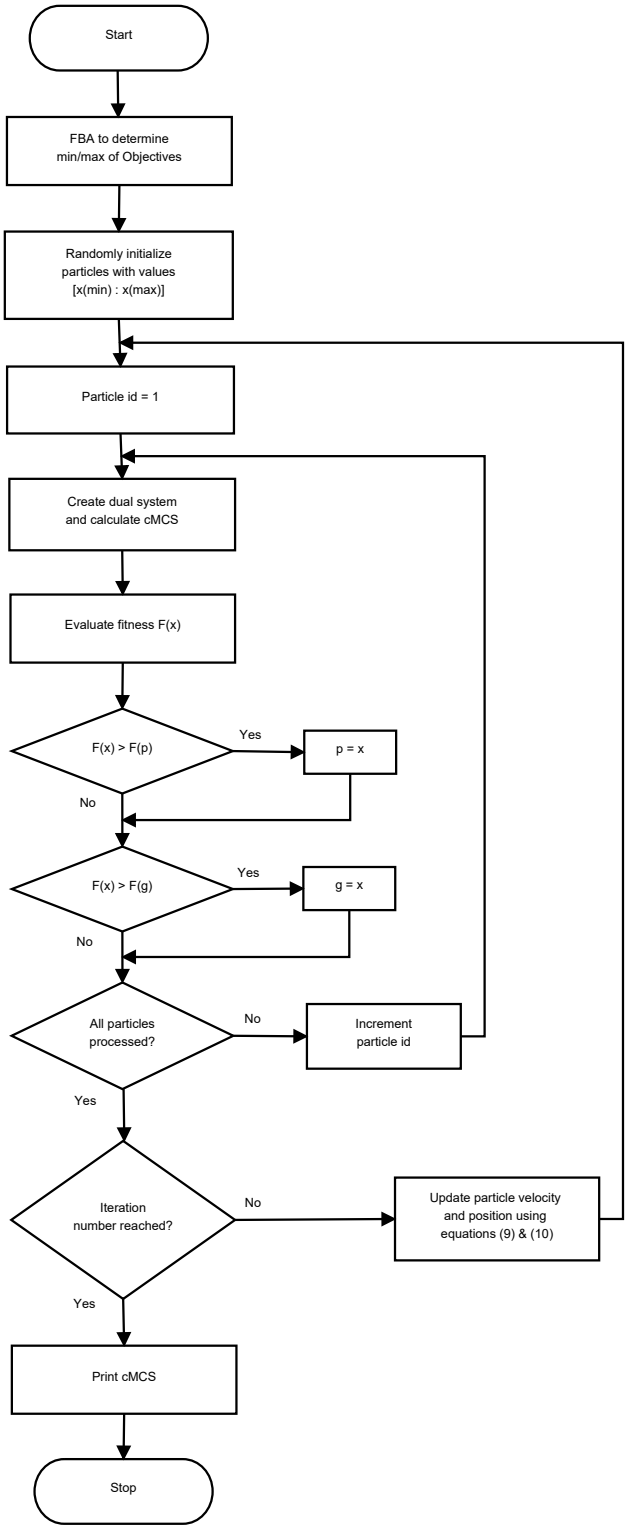
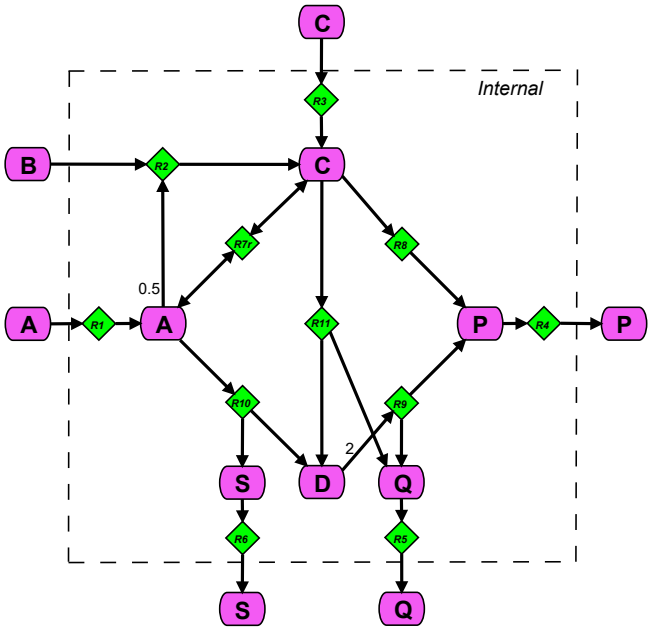


Figure 7.2: **Flowchart of PSOMCS.** p and g are the current particle best and global best respectively. The algorithm stops when the number of iterations reaches a pre-specified maximum or if the maximum fitness remains unchanged for a pre-specified number of iterations.



FBA to determine min/max:
R1 + R2 + R3 = 1 ⟶ unit uptake
R4: min = 0.25 : max = 1.00

Target:
R4/(R1+R2+R3) < x
Desired:
R4/(R1+R2+R3) ≥ x

Fitness function:
$F(x) = \left(1 - \frac{|cMCS|}{11}\right) * \frac{x}{1}$

Iteration 1

Id	x	p	v	$cMCS$	$F(x)$
1	0.19572	undef	0.29870	-	0
2	0.27584	undef	0.33256	R7	0.25076
3	0.27540	undef	0.44428	R7	0.25036
4	0.33756	undef	0.23672	R9	0.30687

Iteration 2

Id	x	p	v	$cMCS$	$F(x)$
1	0.69188	undef	-0.16052	R2 R9	0.56608
2	0.42140	0.27584	-0.27577	R9	0.38309
3	0.28611	0.27540	0.01071	R7	0.2601
4	0.97936	0.33756	-0.10257	R2 R9	0.80129

Figure 7.3: **PSOMCS small example.** Running the PSOMCS on a toy network. This network has three input reactions, which can be assumed to be substrates and three secretion reactions, which can be assumed to be three different products. We want to maximise the yield of R4, that is maximize (R4/(R1 + R2 + R3)). Note that the particles operate in a single dimensional search space and x represents the yield for R4. After performing FBA to determine the maximum and minimum yields for R4 given unit substrate uptake, four particles are initialised within this range. Initial velocities are also assigned. cMCSs are calculated after creating and solving the dual system. Fitness is a function of x and the cardinality of the cMCS. g corresponds to x with the highest fitness which is particle 4 after both the first and second iterations. After the first iteration, every particle except the first has a value for p. Note that for particle 4 a yield higher than 0.98 is guaranteed. In reality, the minimal yield with the corresponding cMCS is 1, which is also the case for particle 1. This is the value the algorithm will return if allowed to run for a few more iterations.

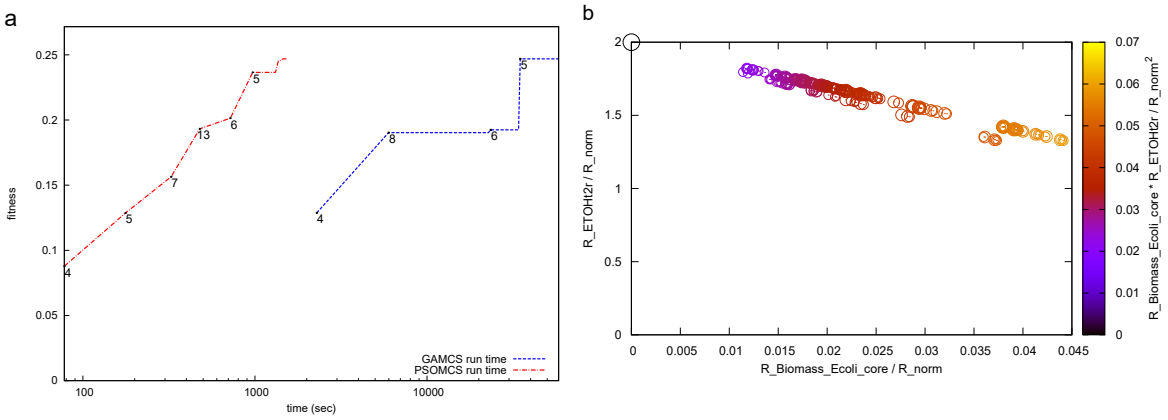


Figure 7.4: **Comparing the runtimes of PSOMCS and GAMCS. a**) Plotting the runtimes of PSOMCS and the GA we had previously implemented clearly shows PSOMCS is orders of magnitude faster than GAMCS. Note that the time axis is logarithmic and that both algorithms reach the same maximum fitness. **b**) Both methods also produce similar designs, an example of which is shown. This design is obtained with 5 knockouts (R_GND R_SUCOAS R_MALS R_ACt2r R_LDH_D). The plot was generated by applying the knockouts on the complete set of 429275 EFMs of the *Escherichia coli* core model. R_norm is the sum of uptake rates for the five carbon substrates, glucose, galactose, mannose, arabinose and xylose under aerobic conditions.

