17

Identifying Enzyme Knockout Strategies on Multiple Enzyme Associations

Bin Song[1], I. Esra Büyüktahtakın[2], Nirmalya Bandyopadhyay[1], Sanjay Ranka[1] and Tamer Kahveci[1]

[1]*CISE Department, University of Florida, Gainesville*
[2]*Systems and Industrial Engineering, University of Arizona, Tucson*
USA

1. Introduction

Many biochemical engineering applications in drug discovery, food generation and cosmetic production, aim to modify the metabolism of a given organism to increase or decrease the production of a specific compound or a set of compounds. For example:

1. Fatty acid biosynthesis pathway converts fatty acids that are used in the cosmetic industry in creams and lotions.
2. Butanoate metabolism produces poly-β-hydroxybutyrate which is essential for producing plastics.
3. Mevalonic acid pathway and MEP/DOXP pathway produce carotenoid that are often used as anti-oxidant in food industry. The metabolisms of many organisms, such as bacteria, algae and plants naturally produce these compounds. A common practice is to extract them from these organisms.

Enzymes play a significant role in metabolism. They catalyze the chemical reactions that transform a set of substrates (i.e., input compounds) into products (i.e., output compounds). Metabolic engineering techniques often aim to manipulate a small set of genes to alter the speed of the targeted enzymatic reactions. Their eventual goal is to reach a desired level of compound concentrations produced or consumed by these reactions. One way to alter the speed of the reactions dramatically is to knockout a set of enzymes. When an enzyme is knocked out, it cannot catalyze a subset of the reactions, resulting in changes to the productions of compounds.
When detailed *in silico* models are available, computational methods can be successfully used to determine the enzyme set to knockout. These methods, when applicable, have much lower time and cost requirements as compared to *in vitro* or *in vivo* experiments conducted in wet labs. Wet-lab experiments often require substantial effort and time of the domain experts and overall time requirements may be hours to several days. Moreover, the cost of wet-lab experimentation significantly increases when the number of enzymes that needs to be knocked out is more than one. Manipulations that involve four to six enzymes are not uncommon. As a result, biologists often employ computational methods as a preprocessing step to filter out less promising compounds.

A number of heuristic *in silico* solutions exist to find a promising set of enzymes. However, finding the set of enzymes whose knockout leads to achieving the optimal compound production rate is a computationally difficult problem. The number of possible subsets of enzymes that can be considered for manipulation grows exponentially with the number of enzymes in the pathway. Even if the size of each potential subset is limited to at most four, the number of possible subsets for a pathway consisting of 500 enzymes is more than 2.5 billion. Therefore, efficient methods that avoid inspecting the entire search space are necessary.
In order to find a promising set of enzymes to knock out, we first need to provide a computational method to evaluate the metabolic system after some enzymes are knocked out. There are several models to simulate the steady state of a metabolic network. We categorize these methods into three different groups named, boolean models, linear models and non-linear models. Boolean models can be an oversimplification of the metabolic network, especially if the number of reactions and their connectivity increase. Non-linear models require additional information about the network, which may not be available. *Flux Balance Analysis, (FBA)* is a popular linear model which is widely used to compute the flux distribution on the steady state of metabolic networks (Bonarius et al., 1997; Forster et al., 2003; Kauffman et al., 2003). Segre et al. presented a quadratic programming method, named minimization of metabolic adjustment(MOMA) (Segre et al., 2002). Shlomi et al. described a MIP method, called regulatory on/off minimization (ROOM) for predicting the metabolic steady states after the gene or enzyme knockouts (Shlomi et al., 2005).
It is easy to use these models to determine the impact on the metabolism, when a given set of genes are knocked out. However, as discussed earlier, we are interested in finding the subset of enzymes that lead to a desired impact. Optknock (Burgard et al., 2003), OptReg (Pharkya & Maranas, 2006) and OptStrain (Pharkya et al., 2004) are three MIP based methods for identifying the enzymes to be knocked out for the FBA model. All these methods make the simplifying assumption that each reaction can be catalyzed by only one enzyme. This simplification allows a quick conversion of the underlying variables using linear constraints, where MILP or quadratic programming can be used to solve the problem. However, in real metabolic networks, more than one enzyme can be involved in catalyzing a reaction. In particular, more than two enzymes can substitute each other or work collaboratively to catalyze a reaction. Figure 1 illustrates this on a real example we adopted from Reed et al. (Reed et al., 2003). Here we describe these two kinds of enzyme collaborations in brief.

- **Collaborative enzymes:** Some reactions require the presence of two or more proteins or enzymes simultaneously. We call such enzymes as *collaborative enzymes*. In this case, absence of even one of these enzymes is sufficient to slow down or stop the reaction. Logically, there is an Boolean *AND* relation among these enzymes. In Figure 1 (top portion), D-Xylose ABC Transporter is responsible for exporting/importing a variety of molecules to and from bacteria. To carry out this function the genes XylF, XylG and XylH jointly work to catalyze the reaction XYLabc.
- **Substitute enzymes:** Two or more enzymes can substitute each other in catalyzing a reaction. We call such enzymes as *substitute enzymes*. In this case, the presence of one of the substitute enzymes suffices to carry out that reaction. Logically, there is a Boolean *OR* relation among these enzymes. In Figure 1 (bottom portion), Glyceraldehyde 3-Phosphate Dehydrogenase works in a number of metabolic pathways such as the Glycolysis / Gluconeogenesis pathway or biosynthesis of phenylpropanoids. In a number of organisms such as *Arabidopsis thaliana* (*A. thaliana*) this can be done by GapA or GapC with *OR* association.