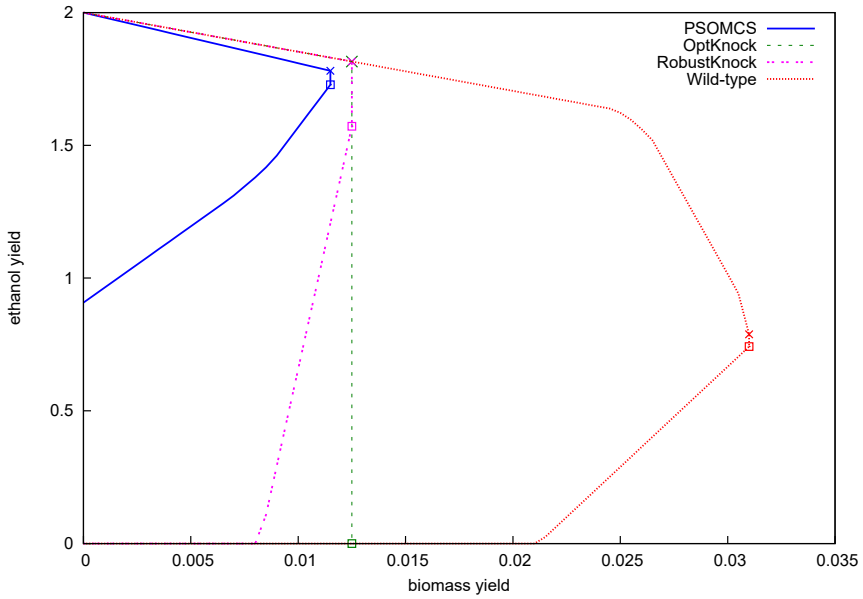


Figure 7.5: **Design for a genome-scale *E. coli* model.** *E. coli* was designed for enhanced ethanol production using the genome-scale iAF1260 model. For comparison, designs obtained using OptKnock and RobustKnock are also presented. The design using PSOMCS guarantees a minimal ethanol yield of 0.9, in contrast this is 0 for both RobustKnock and OptKnock. All designs have a maximum biomass production rate greater than 0.01 with the one for PSOMCS being comparatively lower. The maximum yield for all the designs is 2. The given plots have been generated by using FVA on the iAF1260 model while considering the respective knockouts produced by each method. The FBA solution space at maximum growth is highlighted, with crosses indicating the maximum and squares the minimum ethanol yield. All designs involve 5 knockouts - (R_ACALD R_GLUDy R_Htex R_PGI R_TKT2) for PSOMCS, (R_ACALD R_H2tex R_PHEt2rpp R_PPKr R_TYRtex) for OptKnock and (R_ACKr R_F6PA R_FBA R_GLCptspp R_PGCD) for RobustKnock.

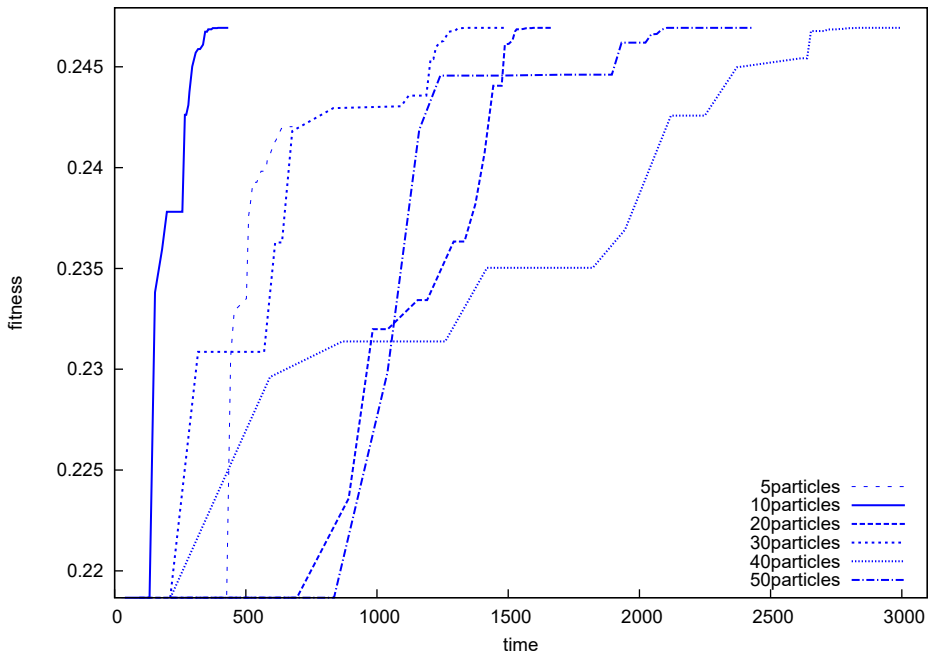


Figure 7.6: **Comparison of runtimes for different swarm sizes.** The time taken for swarms of different sizes to reach the maximum fitness in the *E. coli* core model is plotted. It is clear that a size of 10 is faster than larger swarms and a smaller size of 5 fails to reach the optimum.

Bibliography

[Acuna *et al.* 2009] Vicente Acuna, Flavio Chierichetti, Vincent Lacroix, Alberto Marchetti-Spaccamela, Marie-France Sagot and Leen Stougie. *Modes and cuts in metabolic networks: Complexity and algorithms.* Biosystems, vol. 95, no. 1, pages 51–60, 2009. (Cited on page 18.)

[Alper *et al.* 2005a] Hal Alper, Yong-Su Jin, JF Moxley and G Stephanopoulos. *Identifying gene targets for the metabolic engineering of lycopene biosynthesis in Escherichia coli.* Metabolic engineering, vol. 7, no. 3, pages 155–164, 2005. (Cited on page 12.)

[Alper *et al.* 2005b] Hal Alper, Kohei Miyaoku and Gregory Stephanopoulos. *Construction of lycopene-overproducing E. coli strains by combining systematic and combinatorial gene knockout targets.* Nature biotechnology, vol. 23, no. 5, pages 612–616, 2005. (Cited on page 12.)

[Báez-Viveros *et al.* 2004] José Luis Báez-Viveros, Joel Osuna, Georgina Hernández-Chávez, Xavier Soberón, Francisco Bolívar and Guillermo Gosset. *Metabolic engineering and protein directed evolution increase the yield of L-phenylalanine synthesized from glucose in Escherichia coli.* Biotechnology and bioengineering, vol. 87, no. 4, pages 516–524, 2004. (Cited on page 1.)

[Ballerstein *et al.* 2012] Kathrin Ballerstein, Axel von Kamp, Steffen Klamt and Utz-Uwe Haus. *Minimal cut sets in a metabolic network are elementary modes in a dual network.* Bioinformatics, vol. 28, no. 3, pages 381–387, 2012. (Cited on pages 3, 19, 22, 40, 50, 52, 64, 65 and 71.)

[Baneyx 1999] François Baneyx. *Recombinant protein expression in Escherichia coli.* Current opinion in biotechnology, vol. 10, no. 5, pages 411–421, 1999. (Cited on page 4.)

[Banks *et al.* 2007] Alec Banks, Jonathan Vincent and Chukwudi Anyakoha. *A review of particle swarm optimization. Part I: background and development.* Natural Computing, vol. 6, no. 4, pages 467–484, 2007. (Cited on pages 28, 67 and 68.)

[Banks *et al.* 2008] Alec Banks, Jonathan Vincent and Chukwudi Anyakoha. *A review of particle swarm optimization. Part II: hybridisation, combinatorial, multicriteria and constrained optimization, and indicative applications.* Natural Computing, vol. 7, no. 1, pages 109–124, 2008. (Cited on pages 28 and 67.)

[Beasley *et al.* 1993] David Beasley, RR Martin and DR Bull. *An overview of genetic algorithms: Part 1. Fundamentals.* University computing, vol. 15, pages 58–58, 1993. (Cited on page 44.)

[Becker *et al.* 2007] Scott A Becker, Adam M Feist, Monica L Mo, Gregory Hannum, Bernhard Ø Palsson and Markus J Herrgard. *Quantitative prediction of cellular metabolism with constraint-based models: the COBRA Toolbox.* Nature protocols, vol. 2, no. 3, pages 727–738, 2007. (Cited on pages 9 and 12.)

[Behjati & Tarpey 2013] Sam Behjati and Patrick S Tarpey. *What is next generation sequencing?* Archives of disease in childhood-Education & practice edition, pages edpract–2013, 2013. (Cited on page 2.)

[BioCarta 2017] BioCarta. *BioCarta*. http://www.biocarta.com/, 2017. Accessed: 2017-01-10. (Cited on page 5.)

[BioModels 2017] BioModels. *BioModels*, accessed February 17, 2017. https://www.ebi.ac.uk/biomodels-main/. (Cited on page 6.)

[Boghigian *et al.* 2010] Brett A Boghigian, Hai Shi, Kyongbum Lee and Blaine A Pfeifer. *Utilizing elementary mode analysis, pathway thermodynamics, and a genetic algorithm for metabolic flux determination and optimal metabolic network design.* BMC systems biology, vol. 4, no. 1, page 49, 2010. (Cited on pages 50 and 71.)

[Bonyadi & Michalewicz 2016] Mohammad Reza Bonyadi and Zbigniew Michalewicz. *Particle swarm optimization for single objective continuous space problems: a review.* Evolutionary computation, 2016. (Cited on page 28.)

[BRENDA 2017] BRENDA. *BRaunschweig ENzyme Database (BRENDA)*. http://www.brenda-enzymes.org/, 2017. Accessed: 2017-01-10. (Cited on page 5.)

[Bro *et al.* 2006] Christoffer Bro, Birgitte Regenberg, Jochen Förster and Jens Nielsen. *In silico aided metabolic engineering of Saccharomyces cerevisiae for improved bioethanol production.* Metabolic engineering, vol. 8, no. 2, pages 102–111, 2006. (Cited on page 12.)

[Broddrick 2017] Jared Broddrick. *Available predictive genome-scale metabolic network reconstructions*, accessed February 17, 2017. http://sbrg.ucsd.edu/InSilicoOrganisms/OtherOrganisms. (Cited on page 2.)

[Brooke *et al.* 1988] Anthony Brooke, David A Kendrick, Alexander Meeraus and Richard E Rosenthal. Gams: A user's guide. Course Technology, 1988. (Cited on page 12.)

[Burgard *et al.* 2003] Anthony P Burgard, Priti Pharkya and Costas D Maranas. *Optknock: a bilevel programming framework for identifying gene knockout strategies for microbial strain optimization.* Biotechnology and bioengineering, vol. 84, no. 6, pages 647–657, 2003. (Cited on pages 3, 32, 50 and 64.)

[Burgard *et al.* 2004] Anthony P Burgard, Evgeni V Nikolaev, Christophe H Schilling and Costas D Maranas. *Flux coupling analysis of genome-scale metabolic network reconstructions.* Genome research, vol. 14, no. 2, pages 301–312, 2004. (Cited on page 9.)

[Campodonico *et al.* 2014] Miguel A Campodonico, Barbara A Andrews, Juan A Asenjo, Bernhard O Palsson and Adam M Feist. *Generation of an atlas for commodity chemical production in Escherichia coli and a novel pathway prediction algorithm, GEM-Path.* Metabolic engineering, vol. 25, pages 140–158, 2014. (Cited on page 72.)

[Carlson & Srienc 2004] Ross Carlson and Friedrich Srienc. *Fundamental Escherichia coli biochemical pathways for biomass and energy production: identification of reactions.* Biotechnology and bioengineering, vol. 85, no. 1, pages 1–19, 2004. (Cited on page 15.)

[Caspi *et al.* 2016] Ron Caspi, Richard Billington, Luciana Ferrer, Hartmut Foerster, Carol A Fulcher, Ingrid M Keseler, Anamika Kothari, Markus Krummenacker, Mario Latendresse, Lukas A Mueller*et al. The MetaCyc database of metabolic pathways and enzymes and the BioCyc collection of pathway/genome databases.* Nucleic acids research, vol. 44, no. D1, pages D471–D480, 2016. (Cited on page 5.)

[Causey *et al.* 2004] TB Causey, KT Shanmugam, LP Yomano and LO Ingram. *Engineering Escherichia coli for efficient conversion of glucose to pyruvate.* Proceedings of the National Academy of Sciences of the United States of America, vol. 101, no. 8, pages 2235–2240, 2004. (Cited on page 1.)

[CellML 2017] CellML. *CellML*, accessed February 17, 2017. `https://www.cellml.org/`. (Cited on page 6.)

[Chandran *et al.* 2009] D Chandran, WB Copeland, SC Sleight and Herbert M Sauro. *Mathematical modeling and synthetic biology.* Drug

Discovery Today: Disease Models, vol. 5, no. 4, pages 299–309, 2009. (Cited on page 2.)

[Chassagnole *et al.* 2002] Christophe Chassagnole, Naruemol Noisommit-Rizzi, Joachim W Schmid, Klaus Mauch and Matthias Reuss. *Dynamic modeling of the central carbon metabolism of Escherichia coli.* Biotechnology and bioengineering, vol. 79, no. 1, pages 53–73, 2002. (Cited on page 6.)

[ChEBI 2017] ChEBI. *ChEBI.* https://www.ebi.ac.uk/chebi/, 2017. Accessed: 2017-01-10. (Cited on page 6.)

[Chen & Nielsen 2013] Yun Chen and Jens Nielsen. *Advances in metabolic pathway and strain engineering paving the way for sustainable production of chemical building blocks.* Current opinion in biotechnology, vol. 24, no. 6, pages 965–972, 2013. (Cited on page 1.)

[Choon *et al.* 2014] Yee Wen Choon, Mohd Saberi Mohamad, Safaai Deris, Rosli Md Illias, Chuii Khim Chong and Lian En Chai. *A hybrid of bees algorithm and flux balance analysis with OptKnock as a platform for in silico optimization of microbial strains.* Bioprocess and biosystems engineering, vol. 37, no. 3, pages 521–532, 2014. (Cited on pages 3, 64 and 71.)

[Chotani *et al.* 2000] Gopal Chotani, Tim Dodge, Amy Hsu, Manoj Kumar, Richard LaDuca, Donald Trimbur, Walter Weyler and Karl Sanford. *The commercial production of chemicals using pathway engineering.* Biochimica et Biophysica Acta (BBA)-Protein Structure and Molecular Enzymology, vol. 1543, no. 2, pages 434–455, 2000. (Cited on page 1.)

[Clarke 1988] Bruce L Clarke. *Stoichiometric network analysis.* Cell Biochemistry and Biophysics, vol. 12, no. 1, pages 237–253, 1988. (Cited on pages 6 and 9.)

[Clerc & Kennedy 2002] Maurice Clerc and James Kennedy. *The particle swarm-explosion, stability, and convergence in a multidimensional complex space.* Evolutionary Computation, IEEE Transactions on, vol. 6, no. 1, pages 58–73, 2002. (Cited on page 68.)

[Cornuéjols *et al.* 2006] Gérard Cornuéjols, Miroslav Karamanov and Yanjun Li. *Early estimates of the size of branch-and-bound trees.* INFORMS Journal on Computing, vol. 18, no. 1, pages 86–96, 2006. (Cited on page 73.)

[Covert *et al.* 2001] Markus W Covert, Christophe H Schilling, Iman Famili, Jeremy S Edwards, Igor I Goryanin, Evgeni Selkov and Bernhard O Palsson. *Metabolic modeling of microbial strains in silico.* Trends in biochemical sciences, vol. 26, no. 3, pages 179–186, 2001. (Cited on page 40.)

[CPLEX 2017] CPLEX. *CPLEX*, accessed February 17, 2017. http://www-03.ibm.com/software/products/en/ibmilogcpleoptistud/. (Cited on page 12.)

[David & Bockmayr 2014] Laszlo David and Alexander Bockmayr. *Computing elementary flux modes involving a set of target reactions.* IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB), vol. 11, no. 6, 2014. (Cited on pages 36 and 40.)

[De Figueiredo *et al.* 2009] Luis F De Figueiredo, Adam Podhorski, Angel Rubio, Christoph Kaleta, John E Beasley, Stefan Schuster and Francisco J Planes. *Computing the shortest elementary flux modes in genome-scale metabolic networks.* Bioinformatics, vol. 25, no. 23, pages 3158–3165, 2009. (Cited on pages 15, 22 and 36.)

[Degtyarenko *et al.* 2008] Kirill Degtyarenko, Paula De Matos, Marcus Ennis, Janna Hastings, Martin Zbinden, Alan McNaught, Rafael Alcántara, Michael Darsow, Mickaël Guedj and Michael Ashburner. *ChEBI: a database and ontology for chemical entities of biological interest.* Nucleic acids research, vol. 36, no. suppl 1, pages D344–D350, 2008. (Cited on page 6.)

[Del Valle *et al.* 2008] Yamille Del Valle, Ganesh Kumar Venayagamoorthy, Salman Mohagheghi, Jean-Carlos Hernandez and Ronald G Harley. *Particle swarm optimization: basic concepts, variants and applications in power systems.* Evolutionary Computation, IEEE Transactions on, vol. 12, no. 2, pages 171–195, 2008. (Cited on pages 67 and 68.)

[Di Ventura *et al.* 2006] Barbara Di Ventura, Caroline Lemerle, Konstantinos Michalodimitrakis and Luis Serrano. *From in vivo to in silico biology and back.* Nature, vol. 443, no. 7111, pages 527–533, 2006. (Cited on page 2.)

[Durot *et al.* 2009] Maxime Durot, Pierre-Yves Bourguignon and Vincent Schachter. *Genome-scale models of bacterial metabolism: reconstruction and applications.* FEMS microbiology reviews, vol. 33, no. 1, pages 164–190, 2009. (Cited on page 40.)

[Ellis *et al.* 2009] Tom Ellis, Xiao Wang and James J Collins. *Diversity-based, model-guided construction of synthetic gene networks with predicted functions.* Nature biotechnology, vol. 27, no. 5, pages 465–471, 2009. (Cited on page 2.)

[ENZYME 2017] ENZYME. *ENZYME.* http://enzyme.expasy.org/, 2017. Accessed: 2017-01-10. (Cited on page 5.)

[ERGO 2017] ERGO. *ERGO 2.0 - Igenbio.* http://www.igenbio.com/ergo/, 2017. Accessed: 2017-01-10. (Cited on page 5.)

[ExplorEnz 2017] ExplorEnz. *ExplorEnz.* http://www.enzyme-database.org/, 2017. Accessed: 2017-01-10. (Cited on page 6.)

[Fabregat *et al.* 2016] Antonio Fabregat, Konstantinos Sidiropoulos, Phani Garapati, Marc Gillespie, Kerstin Hausmann, Robin Haw, Bijay Jassal, Steven Jupe, Florian Korninger, Sheldon McKay*et al. The Reactome pathway knowledgebase.* Nucleic acids research, vol. 44, no. D1, pages D481–D487, 2016. (Cited on page 5.)

[Farkas 1902] Julius Farkas. *Theorie der einfachen Ungleichungen.* Journal für die reine und angewandte Mathematik, vol. 124, pages 1–27, 1902. (Cited on page 20.)