One can easily generalize the notion of collaborative and substitute enzymes. Thus, a complex topology consisting of multiple enzymes connected by a combination of *OR* and *AND* may catalyze a reaction.

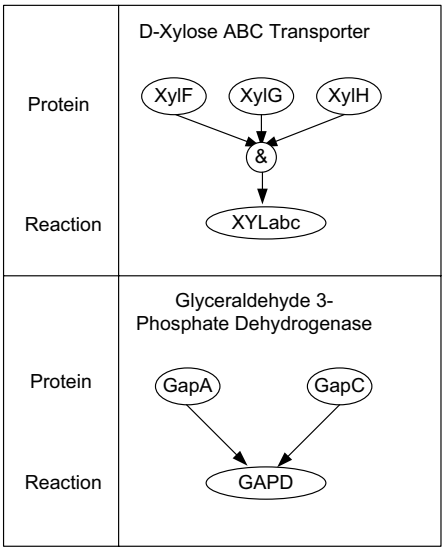

Fig. 1. The figure depicts two examples of reactions catalyzed by multiple enzymes. In the top portion, D-Xylose ABC Transporter is responsible for exporting/importing a variety of molecules to and from bacteria. To carry out this function the genes XylF, XylG and XylH jointly work to catalyze the reaction XYLabc with *AND* association. In the other portion, Glyceraldehyde 3-Phosphate Dehydrogenase works in a number of metabolic pathways such as the Glycolysis / Gluconeogenesis pathway or biosynthesis of phenylpropanoids.

Our goal aims to find the optimal set of enzymes in the presence of multiple enzymes jointly catalyzing the same reaction to knock out so that the production of the system is optimal. In summary, the main contributions of this chapter are as follows:

- We prove that the problem of finding the optimal enzyme set to knockout using MIPL-based approaches is NP-hard even when only one enzyme catalyzes each reaction. This proof is also corroborated by the fact that when the network size increases, the execution time of Optknock framework increases exponentially.
- We develop two solutions to deal with multiple enzyme association along with linear constraints. Our solutions eliminate the limitation that each reaction is catalyzed by a single enzyme. In our model, we allow multiple substitute and collaborative enzymes. Our first solution uses a small number of binary variables in the underlying MILP formulation. The second method increases the number of binary variables but requires a smaller number of constraints. Inclusion of multiple enzymes significantly extends the applicability of our methods, as in real networks, multiple enzymes can catalyze a reaction.

Our experiments using the synthetic and real datasets demonstrate that allowing multiple enzymes to catalyze a reaction increases the computational cost of the solution as compared

to the cases when all reactions are catalyzed by a single enzyme. In our experiments, we observe that our second method that introduces extra binary variables is significantly superior to our first method in terms of execution time. These results also demonstrate that the enzyme topology can have a substantial influence on the performance of the MILP solution.
The rest of the chapter is organized as follows. Section 2 discusses the related work for this chapter. Section 3 proves that finding the optimal set of enzymes to knock out using MILP is NP-hard even when we allow only one enzyme to catalyze each reaction. Section 4 describes the proposed methods when a reaction is catalyzed by multiple enzymes. Section 5 discusses experimental results. We conclude our discussion in Section 6.

2. Related work

In order to identify a promising set of enzymes to knock out, we first require a computational method to evaluate the state of the metabolic system after multiple knockouts. There are several models to simulate the steady state of a metabolic network. These methods can be classified into three categories named Boolean models, linear models and non-linear models.

- **Boolean Models:** Boolean models consider each enzyme as a boolean variable. Each variable can take a either true or false value representing whether the corresponding enzyme is active or inactive. Each reaction is a boolean predicate that depends on these variables. A reaction takes place only if its predicate evaluates to true. Sridhar et al. and Song et al. propose a boolean model of the enzyme knockout strategy (Song et al., 2007; Sridhar et al., 2007; 2008). These methods require the user to supply a list of *targeted compounds* along with a metabolic network. The goal is to identify the set of enzymes whose deletion stop producing all the targeted compounds while causing minimum *damage*. Here, we define damage as the number of non-targeted compounds that are eliminated because of the knockouts. Minimum damage is defined as the minimum number of non-targeted compounds eliminated from the metabolism while eliminating the targeted compounds given all possible ways of eliminating the targeted compounds. Sridhar et al. propounds an optimal algorithm for this model (Sridhar et al., 2008). Song et al. discusses a heuristic algorithm for finding the knockout enzyme strategy (Song et al., 2007). Klamt et al. finds the enzymes for knockout by finding a minimal set of reactions whose deletion leads to an infeasible balanced flux distribution. It employs a minimum cut approach to solve the problem (Klamt & Gilles, 2004).
- **Linear models:** Boolean models can be an oversimplification of the metabolic network, specially when the number of reactions and their connectivity increase. *Flux Balance Analysis, (FBA)* is a popular technique used to analyze the steady state of metabolic networks (Bonarius et al., 1997; Forster et al., 2003; Kauffman et al., 2003). FBA describes a metabolic network as a set of linear equations. FBA finds an optimal steady-state flux distribution that maximizes growth rate under constraints such as mass balance and capacity. FBA achieves a successful description of the metabolic state system by predicting growth rate and by-products of the metabolism (Edwards & Palsson, 2000a;b; Kauffman et al., 2003). However, FBA may not be able to predict an accurate metabolic state after gene or enzyme knockouts. Segre et al. presents a quadratic programming method named minimization of metabolic adjustment (MOMA) for simulation of the resultant state after knockouts (Segre et al., 2002). MOMA attempts to minimize the changes between the flux distribution after a knockout. MOMA uses linear constraints such as mass balance, capacity, and knockout constraints, which are the same set of constraints used by FBA.