[Feist *et al.* 2007] Adam M Feist, Christopher S Henry, Jennifer L Reed, Markus Krummenacker, Andrew R Joyce, Peter D Karp, Linda J Broadbelt, Vassily Hatzimanikatis and Bernhard Ø Palsson. *A genome-scale metabolic reconstruction for Escherichia coli K-12 MG1655 that accounts for 1260 ORFs and thermodynamic information.* Molecular systems biology, vol. 3, no. 121, 2007. (Cited on pages 3 and 70.)

[Feist *et al.* 2009] Adam M Feist, Markus J Herrgård, Ines Thiele, Jennie L Reed and Bernhard Ø Palsson. *Reconstruction of biochemical networks in microorganisms.* Nature Reviews Microbiology, vol. 7, no. 2, pages 129–143, 2009. (Cited on page 2.)

[Feist *et al.* 2010] Adam M Feist, Daniel C Zielinski, Jeffrey D Orth, Jan Schellenberger, Markus J Herrgard and Bernhard Ø Palsson. *Model-driven evaluation of the production potential for growth-coupled products of Escherichia coli.* Metabolic engineering, vol. 12, no. 3, pages 173–186, 2010. (Cited on pages 47 and 50.)

[Fell & Small 1986] David A Fell and J Rankin Small. *Fat synthesis in adipose tissue. An examination of stoichiometric constraints.* Biochemical Journal, vol. 238, no. 3, pages 781–786, 1986. (Cited on page 9.)

[Fleischmann *et al.* 1995] Robert D Fleischmann, Mark D Adams, Owen White, Rebecca A Clayton*et al. Whole-genome random sequencing and assembly of Haemophilus influenzae Rd.* Science, vol. 269, no. 5223, page 496, 1995. (Cited on page 2.)

[Fleischmann *et al.* 2004] Astrid Fleischmann, Michael Darsow, Kirill Degtyarenko, Wolfgang Fleischmann, Sinéad Boyce, Kristian B Axelsen, Amos Bairoch, Dietmar Schomburg, Keith F Tipton and Rolf Apweiler. *IntEnz, the integrated relational enzyme database.* Nucleic acids research, vol. 32, no. suppl 1, pages D434–D437, 2004. (Cited on page 6.)

[Fong *et al.* 2005] Stephen S Fong, Anthony P Burgard, Christopher D Herring, Eric M Knight, Frederick R Blattner, Costas D Maranas and Bernhard O Palsson. *In silico design and adaptive evolution of Escherichia coli for production of lactic acid.* Biotechnology and bioengineering, vol. 91, no. 5, pages 643–648, 2005. (Cited on page 12.)

[Förster *et al.* 2003] Jochen Förster, Iman Famili, Patrick Fu, Bernhard Ø Palsson and Jens Nielsen. *Genome-scale reconstruction of the Saccharomyces cerevisiae metabolic network.* Genome research, vol. 13, no. 2, pages 244–253, 2003. (Cited on page 3.)

[Fredman & Khachiyan 1996] Michael L Fredman and Leonid Khachiyan. *On the complexity of dualization of monotone disjunctive normal forms.* Journal of Algorithms, vol. 21, no. 3, pages 618–628, 1996. (Cited on page 19.)

[Fukuda & Prodon 1996] Komei Fukuda and Alain Prodon. *Double description method revisited.* In Combinatorics and computer science, pages 91–111. Springer, 1996. (Cited on page 14.)

[Gagneur & Klamt 2004] Julien Gagneur and Steffen Klamt. *Computation of elementary modes: a unifying framework and the new binary approach.* BMC bioinformatics, vol. 5, no. 1, page 175, 2004. (Cited on pages 14 and 40.)

[Ganter *et al.* 2013] Mathias Ganter, Thomas Bernard, Sébastien Moretti, Joerg Stelling and Marco Pagni. *MetaNetX. org: a website and repository*

for accessing, analysing and manipulating metabolic networks. Bioinformatics, page btt036, 2013. (Cited on page 6.)

[Gerstl *et al.* 2015a] Matthias P Gerstl, Christian Jungreuthmayer and Jürgen Zanghellini. *tEFMA: computing thermodynamically feasible elementary flux modes in metabolic networks.* Bioinformatics, page btv111, 2015. (Cited on pages 15 and 36.)

[Gerstl *et al.* 2015b] Matthias P Gerstl, David E Ruckerbauer, Diethard Mattanovich, Christian Jungreuthmayer and Jürgen Zanghellini. *Metabolomics integrated elementary flux mode analysis in large metabolic networks.* Scientific reports, vol. 5, page 8930, 2015. (Cited on page 15.)

[Gleeson & Ryan 1990] John Gleeson and Jennifer Ryan. *Identifying minimally infeasible subsystems of inequalities.* ORSA Journal on Computing, vol. 2, no. 1, pages 61–63, 1990. (Cited on page 20.)

[GLPK 2017] GLPK. *GLPK (GNU Linear Programming Kit)*, accessed February 17, 2017. `https://www.gnu.org/software/glpk/`. (Cited on page 12.)

[Golberg 1989] DE Golberg. Genetic algorithms in search, optimization and machine learning reading. MA: Addisonn-Wisley, USA, 1989. (Cited on page 26.)

[Goldberg & Deb 1991] David E Goldberg and Kalyanmoy Deb. *A comparative analysis of selection schemes used in genetic algorithms.* Urbana, vol. 51, pages 61801–2996, 1991. (Cited on page 44.)

[Gong & Zhang 2013] Yue-jiao Gong and Jun Zhang. *Small-world particle swarm optimization with topology adaptation.* In Proceedings of the 15th annual conference on Genetic and evolutionary computation, Amsterdam, Netherlands, pages 25–32. ACM, New York, NY, United States, 2013. (Cited on page 68.)

[Hädicke & Klamt 2011] Oliver Hädicke and Steffen Klamt. *Computing complex metabolic intervention strategies using constrained minimal cut sets.* Metabolic engineering, vol. 13, no. 2, pages 204–213, 2011. (Cited on pages 3, 7, 17, 40, 42, 43 and 64.)

[Harder *et al.* 2016] Björn-Johannes Harder, Katja Bettenbrock and Steffen Klamt. *Model-based metabolic engineering enables high yield itaconic acid production by Escherichia coli.* Metabolic engineering, vol. 38, pages 29–37, 2016. (Cited on page 22.)

[Hecker *et al.* 2009] Michael Hecker, Sandro Lambeck, Susanne Toepfer, Eugene Van Someren and Reinhard Guthke. *Gene regulatory network inference: data integration in dynamic modelsâa review.* Biosystems, vol. 96, no. 1, pages 86–103, 2009. (Cited on page 2.)

[Henry *et al.* 2009] Christopher S Henry, Jenifer F Zinner, Matthew P Cohoon and Rick L Stevens. *i Bsu1103: a new genome-scale metabolic model of Bacillus subtilis based on SEED annotations.* Genome biology, vol. 10, no. 6, page 1, 2009. (Cited on page 3.)

[Henry *et al.* 2010] Christopher S Henry, Matthew DeJongh, Aaron A Best, Paul M Frybarger, Ben Linsay and Rick L Stevens. *High-throughput generation, optimization and analysis of genome-scale metabolic models.* Nature biotechnology, vol. 28, no. 9, pages 977–982, 2010. (Cited on page 40.)

[Hjersted *et al.* 2007] Jared L Hjersted, Michael A Henson and Radhakrishnan Mahadevan. *Genome-scale analysis of Saccharomyces cerevisiae metabolism and ethanol production in fed-batch culture.* Biotechnology and bioengineering, vol. 97, no. 5, pages 1190–1204, 2007. (Cited on page 12.)

[Holland 1975] John H Holland. *Adaptation in natural and artificial systems. An introductory analysis with application to biology, control, and artificial intelligence.* Ann Arbor, MI: University of Michigan Press, 1975. (Cited on page 26.)

[Hoppe *et al.* 2011] Andreas Hoppe, Sabrina Hoffmann, Andreas Gerasch, Christoph Gille and Hermann-Georg Holzhütter. *FASIMU: flexible software for flux-balance computation series in large metabolic networks.* BMC bioinformatics, vol. 12, no. 1, page 28, 2011. (Cited on page 9.)

[Hunt *et al.* 2014] Kristopher A Hunt, James P Folsom, Reed L Taffs and Ross P Carlson. *Complete enumeration of elementary flux modes through scalable, demand-based subnetwork definition.* Bioinformatics, page btu021, 2014. (Cited on page 15.)

[Ibarra *et al.* 2002] Rafael U Ibarra, Jeremy S Edwards and Bernhard O Palsson. *Escherichia coli K-12 undergoes adaptive evolution to achieve in silico predicted optimal growth.* Nature, vol. 420, no. 6912, pages 186–189, 2002. (Cited on page 65.)

[IntEnz 2017] IntEnz. *IntEnz*. http://www.ebi.ac.uk/intenz/, 2017. Accessed: 2017-01-10. (Cited on page 6.)

[Jevremović *et al.* 2011] Dimitrije Jevremović, Cong T Trinh, Friedrich Srienc, Carlos P Sosa and Daniel Boley. *Parallelization of nullspace algorithm for the computation of metabolic pathways*. Parallel computing, vol. 37, no. 6, pages 261–278, 2011. (Cited on page 15.)

[Jian *et al.* 2016] Xingxing Jian, Shengguo Zhou, Cheng Zhang and Qiang Hua. *In silico identification of gene amplification targets based on analysis of production and growth coupling*. Biosystems, vol. 145, pages 1–8, 2016. (Cited on page 36.)

[Joshi & Palsson 1989] Abhay Joshi and Bernhard O Palsson. *Metabolic dynamics in the human red cell: Part IâA comprehensive kinetic model*. Journal of theoretical biology, vol. 141, no. 4, pages 515–528, 1989. (Cited on page 6.)

[Jungreuthmayer & Zanghellini 2012] Christian Jungreuthmayer and Jürgen Zanghellini. *Designing optimal cell factories: integer programming couples elementary mode analysis with regulation*. BMC systems biology, vol. 6, no. 103, 2012. (Cited on pages 17, 40, 43 and 50.)

[Jungreuthmayer *et al.* 2013a] Christian Jungreuthmayer, Marie Beurton-Aimar and Jürgen Zanghellini. *Fast computation of minimal cut sets in metabolic networks with a Berge algorithm that utilizes binary bit pattern trees*. IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB), vol. 10, no. 5, page 1, 2013. (Cited on pages 18, 44, 46, 49 and 50.)

[Jungreuthmayer *et al.* 2013b] Christian Jungreuthmayer, Govind Nair, Steffen Klamt and Jürgen Zanghellini. *Comparison and improvement of algorithms for computing minimal cut sets*. BMC bioinformatics, vol. 14, no. 318, 2013. (Cited on pages 17, 40, 43 and 64.)

[Jungreuthmayer *et al.* 2013c] Christian Jungreuthmayer, David E Ruckerbauer and Jürgen Zanghellini. *regEfmtool: Speeding up elementary flux mode calculation using transcriptional regulatory rules in the form of three-state logic*. Biosystems, vol. 113, no. 1, pages 37–39, 2013. (Cited on pages 15, 36, 40 and 46.)

[Jungreuthmayer *et al.* 2013d] Christian Jungreuthmayer, Margot Sonnleitner, Gerald Striedner, Jürgen Mairhofer and Jürgen Zanghellini. *Designing an optimally ethanol producing E. coli strain using constrained*

minimal cut sets. In Proceedings of the 21st European signal processing conference, September 2013. (Cited on page 41.)

[Kaleta *et al.* 2009] Christoph Kaleta, LuiÌs Filipe De Figueiredo, Jörn Behre and Stefan Schuster. *EFMEvolver: Computing elementary flux modes in genome-scale metabolic networks*. Lect Notes Inform, pages 179–89, 2009. (Cited on pages 15 and 36.)

[Kanehisa *et al.* 2004] Minoru Kanehisa, Susumu Goto, Shuichi Kawashima, Yasushi Okuno and Masahiro Hattori. *The KEGG resource for deciphering the genome*. Nucleic acids research, vol. 32, no. suppl 1, pages D277–D280, 2004. (Cited on page 5.)

[Kennedy & Eberhart 1995] James Kennedy and Russell C Eberhart. *Particle Swarm Optimization*. In Proceedings of the IEEE International Conference on Neural Networks, Perth, Australia, volume 4, pages 1942–1948. IEEE, Piscataway, NJ, United States, 1995. (Cited on pages 27 and 67.)

[Kennedy & Mendes 2002] James Kennedy and Rui Mendes. *Population structure and particle swarm performance*. In Proceedings of the IEEE congress on evolutionary computation (CEC), Honolulu, HI, volume 4, pages 1671–1676. IEEE, Piscataway, NJ, United States, 2002. (Cited on page 68.)

[Kim & Reed 2010] Joonhoon Kim and Jennifer L Reed. *OptORF: Optimal metabolic and regulatory perturbations for metabolic engineering of microbial strains*. BMC systems biology, vol. 4, no. 1, page 53, 2010. (Cited on pages 3 and 36.)

[Kitano 2002] Hiroaki Kitano. *Systems biology: a brief overview*. Science, vol. 295, no. 5560, pages 1662–1664, 2002. (Cited on page 2.)

[Klamt & Gilles 2004] Steffen Klamt and Ernst Dieter Gilles. *Minimal cut sets in biochemical reaction networks*. Bioinformatics, vol. 20, no. 2, pages 226–234, 2004. (Cited on pages 3, 17, 42 and 64.)

[Klamt & Mahadevan 2015] Steffen Klamt and Radhakrishnan Mahadevan. *On the feasibility of growth-coupled product synthesis in microbial strains*. Metabolic engineering, vol. 30, pages 166–178, 2015. (Cited on page 72.)

[Klamt & Stelling 2002] Steffen Klamt and Jörg Stelling. *Combinatorial complexity of pathway analysis in metabolic networks*. Molecular biology

reports, vol. 29, no. 1-2, pages 233–236, 2002. (Cited on pages 15, 19 and 40.)

[Klamt *et al.* 2005] Steffen Klamt, Julien Gagneur and Axel von Kamp. *Algorithmic approaches for computing elementary modes in large biochemical reaction networks.* IEE Proceedings-Systems Biology, vol. 152, no. 4, pages 249–255, 2005. (Cited on page 14.)

[Klamt *et al.* 2007] Steffen Klamt, Julio Saez-Rodriguez and Ernst D Gilles. *Structural and functional analysis of cellular networks with CellNetAnalyzer.* BMC systems biology, vol. 1, no. 1, page 2, 2007. (Cited on page 50.)

[Klamt 2006] Steffen Klamt. *Generalized concept of minimal cut sets in biochemical networks.* Biosystems, vol. 83, no. 2, pages 233–247, 2006. (Cited on pages 3, 17 and 19.)

[Klotz & Newman 2013] Ed Klotz and Alexandra M Newman. *Practical guidelines for solving difficult mixed integer linear programs.* Surveys in Operations Research and Management Science, vol. 18, no. 1, pages 18–32, 2013. (Cited on pages 25 and 73.)

[Lange *et al.* 2016] Julian Lange, Ralf Takors and Bastian Blombach. *Zero-growth bioprocesses: A challenge for microbial production strains and bioprocess engineering.* Engineering in Life Sciences, vol. 16, no. 8, pages n/a–n/a, 2016. (Cited on pages 36 and 72.)

[Le Novere *et al.* 2006] Nicolas Le Novere, Benjamin Bornstein, Alexander Broicher, Melanie Courtot, Marco Donizelli, Harish Dharuri, Lu Li, Herbert Sauro, Maria Schilstra, Bruce Shapiro*et al.* *BioModels Database: a free, centralized database of curated, published, quantitative kinetic models of biochemical and cellular systems.* Nucleic acids research, vol. 34, no. suppl 1, pages D689–D691, 2006. (Cited on page 6.)

[Lee & Schmidt-Dannert 2002] P Lee and C Schmidt-Dannert. *Metabolic engineering towards biotechnological production of carotenoids in microorganisms.* Applied Microbiology and Biotechnology, vol. 60, no. 1-2, pages 1–11, 2002. (Cited on page 1.)

[Lee *et al.* 2005] Sang Jun Lee, Dong-Yup Lee, Tae Yong Kim, Byung Hun Kim, Jinwon Lee and Sang Yup Lee. *Metabolic engineering of Escherichia coli for enhanced production of succinic acid, based on genome comparison and in silico gene knockout simulation.* Applied

and environmental microbiology, vol. 71, no. 12, pages 7880–7887, 2005. (Cited on page 12.)

[Lee *et al.* 2007] Kwang Ho Lee, Jin Hwan Park, Tae Yong Kim, Hyun Uk Kim and Sang Yup Lee. *Systems metabolic engineering of Escherichia coli for L-threonine production.* Molecular systems biology, vol. 3, no. 1, page 149, 2007. (Cited on page 12.)

[Lee *et al.* 2008] Sung Kuk Lee, Howard Chou, Timothy S Ham, Taek Soon Lee and Jay D Keasling. *Metabolic engineering of microorganisms for biofuels production: from bugs to synthetic biology to fuels.* Current opinion in biotechnology, vol. 19, no. 6, pages 556–563, 2008. (Cited on page 1.)

[Li & Yunfei 2002] Lin Li and Jiang Yunfei. *Computing minimal hitting sets with genetic algorithm.* Technical report, DTIC Document, 2002. (Cited on page 44.)

[Lima & Grossmann 2011] Ricardo M Lima and Ignacio E Grossmann. *Computational advances in solving mixed integer linear programming problems.* 2011. (Cited on page 25.)

[Liu *et al.* 2004] Hong Liu, Ramanathan Ramnarayanan and Bruce E Logan. *Production of electricity during wastewater treatment using a single chamber microbial fuel cell.* Environmental science & technology, vol. 38, no. 7, pages 2281–2285, 2004. (Cited on page 1.)

[Llaneras & Picó 2008] Francisco Llaneras and Jesús Picó. *Stoichiometric modelling of cell metabolism.* Journal of Bioscience and Bioengineering, vol. 105, no. 1, pages 1–11, 2008. (Cited on pages 10 and 11.)

[Lloyd *et al.* 2008] Catherine M Lloyd, James R Lawson, Peter J Hunter and Poul F Nielsen. *The CellML model repository.* Bioinformatics, vol. 24, no. 18, pages 2122–2123, 2008. (Cited on page 6.)

[Machado *et al.* 2012] Daniel Machado, Zita Soons, Kiran Raosaheb Patil, Eugénio C Ferreira and Isabel Rocha. *Random sampling of elementary flux modes in large-scale metabolic networks.* Bioinformatics, vol. 28, no. 18, pages i515–i521, 2012. (Cited on page 36.)

[Mahadevan & Schilling 2003] R Mahadevan and CH Schilling. *The effects of alternate optimal solutions in constraint-based genome-scale metabolic models.* Metabolic engineering, vol. 5, no. 4, pages 264–276, 2003. (Cited on pages 9, 13 and 70.)