Shlomi et al. describes a mixed integer programming method, named regulatory on/off minimization (ROOM), for predicting the metabolic steady states after gene or enzyme knockouts (Shlomi et al., 2005). ROOM finds the flux distribution which minimizes the number of significant flux changes from the wild-type flux distribution. Experiments demonstrate that MOMA and ROOM are superior to FBA in their ability to predict the resultant states after gene or enzyme knockouts. Optknock is an enzyme knockout strategy based on the FBA model (Burgard et al., 2003). It uses a bi-level programming framework for identifying the enzymes to be knocked out. In the inner level, the optimization finds the flux distribution for a given cellular objective such as maximization of biomass yield or minimization of metabolic adjustment (MOMA) (Alper et al., 2005; Segre et al., 2002) etc). In the outer level, the optimization finds the enzymes to be knocked out to optimize a biological objective (e.g., chemical production). OptReg is another bilevel programming method for the enzyme knockout strategy (Pharkya & Maranas, 2006). The difference between Optknock and OptReg is that Optknock framework considers only two states (knockout vs non-knockout) for each reaction which are controlled by enzymes. However, OptReg considers three sets of binary variables for each reaction. These correspond to knockout or non-knockout and down regulation or up regulation. Thus, OptReg provides more candidate manipulation solutions for enzymes. OptStrain, a MILP based method, identifies desired phenotypes by adding or deleting genes or enzymes (Pharkya et al., 2004). All of the above methods use a MILP or a quadratic programming method. Although the objective function of these methods may not be linear, the constraints are linear.

- **Non-linear models:** Although less prevalent, these methods are also used to describe metabolic networks. These methods incorporate further details about the network and thus can simulate the cell system better than the linear model. S-systems (Savageau & Voit, 1987; Voit, 2000) and GMA model (Peschel & Mende, 1986; Voit, 2000) are two examples of non-linear models for metabolic networks. Song et al. proposes methods for these non-linear models (Song et al., 2011). Patil et al. presents an evolutionary programming method which can be applied to non-linear models (Patil et al., 2005). These heuristic solutions use non-linear models with non-linear constraints. They are not guaranteed to produce optimal solutions. Also, the non-linear constraint models require additional information about the network, which may not be available. Therefore, in this chapter we still focus on the linear constraint model.

The methods described in this chapter use a linear model. Our major contribution, as discussed already, is to allow multiple enzymes to catalyze a reaction. This significantly extends the usability of such methods, as in real networks more than one enzymes can catalyze a reaction.

3. Problem formulation

Given a metabolic network and an objective function, one standard way to find the optimal set of enzyme knockouts is to solve the problem as an MILP which is modeled using FBA. In this section, we focus on the MILP formulation of the enzyme knockout problem and prove that the problem is NP-hard even when a single enzyme catalyzes each reaction.

3.1 Formulation

Given a set $N = \{1, ..., \bar{N}\}$ of N metabolites and a set $M = \{1, ..., \bar{M}\}$ of M metabolic reactions, our goal is to determine the maximum yield of the desired products in a metabolic network while minimizing the enzyme knockout costs. We summarize the decision variables as follows:

v_j : the flux of reaction j;
y_j : binary variable which equals to 0 if an enzyme in reaction j is knocked out, and 1 otherwise.

Other relevant parameters used in this problem are:

S_{ij} : stoichiometric matrix coefficient of metabolite i in reaction j;
l_j : minimum possible flow corresponding to flux j;
u_j : maximum possible flow corresponding to flux j;
h_j : cost of blocking the enzyme corresponding to reaction j;
w_j : weight corresponding to the value of flux j.