[Mahadevan *et al.* 2015] Radhakrishnan Mahadevan, Axel von Kamp and Steffen Klamt. *Genome-scale strain designs based on regulatory minimal cut sets.* Bioinformatics, vol. 31, pages 2844–2851, 2015. (Cited on pages 3, 21, 36, 40, 50, 52, 64, 65, 66, 70, 71 and 73.)

[Martin *et al.* 2003] Vincent JJ Martin, Douglas J Pitera, Sydnor T Withers, Jack D Newman and Jay D Keasling. *Engineering a mevalonate pathway in Escherichia coli for production of terpenoids.* Nature biotechnology, vol. 21, no. 7, pages 796–802, 2003. (Cited on page 1.)

[MATLAB 2017] MATLAB. *MATLAB*, accessed February 17, 2017. https://www.mathworks.com/products/matlab.html. (Cited on page 12.)

[McDonald *et al.* 2009] Andrew G McDonald, Sinead Boyce and Keith F Tipton. *ExplorEnz: the primary source of the IUBMB enzyme list.* Nucleic acids research, vol. 37, no. suppl 1, pages D593–D597, 2009. (Cited on page 6.)

[MetaNetX 2017] MetaNetX. *MetaNetX*, accessed February 17, 2017. http://www.metanetx.org/. (Cited on page 6.)

[metaTIGER 2017] metaTIGER. *metaTIGER*. http://www.bioinformatics.leeds.ac.uk/metatiger/, 2017. Accessed: 2017-01-10. (Cited on page 5.)

[Mi *et al.* 2009] Huaiyu Mi, Qing Dong, Anushya Muruganujan, Pascale Gaudet, Suzanna Lewis and Paul D Thomas. *PANTHER version 7: improved phylogenetic trees, orthologs and collaboration with the Gene Ontology Consortium.* Nucleic acids research, page gkp1019, 2009. (Cited on page 6.)

[Milner 1964] Paul C Milner. *The possible mechanisms of complex reactions involving consecutive steps.* Journal of the Electrochemical Society, vol. 111, no. 2, pages 228–232, 1964. (Cited on page 6.)

[Misawa *et al.* 1991] Norihiko Misawa, SHIGEYUKI Yamano and HIROSHI Ikenaga. *Production of beta-carotene in Zymomonas mobilis and Agrobacterium tumefaciens by introduction of the biosynthesis genes from Erwinia uredovora.* Applied and environmental microbiology, vol. 57, no. 6, pages 1847–1849, 1991. (Cited on page 1.)

[Mitchell 1998] Melanie Mitchell. An introduction to genetic algorithms. MIT press, 1998. (Cited on pages 26 and 44.)

[Mullis *et al.* 1987] Kary B Mullis, Henry A Erlich, Norman Arnheim, Glenn T Horn, Randall K Saiki and Stephen J Scharf. *Process for amplifying, detecting, and/or-cloning nucleic acid sequences*, July 28 1987. US Patent 4,683,195. (Cited on page 4.)

[Nair *et al.* 2015] Govind Nair, Christian Jungreuthmayer, Michael Hanscho and Jürgen Zanghellini. *Designing minimal microbial strains of desired functionality using a genetic algorithm.* Algorithms for Molecular Biology, vol. 10, no. 1, page 1, 2015. (Cited on pages 64, 69, 70 and 71.)

[Nakamura & Whited 2003] Charles E Nakamura and Gregory M Whited. *Metabolic engineering for the microbial production of 1, 3-propanediol.* Current opinion in biotechnology, vol. 14, no. 5, pages 454–459, 2003. (Cited on page 1.)

[Oberhardt *et al.* 2009] Matthew A Oberhardt, Bernhard Ø Palsson and Jason A Papin. *Applications of genome-scale metabolic reconstructions.* Molecular systems biology, vol. 5, no. 1, 2009. (Cited on page 40.)

[Orth *et al.* 2010] Jeffrey D Orth, Ines Thiele and Bernhard Ø Palsson. *What is flux balance analysis?* Nature biotechnology, vol. 28, no. 3, pages 245–248, 2010. (Cited on page 40.)

[Paddon & Keasling 2014] Chris J Paddon and Jay D Keasling. *Semi-synthetic artemisinin: a model for the use of synthetic biology in pharmaceutical development.* Nature Reviews Microbiology, vol. 12, no. 5, pages 355–367, 2014. (Cited on page 1.)

[Palsson 2015] Bernhard Palsson. Systems biology. Cambridge university press, 2015. (Cited on page 12.)

[PANTHER 2017] PANTHER. *Protein ANalysis THrough Evolutionary Relationships (PANTHER).* `http://www.pantherdb.org/`, 2017. Accessed: 2017-01-10. (Cited on page 6.)

[Papin *et al.* 2005] Jason A Papin, Tony Hunter, Bernhard O Palsson and Shankar Subramaniam. *Reconstruction of cellular signalling networks and analysis of their properties.* Nature reviews Molecular cell biology, vol. 6, no. 2, pages 99–111, 2005. (Cited on page 2.)

[Papoutsakis 1984] Eleftherios Terry Papoutsakis. *Equations and calculations for fermentations of butyric acid bacteria.* Biotechnology and bioengineering, vol. 26, no. 2, pages 174–187, 1984. (Cited on page 9.)

[Park *et al.* 2007] Jin Hwan Park, Kwang Ho Lee, Tae Yong Kim and Sang Yup Lee. *Metabolic engineering of Escherichia coli for the production of L-valine based on transcriptome analysis and in silico gene knockout simulation.* Proceedings of the National Academy of Sciences, vol. 104, no. 19, pages 7797–7802, 2007. (Cited on page 12.)

[Park *et al.* 2012] Jong Myoung Park, Hye Min Park, Won Jun Kim, Hyun Uk Kim, Tae Yong Kim and Sang Yup Lee. *Flux variability scanning based on enforced objective flux for identifying gene amplification targets.* BMC systems biology, vol. 6, no. 1, page 106, 2012. (Cited on page 36.)

[Patil *et al.* 2005] Kiran R Patil, Isabel Rocha, Jochen Förster and Jens Nielsen. *Evolutionary programming as a platform for in silico metabolic engineering.* BMC bioinformatics, vol. 6, no. 308, 2005. (Cited on pages 3, 49, 64 and 71.)

[Peskov *et al.* 2012] Kirill Peskov, Ekaterina Mogilevskaya and Oleg Demin. *Kinetic modelling of central carbon metabolism in Escherichia coli.* Febs Journal, vol. 279, no. 18, pages 3374–3385, 2012. (Cited on page 6.)

[Pey & Planes 2014] Jon Pey and Francisco J Planes. *Direct calculation of elementary flux modes satisfying several biological constraints in genome-scale metabolic networks.* Bioinformatics, page btu193, 2014. (Cited on page 15.)

[Pey *et al.* 2014] Jon Pey, Juan A Villar, Luis Tobalina, Alberto Rezola, José Manuel García, John E Beasley and Francisco J Planes. *TreeEFM: calculating elementary flux modes using linear optimization in a tree-based algorithm.* Bioinformatics, page btu733, 2014. (Cited on page 15.)

[Poli *et al.* 2007] Riccardo Poli, James Kennedy and Tim Blackwell. *Particle swarm optimization.* Swarm intelligence, vol. 1, no. 1, pages 33–57, 2007. (Cited on pages 27, 28, 67 and 68.)

[Poolman *et al.* 2009] Mark G Poolman, Laurent Miguet, Lee J Sweetlove and David A Fell. *A genome-scale metabolic model of Arabidopsis and some of its properties.* Plant physiology, vol. 151, no. 3, pages 1570–1581, 2009. (Cited on page 3.)

[Price *et al.* 2003] Nathan D Price, Jason A Papin, Christophe H Schilling and Bernhard O Palsson. *Genome-scale microbial in silico models: the*

constraints-based approach. Trends in biotechnology, vol. 21, no. 4, pages 162–169, 2003. (Cited on page 3.)

[PubChem 2017] PubChem. *PubChem.* https://pubchem.ncbi.nlm.nih.gov/, 2017. Accessed: 2017-01-10. (Cited on page 5.)

[Purnick & Weiss 2009] Priscilla EM Purnick and Ron Weiss. *The second wave of synthetic biology: from modules to systems.* Nature reviews Molecular cell biology, vol. 10, no. 6, pages 410–422, 2009. (Cited on page 2.)

[Quek & Nielsen 2014] Lake-Ee Quek and Lars K Nielsen. *A depth-first search algorithm to compute elementary flux modes by linear programming.* BMC systems biology, vol. 8, no. 1, page 94, 2014. (Cited on page 15.)

[Ranganathan *et al.* 2010] Sridhar Ranganathan, Patrick F Suthers and Costas D Maranas. *OptForce: an optimization procedure for identifying all genetic manipulations leading to targeted overproductions.* PLoS Comput Biol, vol. 6, no. 4, page e1000744, 2010. (Cited on page 36.)

[Rebnegger *et al.* 2016] Corinna Rebnegger, Tim Vos, Alexandra B Graf, Minoska Valli, Jack T Pronk, Pascale Daran-Lapujade and Diethard Mattanovich. *Pichia pastoris exhibits high viability and low maintenance-energy requirement at near-zero specific growth rates.* Applied and environmental microbiology, vol. 82, no. 15, pages 4570–4583, 2016. (Cited on pages 36 and 72.)

[Reed *et al.* 2006] Jennifer L Reed, Iman Famili, Ines Thiele and Bernhard O Palsson. *Towards multidimensional genome annotation.* Nature Reviews Genetics, vol. 7, no. 2, pages 130–141, 2006. (Cited on page 35.)

[Rezola *et al.* 2011] Alberto Rezola, Luis F de Figueiredo, M Brock, Jon Pey, Adam Podhorski, Christoph Wittmann, Stefan Schuster, Alexander Bockmayr and Francisco J Planes. *Exploring metabolic pathways in genome-scale networks via generating flux modes.* Bioinformatics, vol. 27, no. 4, pages 534–540, 2011. (Cited on page 15.)

[Rocha *et al.* 2008] Miguel Rocha, Paulo Maia, Rui Mendes, José P Pinto, Eugénio C Ferreira, Jens Nielsen, Kiran Raosaheb Patil and Isabel Rocha. *Natural computation meta-heuristics for the in silico optimization of microbial strains.* BMC bioinformatics, vol. 9, no. 1, page 1, 2008. (Cited on pages 3, 64 and 71.)

[Ruckerbauer *et al.* 2014] David E Ruckerbauer, Christian Jungreuthmayer and Jürgen Zanghellini. *Design of optimally constructed metabolic networks of minimal functionality.* PloS one, vol. 9, no. 3, page e92583, 2014. (Cited on pages 29, 41, 43, 47, 49, 50, 59, 64 and 70.)

[Saa & Nielsen 2016] Pedro A Saa and Lars K Nielsen. *Construction of feasible and accurate kinetic models of metabolism: A Bayesian approach.* Scientific Reports, vol. 6, 2016. (Cited on page 6.)

[Savinell & Palsson 1992] Joanne M Savinell and Bernhard O Palsson. *Network analysis of intermediary metabolism using linear optimization. I. Development of mathematical formalism.* Journal of theoretical biology, vol. 154, no. 4, pages 421–454, 1992. (Cited on page 9.)

[Scheer *et al.* 2010] Maurice Scheer, Andreas Grote, Antje Chang, Ida Schomburg, Cornelia Munaretto, Michael Rother, Carola Söhngen, Michael Stelzer, Juliane Thiele and Dietmar Schomburg. *BRENDA, the enzyme information system in 2011.* Nucleic acids research, page gkq1089, 2010. (Cited on page 5.)

[Schellenberger *et al.* 2011] Jan Schellenberger, Richard Que, Ronan MT Fleming, Ines Thiele, Jeffrey D Orth, Adam M Feist, Daniel C Zielinski, Aarash Bordbar, Nathan E Lewis, Sorena Rahmanian*et al. Quantitative prediction of cellular metabolism with constraint-based models: the COBRA Toolbox v2. 0.* Nature protocols, vol. 6, no. 9, pages 1290–1307, 2011. (Cited on pages 9 and 12.)

[Schilling *et al.* 2000] Christophe H Schilling, David Letscher and Bernhard Ø Palsson. *Theory for the systemic definition of metabolic pathways and their use in interpreting metabolic function from a pathway-oriented perspective.* Journal of theoretical biology, vol. 203, no. 3, pages 229–248, 2000. (Cited on page 3.)

[Schilling *et al.* 2002] Christophe H Schilling, Markus W Covert, Iman Famili, George M Church, Jeremy S Edwards and Bernhard O Palsson. *Genome-scale metabolic model of Helicobacter pylori 26695.* Journal of bacteriology, vol. 184, no. 16, pages 4582–4593, 2002. (Cited on page 3.)

[Schuster & Hilgetag 1994] Stefan Schuster and Claus Hilgetag. *On elementary flux modes in biochemical reaction systems at steady state.* Journal of Biological Systems, vol. 2, no. 02, pages 165–182, 1994. (Cited on pages 3, 13 and 40.)

[Schuster & Schuster 1993] R Schuster and Stefan Schuster. *Refined algorithm and computer program for calculating all non–negative fluxes admissible in steady states of biochemical reaction systems with or without some flux rates fixed.* Computer applications in the biosciences: CABIOS, vol. 9, no. 1, pages 79–85, 1993. (Cited on page 9.)

[Schuster *et al.* 2000] Stefan Schuster, David A Fell and Thomas Dandekar. *A general definition of metabolic pathways useful for systematic organization and analysis of complex metabolic networks.* Nature biotechnology, vol. 18, no. 3, pages 326–332, 2000. (Cited on page 40.)

[Schuster *et al.* 2002] Stefan Schuster, Claus Hilgetag, John H Woods and David A Fell. *Reaction routes in biochemical reaction systems: algebraic properties, validated calculation procedure and example from nucleotide metabolism.* Journal of mathematical biology, vol. 45, no. 2, pages 153–181, 2002. (Cited on pages 3, 13 and 14.)

[Segre *et al.* 2002] Daniel Segre, Dennis Vitkup and George M Church. *Analysis of optimality in natural and perturbed metabolic networks.* Proceedings of the National Academy of Sciences, vol. 99, no. 23, pages 15112–15117, 2002. (Cited on pages 9 and 49.)

[Shendure & Ji 2008] Jay Shendure and Hanlee Ji. *Next-generation DNA sequencing.* Nature biotechnology, vol. 26, no. 10, pages 1135–1145, 2008. (Cited on page 4.)

[Shlomi *et al.* 2005] Tomer Shlomi, Omer Berkman and Eytan Ruppin. *Regulatory on/off minimization of metabolic flux changes after genetic perturbations.* Proceedings of the National Academy of Sciences of the United States of America, vol. 102, no. 21, pages 7695–7700, 2005. (Cited on page 9.)

[Smale 1998] Steve Smale. *Mathematical problems for the next century.* The Mathematical Intelligencer, vol. 20, no. 2, pages 7–15, 1998. (Cited on page 24.)

[Smallbone *et al.* 2013] Kieran Smallbone, Hanan L Messiha, Kathleen M Carroll, Catherine L Winder, Naglis Malys, Warwick B Dunn, Ettore Murabito, Neil Swainston, Joseph O Dada, Farid Khan*et al. A model of yeast glycolysis based on a consistent kinetic characterisation of all its enzymes.* FEBS letters, vol. 587, no. 17, pages 2832–2841, 2013. (Cited on page 6.)

[Smolke & Silver 2011] Christina D Smolke and Pamela A Silver. *Informing biological design by integration of systems and synthetic biology.* Cell, vol. 144, no. 6, pages 855–859, 2011. (Cited on page 2.)

[Srinivasan *et al.* 2015] Shyam Srinivasan, William R Cluett and Radhakrishnan Mahadevan. *Constructing kinetic models of metabolism at genome-scales: A review.* Biotechnology journal, vol. 10, no. 9, pages 1345–1359, 2015. (Cited on page 6.)

[Stelling *et al.* 2002] Jörg Stelling, Steffen Klamt, Katja Bettenbrock, Stefan Schuster and Ernst Dieter Gilles. *Metabolic network structure determines key aspects of functionality and regulation.* Nature, vol. 420, no. 6912, pages 190–193, 2002. (Cited on pages 9 and 64.)

[Stephanopoulos *et al.* 1998] George Stephanopoulos, Aristos A Aristidou and Jens Nielsen. Metabolic engineering: principles and methodologies. Academic press, 1998. (Cited on page 2.)

[Stephanopoulos 2012] Gregory Stephanopoulos. *Synthetic biology and metabolic engineering.* ACS synthetic biology, vol. 1, no. 11, pages 514–525, 2012. (Cited on page 1.)

[Tamura *et al.* 2012] Takeyuki Tamura, Kazuhiro Takemoto and Tatsuya Akutsu. *Finding minimum reaction cuts of metabolic networks under a Boolean model using integer programming and feedback vertex sets.* Comput Knowl Disco Bioinformatics Res, vol. 1, pages 240–258, 2012. (Cited on page 18.)

[Tenazinha & Vinga 2011] Nuno Tenazinha and Susana Vinga. *A survey on methods for modeling and analyzing integrated biological networks.* IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB), vol. 8, no. 4, pages 943–958, 2011. (Cited on page 40.)

[Tepper & Shlomi 2010] Naama Tepper and Tomer Shlomi. *Predicting metabolic engineering knockout strategies for chemical production: accounting for competing pathways.* Bioinformatics, vol. 26, no. 4, pages 536–543, 2010. (Cited on pages 3, 32, 50 and 64.)

[Terzer & Stelling 2006] Marco Terzer and Jörg Stelling. *Accelerating the computation of elementary modes using pattern trees.* In International Workshop on Algorithms in Bioinformatics, pages 333–343. Springer, 2006. (Cited on page 14.)

[Terzer & Stelling 2008] Marco Terzer and Jörg Stelling. *Large-scale computation of elementary flux modes with bit pattern trees.* Bioinformatics, vol. 24, no. 19, pages 2229–2235, 2008. (Cited on pages 14 and 40.)

[Teusink *et al.* 2000] Bas Teusink, Jutta Passarge, Corinne A Reijenga, Eugenia Esgalhado, Coen C van der Weijden, Mike Schepper, Michael C Walsh, Barbara M Bakker, Karel van Dam, Hans V Westerhoff*et al. Can yeast glycolysis be understood in terms of in vitro kinetics of the constituent enzymes? Testing biochemistry.* European Journal of Biochemistry, vol. 267, no. 17, pages 5313–5329, 2000. (Cited on page 6.)