Here, l_j and u_j are estimated by minimizing and maximizing every reaction flux subject to the constraints from the *enzyme knockout flux balance model (EKFB)* framework given below.
Let I be a set of external metabolites that are imposed on the pathway, and J be the set of metabolites that will not be used within the pathway once they are produced. We denote the flux of the source metabolites in the metabolic pathway by b_i and the flux of the sink metabolites by c_i.
Given these variables and parameters, we represent the integer programming formulation for EKFB as follows:

$$\max \sum_{j \in M} w_j v_j - \sum_{j \in M} h_j (1 - y_j) \tag{1}$$

$$\text{s.t.} \quad \sum_{j \in M} S_{ij} v_j = \begin{cases} -b_i & \text{if } i \in I; \\ c_i & \text{if } i \in J; \\ 0 & \text{if } i \in N \setminus \{J \cup I\}. \end{cases} \tag{2}$$

$$l_j y_j \leq v_j \leq u_j y_j \quad j \in M \tag{3}$$

$$\sum_{j \in M} (1 - y_j) \leq K, \quad \forall j \in M \tag{4}$$

$$y_j \in \{0, 1\} \qquad \forall j \in M. \tag{5}$$

The objective function (1) maximizes weighted flux less fixed charge corresponding to the enzyme knockouts. Constraint (2) provides flux balance equations defined by the stoichiometric matrix. Constraint (3) includes the fixed charge variable y_j. If the enzyme corresponding to reaction j is knocked out, the value of the flux is set to zero and a fixed charge h_j for knocking out the enzyme is imposed. If the fixed charge variable y_j takes value 1, then the lowest flux value is l_j while the highest possible flux value is u_j. Constraint (4) imposes the condition that the maximum number of knockouts is K. Constraints (5) enforce integrality on the fixed charge variables. Similar formulations are provided in Burgard et al. (Burgard et al., 2003), Cover et al. (Covert et al., 2001) and Palsson (Palsson, 2000).

3.2 NP completeness

To prove that finding the enzymes to knockout by EKFB is NP-hard, we show that the uncapacitated fixed charge network flow problem, which is NP-hard, is a special case of the EKFB (Ng & Rardin, 1996). Let $G = (V, A)$ be a directed graph, where V is the set of nodes, A is the set of arcs, $s \in V$ is the single source node, $T \subseteq V$ is a collection of sink vertices and $d_t > 0$ is the demand for node t. Let x_{ij} denote the flow on arc (i, j) with a cost c_{ij}. Let the variable z_{ij} be equal to 1 if arc (i, j) is selected with a fixed cost f_{ij} and 0 otherwise. We then define the *uncapacitated fixed charge network flow problem (UFNF)* as the problem of finding a set of arcs that allow a supply node to send resources to a set of demand nodes, such that the sum of fixed and variable costs are minimized. UFNF can be formulated using the following mixed-integer program:

$$\min \sum_{(i,j)\in A} f_{ij} z_{ij} + \sum_{(i,j)\in A} c_{ij} x_{ij} \tag{6}$$

$$\text{s.t.} \quad \sum_{(i,k)\in A} x_{ik} - \sum_{(k,j)\in A} x_{kj} = \begin{cases} -\sum_{t\in T} d_t & \text{if } k = s; \\ d_k & \text{if } k \in T; \\ 0 & \text{if } k \in V \setminus \{T \cup s\}. \end{cases} \tag{7}$$

$$x_{ij} \leq \lambda z_{ij} \qquad \forall\, (i,j) \in A \tag{8}$$

$$x_{ij} \geq 0 \qquad \forall\, (i,j) \in A \tag{9}$$

$$z_{ij} \in \{0,1\} \qquad \forall\, (i,j) \in A. \tag{10}$$

The objective function (6) minimizes the sum of the fixed costs associated with selecting arc (i, j) and variable costs for sending flow through (i, j). Constraints (7) are classical flow conservation constraints. Constraints (8) ensure that there can not be any flow if z_{ij} is 0. Also, the maximum flow can be at most λ if z_{ij} is 1. Constraints (9) and (10) ensure that x_{ij} is nonnegative and z_{ij} is binary respectively.

Theorem 1. *Finding the enzyme knockout strategy by EKFB is NP-Hard.*

Proof: Let N' be the set of the metabolites and M' be the set of the reactions in a special case metabolic pathway in EKFB. We model it as a network graph $G' = (N', M')$, where each node represents a metabolite $i \in N'$ and each arc represents a reaction $k \in M'$ using metabolite i to produce metabolite j.

For $i \in N'$ and $k \in M'$, we redefine the stoichiometric matrix S_{ik} as S'_{ij} $(i, j \in N')$ such that (i, j) represents reaction k as follows:

$$S'_{ij} = \begin{cases} 1 & \text{if } (i,j) \in M'; \\ -1 & \text{if } (j,i) \in M'; \\ 0 & \text{otherwise.} \end{cases} \tag{11}$$

Note that S'_{ij} has entries 1, -1, and 0, and thus is a special case of S_{ik}. We also define a new variable $\bar{v}_{ij}$ as the flux corresponding to reaction $k \in M'$. Let $I' \subseteq I$ be a set of external metabolites that are imposed to the pathway, and $J' \subseteq J$ be the set of metabolites that will not be used within the pathway after they are produced. Let us define a parameter $\bar{b}_i$ such that $b_i = \bar{b}_i$ and $c_i = \bar{b}_i$ for each $i \in N'$. By using the stoichiometric matrix S'_{ij} and the new variables $\bar{v}_{ij}$, the constraint (2) can be written as,

$$\sum_{(i,l)\in M'} \bar{v}_{il} - \sum_{(l,j)\in M'} \bar{v}_{lj} = \begin{cases} -\bar{b}_l & \text{if } l \in I'; \\ \bar{b}_l & \text{if } l \in J'; \\ 0 & \text{if } l \in N' \setminus \{I' \cup J'\}. \end{cases} \tag{12}$$

We now define a binary variable $\bar{z}_{ij}$ for each variable $\underline{y}_k$, which assumes value 1 if the arc (i,j) is selected and 0 otherwise. We define costs $\bar{c}_{ij}$ and $\bar{f}_{ij}$ such that $\bar{c}_{ij} = -w_k$, $\bar{f}_{ij} = h_k$. Finally, we define a constant $\bar{\lambda}$ as $\bar{\lambda} = u_k$, and set $l_k = 0$ for each reaction $k \in M'$, which is defined by the arc (i,j). Then, the constraint (3) can be written as,

$$0 \leq \bar{v}_{ij} \leq \bar{\lambda}\bar{z}_{ij} \qquad \forall\ (i,j) \in M' \tag{13}$$

with an objective function,

$$\min \sum_{(i,j)\in M'} \bar{c}_{ij}\bar{v}_{ij} + \sum_{(i,j)\in M'} \bar{f}_{ij}\bar{z}_{ij} \tag{14}$$

Thus, a special case of EKFB with an objective function (14) and constraints (11), (12), (13) and $\bar{z}_{ij} \in \{0,1\}$ is a UFNF and hence EKFB is NP-Hard. ■

4. Methods for multiple enzymes

In this section, we develop a more general version of EKFB where we allow multiple enzymes to catalyze a reaction. This extension improves the applicability of our methods as in real networks more than one enzymes can catalyze a reaction. In particular, we focus on the constraints (3) and model the possible interactions between enzymes regarding the reactions they catalyze.

Let E_i be a Boolean variable that denotes whether the ith enzyme is active (i.e., E_i = *true*) or inhibited (i.e., E_i = *false*). As discussed earlier, in EKFB, we assume that a reaction can be catalyzed only by a single enzyme. We use the Boolean variable y_i which is equal to 1 if an enzyme is active, and 0 otherwise.

Let us denote the set of variables for the enzymes that are involved in catalyzing the ith reaction with $\mathcal{E}_i \subseteq \{E_1, E_2, \cdots, E_M\}$. For simplicity, we will use the notation $\mathcal{E}_i = \{E_{ij} |\ E_{ij} \in \{E_1, E_2, \cdots, E_M\}\}$ to denote this set. Let F_i be a function on $\{0,1\}^{|\mathcal{E}_i|}$ representing the relationship between the enzymes for the ith reaction. This function takes $\mathcal{E}_i$ as input and produces an integer. It evaluates to 1 if the ith reaction takes place according to the values of the variables in $\mathcal{E}_i$. It evaluates to 0 otherwise. Also, let the constants l_i and u_i represent the minimum and the maximum flux values. We write the second set of constraints as:

$$l_i F_i \leq v_i \leq u_i F_i. \tag{15}$$

Depending on association between the enzymes that catalyze a reaction, we formulate F_i for three different scenarios.

- A topology consisting only of substitute enzymes that catalyze any reaction. Each reaction may be catalyzed by a single enzyme or a set of enzyme based on the *OR* association i.e., only one of the enzymes need be present to catalyze the reaction (Section 4.1).

- A topology consisting only of collaborative enzymes that catalyze any reaction. Each reaction may be catalyzed by a single enzyme or a set of enzymes based on the *AND* association i.e., all of the enzymes need to be present to catalyze the reaction (Section 4.2).
- A complex topology consisting of multiple enzymes related by a combination of *OR* and *AND* may catalyze a reaction (Section 4.3).

Shlomi et al. presents a way of replacing Boolean expressions that contains two Boolean variables with linear inequalities (Shlomi et al., 2007). However, as the number of Boolean variables grows, the number of additional variables required by this method grows rapidly making the problem nontrivial. In the following sections we discuss two alternative strategies to deal with each of these three scenarios. We name these strategies the *Binary Method* and *Continuous Method*. The former one introduces additional Boolean variables. The second one avoids the addition of Boolean variables, but comes at the expense of additional constraints. We discuss these in detail in the following sections.