[Thiele & Palsson 2010] Ines Thiele and Bernhard Ø Palsson. *A protocol for generating a high-quality genome-scale metabolic reconstruction.* Nature protocols, vol. 5, no. 1, pages 93–121, 2010. (Cited on page 40.)

[Tobalina *et al.* 2016] Luis Tobalina, Jon Pey and Francisco J Planes. *Direct calculation of minimal cut sets involving a specific reaction knock-out.* Bioinformatics, vol. 32, no. 13, pages 2001–2007, 2016. (Cited on page 22.)

[Tomar & De 2013] Namrata Tomar and Rajat K De. *Comparing methods for metabolic network analysis and an application to metabolic engineering.* Gene, vol. 521, no. 1, pages 1–14, 2013. (Cited on page 6.)

[Trinh & Srienc 2009] Cong T Trinh and Friedrich Srienc. *Metabolic engineering of Escherichia coli for efficient conversion of glycerol to ethanol.* Applied and environmental microbiology, vol. 75, no. 21, pages 6696–6705, 2009. (Cited on page 15.)

[Trinh *et al.* 2006] Cong T Trinh, Ross Carlson, Aaron Wlaschin and Friedrich Srienc. *Design, construction and performance of the most efficient biomass producing E. coli bacterium.* Metabolic engineering, vol. 8, no. 6, pages 628–638, 2006. (Cited on page 15.)

[Trinh *et al.* 2008] Cong T Trinh, Pornkamol Unrean and Friedrich Srienc. *Minimal Escherichia coli cell for the most efficient production of ethanol from hexoses and pentoses.* Applied and environmental microbiology, vol. 74, no. 12, pages 3634–3643, 2008. (Cited on pages 15, 29, 30, 32, 41, 47, 49, 51, 57, 69 and 72.)

[Trinh *et al.* 2011] Cong T Trinh, Johnny Li, Harvey W Blanch and Douglas S Clark. *Redesigning Escherichia coli metabolism for anaerobic production of isobutanol.* Applied and environmental microbiology, vol. 77, no. 14, pages 4894–4904, 2011. (Cited on page 15.)

[UniProt 2017] UniProt. *UniProt*. http://www.uniprot.org/, 2017. Accessed: 2017-01-10. (Cited on page 5.)

[Unrean *et al.* 2010] Pornkamol Unrean, Cong T Trinh and Friedrich Srienc. *Rational design and construction of an efficient E. coli for production of diapolycopendioic acid*. Metabolic engineering, vol. 12, no. 2, pages 112–122, 2010. (Cited on page 15.)

[Urbanczik & Wagner 2005] Robert Urbanczik and Carl Wagner. *An improved algorithm for stoichiometric network analysis: theory and applications*. Bioinformatics, vol. 21, no. 7, pages 1203–1210, 2005. (Cited on page 14.)

[Varma & Palsson 1993] Amit Varma and Bernhard O Palsson. *Metabolic capabilities of Escherichia coli: I. Synthesis of biosynthetic precursors and cofactors*. Journal of theoretical biology, vol. 165, no. 4, pages 477–502, 1993. (Cited on page 9.)

[Vasilakou *et al.* 2016] Eleni Vasilakou, Daniel Machado, Axel Theorell, Isabel Rocha, Katharina Nöh, Marco Oldiges and S Aljoscha Wahl. *Current state and challenges for dynamic metabolic modeling*. Current Opinion in Microbiology, vol. 33, pages 97–104, 2016. (Cited on page 6.)

[von Kamp & Klamt 2014] Axel von Kamp and Steffen Klamt. *Enumeration of smallest intervention strategies in genome-scale metabolic networks*. PLoS computational biology, vol. 10, no. 1, page e1003378, 2014. (Cited on pages 3, 19, 21, 22, 31, 40, 50, 51, 52, 64, 65, 66, 70 and 71.)

[Von Kamp & Schuster 2006] Axel Von Kamp and Stefan Schuster. *Metatool 5.0: fast and flexible elementary modes analysis*. Bioinformatics, vol. 22, no. 15, pages 1930–1931, 2006. (Cited on pages 14 and 22.)

[Watson 1984] MR Watson. *Metabolic maps for the Apple II*. Biochemical Society Transactions, vol. 12, no. 6, pages 1093–1094, 1984. (Cited on page 9.)

[Watson 1986] MR Watson. *A discrete model of bacterial metabolism*. Computer applications in the biosciences: CABIOS, vol. 2, no. 1, pages 23–27, 1986. (Cited on page 9.)

[Whitaker *et al.* 2009] John W Whitaker, Ivica Letunic, Glenn A McConkey and David R Westhead. *metaTIGER: a metabolic evolution resource*.

Nucleic acids research, vol. 37, no. suppl 1, pages D531–D538, 2009. (Cited on page 5.)

[Whitley 1994] Darrell Whitley. *A genetic algorithm tutorial*. Statistics and computing, vol. 4, no. 2, pages 65–85, 1994. (Cited on pages 26 and 44.)

[Wolpert & Macready 1997] David H Wolpert and William G Macready. *No free lunch theorems for optimization*. Evolutionary Computation, IEEE Transactions on, vol. 1, no. 1, pages 67–82, 1997. (Cited on page 72.)

[Xue *et al.* 2013] Zhixiong Xue, Pamela L Sharpe, Seung-Pyo Hong, Narendra S Yadav, Dongming Xie, David R Short, Howard G Damude, Ross A Rupert, John E Seip, Jamie Wang*et al. Production of omega-3 eicosapentaenoic acid by metabolic engineering of Yarrowia lipolytica*. Nature biotechnology, vol. 31, no. 8, pages 734–740, 2013. (Cited on page 1.)

[Yim *et al.* 2011] Harry Yim, Robert Haselbeck, Wei Niu, Catherine Pujol-Baxley, Anthony Burgard, Jeff Boldt, Julia Khandurina, John D Trawick, Robin E Osterhout, Rosary Stephen*et al. Metabolic engineering of Escherichia coli for direct production of 1, 4-butanediol*. Nature chemical biology, vol. 7, no. 7, pages 445–452, 2011. (Cited on page 1.)

[Zanghellini *et al.* 2013] Jürgen Zanghellini, David E Ruckerbauer, Michael Hanscho and Christian Jungreuthmayer. *Elementary flux modes in a nutshell: properties, calculation and applications*. Biotechnology journal, vol. 8, no. 9, pages 1009–1016, 2013. (Cited on page 42.)

[Zhang *et al.* 2015] Yudong Zhang, Shuihua Wang and Genlin Ji. *A comprehensive survey on particle swarm optimization algorithm and its applications*. Mathematical Problems in Engineering, vol. 2015, 2015. (Cited on page 28.)

[Zomorrodi *et al.* 2012] Ali R Zomorrodi, Patrick F Suthers, Sridhar Ranganathan and Costas D Maranas. *Mathematical optimization applications in metabolic networks*. Metabolic engineering, vol. 14, no. 6, pages 672–686, 2012. (Cited on page 3.)

Appendix A

Publications

- **Nair, G.**, Jungreuthmayer, C., Zanghellini, J. (2017) Optimal knockout strategies in genome-scale metabolic networks using particle swarm optimization. - BMC Bioinformatics, 18:78.
- Zanghellini, J., Gerstl, M. P., Hanscho, M., **Nair, G.**, Regensburger, G., Müller, S. and Jungreuthmayer, C. (2017) Toward Genome-Scale Metabolic Pathway Analysis, in Industrial Biotechnology: Microorganisms (eds C. Wittmann and J. C. Liao), Wiley-VCH Verlag GmbH & Co. KGaA, Weinheim, Germany. doi: 10.1002/9783527807796.ch3
- **Nair, G.**, Jungreuthmayer, C., Hanscho, M., Zanghellini, J. (2015) Designing minimal microbial strains of desired functionality using a genetic algorithm. Algorithms for Molecular Biology, 10:29.
- Jungreuthmayer, C., **Nair, G.**, Klamt, S., Zanghellini, J. (2013) Comparison and improvement of algorithms for computing minimal cut sets. BMC bioinformatics 14: 318.

Appendix B
Curriculum vitae

Education

2012–2017 **PhD**, *University of Natural Resources and Life Sciences*, Vienna, Austria.

2010–2011 **MSc, Computational Biology**, *University of East Anglia*, Norwich, UK.

2006–2007 **Postgraduate Diploma in Biotechniques**, *Institute of Bioinformatics and Applied Biotechnology*, Bangalore, India.

2002–2006 **Bachelor of Engineering, Biotechnology**, *Visveswaraiah Technological University*, India.

Experience

2012–2015 **Junior Researcher**, *Austrian Centre of Industrial Biotechnology (**ACIB**)*, Vienna, Austria.

2007–2009 **Scientist**, *Abexome Biosciences Pvt. Ltd.*, Bangalore, India.

Skills

Computer

Worked extensively in a Linux environment. Developed and implemented algorithms using Perl, Matlab, CPLEX and GLPK. Also skilled in publication-related tools like LaTeXand Gnuplot. Have some experience in C, Java, SQL and R

Wetlab

Basic molecular biology techniques like basic DNA/RNA handling, recombinant DNA procedures and protein handling procedures.

Languages

English **First language**

German **ÖSD B1 Zertifikat Deutsch** *Conversationally fluent*

Interests

Climbing, Reading

Links

GitHub https://github.com/gogothegreen

Linkedin https://www.linkedin.com/in/govind-nair-07b7aa4b

Researchgate https://www.researchgate.net/profile/Govind_Nair5