4.1 MILP solution in the presence of substitute enzymes

In this section, we consider the case when all the enzymes that catalyze the same reaction can substitute each other. In this case, the presence of at least one of the substitute enzymes is sufficient to carry out the corresponding reaction. Let $\mathcal{E}_i = \{E_{ij} | \, E_{ij} \in \{E_1, E_2, \cdots, E_M\}\}$ denote a set of variables representing the substitute enzymes for reaction i (i.e., flux v_i). Then we write the function F_i that governs the relationship between the variables in $\mathcal{E}_i$ as:

$$F_i = \max_{E_{ij} \in \mathcal{E}_i} \{E_{ij}\}$$

Thus the constraint (15) becomes:

$$l_i \max_{E_{ij} \in \mathcal{E}_i} \{E_{ij}\} \leq v_i \leq u_i \max_{E_{ij} \in \mathcal{E}_i} \{E_{ij}\}. \tag{16}$$

We address the problem of nonlinearity in constraint (16) by performing a variable transformation, which leads to a set of linear constraints. We solve them using traditional MILP solution techniques such as simplex method.
Our linearization technique considers lower and upper bounds separately. We linearize lower bounding constraints given by the inequality $l_i \max_{E_{ij} \in \mathcal{E}_i} \{E_{ij}\} \leq v_i$ as follows,

$$l_i E_{ij} \leq v_i \qquad \forall \, E_{ij} \in \mathcal{E}_i. \tag{17}$$

Linearization of the upper bounding constraints given by the inequality $v_i \leq u_i \max_{E_{ij} \in \mathcal{E}_i} \{E_{ij}\}$ is more complex compared to that of the lower bound. For the linearization, we consider two approaches, namely binary and continuous methods.

Binary method: In this method, we propose the following linear constraints in order to enforce binary restrictions on F_i (i.e., $F_i \in \{0, 1\}$):

$$F_i \geq \frac{\sum_j E_{ij}}{n} \qquad \forall i \tag{18a}$$

$$F_i \leq \sum_j E_{ij} \qquad \forall i \tag{18b}$$

$$F_i \in \{0, 1\} \qquad \forall i \tag{18c}$$

Continuous method: In this method, we define F_i using a continuous variable that takes value in the real domain. We replace the upper bound constraint $v_i \leq u_i \max_{E_{ij} \in \mathcal{E}_i}\{E_{ij}\}$ with the following linear constraints:

$$F_i \leq \sum_j E_{ij} \qquad \forall i \tag{19a}$$

$$F_i \leq 1 \qquad \forall i \tag{19b}$$

$$F_i \geq E_{ij} \qquad \forall i, j \tag{19c}$$

The constraints (19b)- (19c) enforces F_i to assume a binary value, even though we do not directly impose binary restrictions on it.

4.2 MILP solution in the presence of collaborative enzymes

In this section, we consider the case where multiple enzymes collaborate with each other to catalyze the same reaction. In this case, all the enzymes are necessary for the reaction to initiate. Let $\mathcal{E}_i = \{E_{ij} | \ E_{ij} \in \{E_1, E_2, \cdots, E_M\}\}$ denote a set of variables representing the substitute enzymes for reaction i (i.e., flux v_i). We write the function F_i that governs the relationship between the variables in $\mathcal{E}_i$ as:

$$F_i = \min_{E_{ij} \in \mathcal{E}_i}\{E_{ij}\}$$

Thus, we write constraint (15) as,

$$l_i \min_{E_{ij} \in \mathcal{E}_i}\{E_{ij}\} \leq v_i \leq u_i \min_{E_{ij} \in \mathcal{E}_i}\{E_{ij}\}. \tag{20}$$

As we discussed in Section 4.1, constraint (20) is nonlinear. We linearize this constraint using additional variables. We address lower and upper bounds separately.
First, we focus on the upper bound constraints given by the inequality $v_i \leq u_i \max_{E_{ij} \in \mathcal{E}_i}\{E_{ij}\}$. We linearize this part without introducing new variables as follows:

$$v_i \leq u_i E_{ij} \qquad \forall \ E_{ij} \in \mathcal{E}_i \tag{21}$$

The linearization of the lower bound constraints given by the inequality $l_i \min_{E_{ij} \in \mathcal{E}_i}\{E_{ij}\} \leq v_i$ is more complicated. Analogous to the substitute enzyme case, we develop both binary and continuous methods presented in the following two sections.

Binary method: We have already assumed that $F_i \in \{0, 1\}$. We linearize the nonlinear constraint $l_i \min_{E_{ij} \in \mathcal{E}_i}\{E_{ij}\} \leq v_i$ under this assumption as follows:

$$F_i \leq \frac{\sum_j E_{ij}}{n} \qquad \forall i \tag{22a}$$

$$F_i > \frac{\sum_j E_{ij}}{n} - 1 \qquad \forall i \tag{22b}$$

$$F_i \in \{0, 1\} \qquad \forall i \tag{22c}$$

Continuous method: For the continuous method, we replace the lower bound constraint $l_i \min_{E_{ij} \in \mathcal{E}_i} \{E_{ij}\} \leq v_i$ with the following linear constraints:

$$F_i \leq E_{ij} \qquad \forall i \tag{23a}$$

$$F_i \geq \sum_j E_{ij} - (n-1) \qquad \forall i \tag{23b}$$

$$F_i \geq 0 \qquad \forall i \tag{23c}$$

4.3 MILP solution in the presence of complex association of enzymes

In this subsection, we generalize the methods described in the previous two subsections in order to allow associations with arbitrary forms. We consider the case when the reaction can be catalyzed by a set of enzymes such that some of them can substitute for each other and others need to work collaboratively.

For example, assume that ith reaction can be catalyzed by two alternative enzyme complexes that can substitute each other. Also assume that the first and the second of these complexes are formed from two and three enzymes, respectively. These two or three enzymes in the complexes collaborate with each other. We formulate this relationship as F_i = max { min {E_{i1}, E_{i2}}, min {E_{i3}, E_{i4}, E_{i4}} }.

Using standard rules from Boolean algebra, all Boolean equations can be written into disjunctive or conjunctive normal forms. Thus, we transform the equation for each reaction into the following form:

$$F_i = \max_{\mathcal{E}_i^k} \{ \min_{E_{ij} \in \mathcal{E}_i^k} \{E_{ij}\} \}. \tag{24}$$

In this equation, $\mathcal{E}_i^k$ denotes the kth set of collaborative enzymes required by the ith reaction. Thus, we have $\bigcup_k \mathcal{E}_i^k = \mathcal{E}_i$. We define a new binary variable $Z_i^k \in \{0, 1\}$ corresponding to each $\mathcal{E}_i^k$ and rewrite Equation (24) as,

$$F_i = \max_{\mathcal{E}_i^k} Z_i^k. \tag{25}$$

where,

$$Z_i^k = \min_{E_{ij} \in \mathcal{E}_i^k} \{E_{ij}\}. \tag{26}$$

The methods in Section 4.1 and Section 4.2 are used for constraints (26) and (25) respectively to linearize the constraint (15).

5. Experiments

In this section, we evaluate the performance and the limitations of our methods on real and artificially generated metabolic networks. The synthetic datasets provide us a controlled simulation environment that allows us to determine the impact of different characteristics of the network on the performance of our algorithms. We evaluate the performance of our methods quantitatively in terms of their execution time (in seconds).

5.1 Datasets

In our experiments, we used the following real and synthetic datasets.

- **Synthetic datasets:** We randomly generated ten networks of different sizes (given by the number of compounds and the number of reactions). In order to simulate the real networks accurately, we generated these networks so that the number of reactions that involve a compound is distributed according to the power law distribution (Voit, 2000). In other words, the probability of the number of reactions that each compound involves in decreases exponentially with the number of reactions.

  In order to evaluate the impact of multiple enzymes for catalyzing a reaction, on the performance of the algorithms, we generated two types of datasets:

  Single enzyme dataset: In this dataset, each reaction is catalyzed by only one enzyme. Thus, the number of enzymes is equal to the number of reactions.

  Multiple enzyme dataset: In this dataset, all the reactions are catalyzed by at least one enzyme. The number of enzymes attached to a reaction is based on the power law distribution: *the probability that a reaction is catalyzed by k enzymes decreases exponentially with k*. Roughly, 40% of the reactions are catalyzed by at least two enzymes; 30% of the reactions are catalyzed by at least three enzymes; 23.5% of reactions are catalyzed by at least four enzymes; 18.5% of reactions are catalyzed by at least five enzymes and 5% of reactions are catalyzed by at least nine enzymes. Based on these probabilities, we build ten synthetic networks for each network size. Section 5.2.1 describes the results for the synthetic datasets.

- **Real dataset:** We use the metabolic pathways of *Homo sapiens* (*H. sapiens*) from KEGG (Kanehisa & Goto, 2000). The entire *H. sapiens* metabolism consists of 640 enzymes, 1176 reactions and 1067 compounds. Section 5.2.2 provides the results for these real datasets.

Experiment platform: We implemented our algorithms in C++. We applied ILOG CPLEX 11.2 to find the integer linear programming solutions. We executed our experiments on a system with two Pentium 4 3.2Ghz and 1M cache processors, 6 gigabytes of RAM, and a Linux operating system.

5.2 Results

In this section, we evaluate the performance of our algorithms on the synthetic (Section 5.2.1) and real datasets (Section 5.2.2).

5.2.1 Evaluation on synthetic datasets

Our goal in this section is to evaluate the performance of our algorithm for a variety of network parameters using synthetic datasets. These experiments can be decomposed into two sets as described in the previous subsection, namely, single enzyme dataset and multiple enzyme dataset. For an effective comparison, we use identical topology of reactions and compounds for both multiple and single enzyme set. We consider two cases for the multiple enzyme set: a) All multiple enzymes substitute each other. b) All multiple enzymes collaborate with each other.

Performance analysis on single enzyme set: Section 3 proves that finding the enzyme knockout strategy using MILP is NP-Hard. Consider the case when only one enzyme

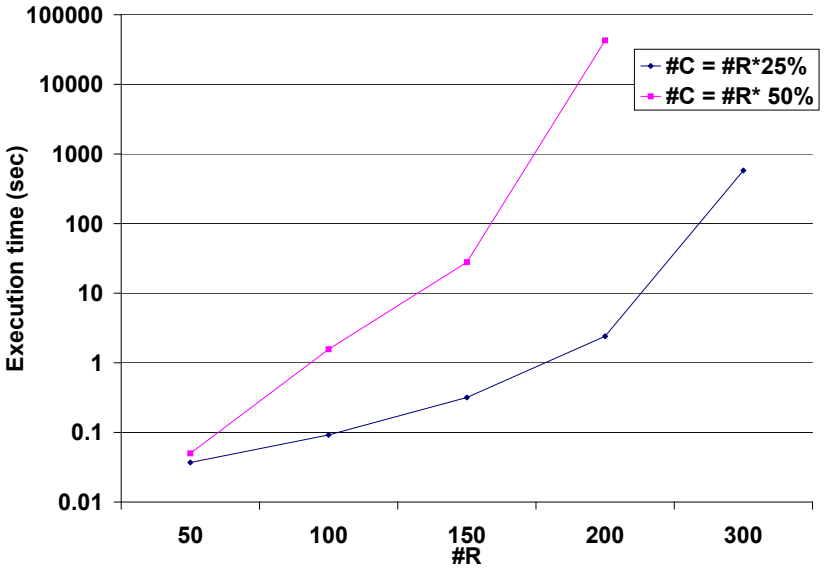

Fig. 2. The average execution time (in seconds) for the networks on single enzyme set. #R denotes the number of reactions and #C denotes the number of compounds in the network. The execution time grows exponentially as the number of reactions increases for both the cases and can be prohibitive even for a few hundred reactions.

catalyzes a reaction. We conduct our experiments using the MILP formulation for two different settings. In the first setting, the number of compounds is 25% of that of the reactions, while for the second setting, it is 50%. Figure 2 plots the average execution times for networks with different number of reactions.

The execution time grows exponentially as the number of reactions increases for both the cases and can be prohibitive even for a few hundred reactions. This time constraint necessitates the advent of heuristic methods for large networks. Also, we observe a steep increase in execution time for larger number of compounds. For the same number of reactions, doubling the number of compounds leads to an overall time increase by several orders of magnitude. It can be concluded that, heuristic methods which can reduce the number of compounds from the constraint set, can have the potential to improve the execution time of the MILP solutions.

Performance analysis for multiple enzymes set: The results in the previous section (along with the NP-hardness of the problem) show that the MILP solution has exponential execution time complexity in terms of the network size. We now study performance of our two solutions with multiple enzymes per reaction. In this experiment, we study the running time requirements in the presence of multiple substitute and collaborative enzymes. We compare these times to those of single enzymes. Note that, the comparison against single enzyme favors the single enzyme dataset as it has fewer variables. This, however, should serve as a lower bound for execution time for the multiple enzyme cases. We summarize the result as follows:

1. Binary method: Figure 3 depicts the results of our binary method for variable number of compounds and reactions. The results demonstrate that the presence of multiple enzymes

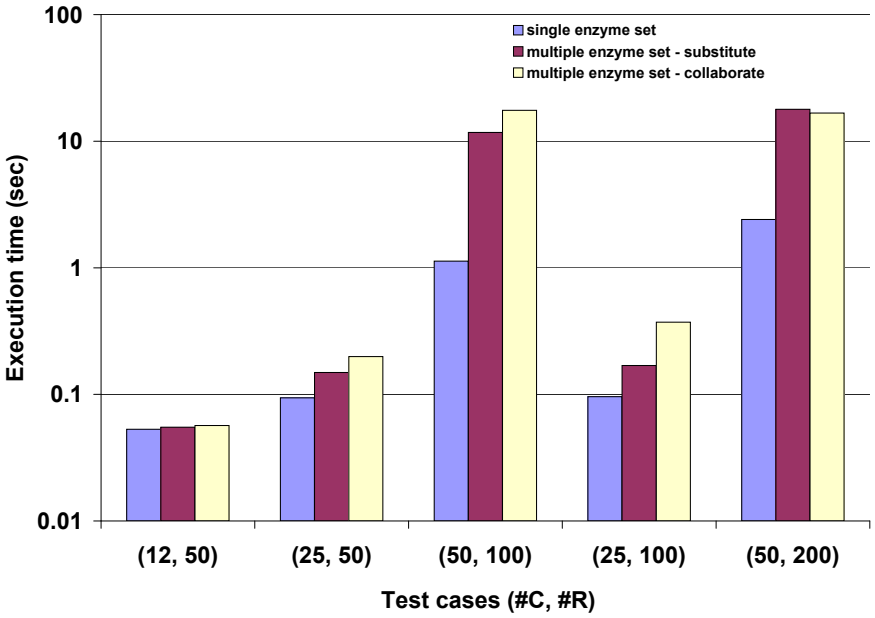

Fig. 3. The average execution time (in seconds) for the networks with single and multiple enzymes. All multiple enzymes cases are either all substitutions or all collaborations. For multiple enzymes set, we use binary method. The results demonstrate that the presence of multiple enzymes increases the execution time significantly as compared to the case when only a single enzyme catalyzes a reaction.

increases the execution time significantly as compared to the case when only single enzyme catalyzes a reaction. This improvement holds true for both substitute and collaborative enzymes. The running time for multiple enzymes is two to 16 times that of the single enzyme case. In most of the test cases, collaborative enzymes resulted in a higher increase in execution time.

2. Continuous method: Figure 4 shows the execution time of multiple enzymes set by continuous method and that for the single enzyme set. Similar to the binary method, multiple enzymes set requires much more time than that of the single enzyme set. As the network size increases, the gap between the execution time of the multiple enzymes set and that of the single enzyme set increases exponentially. This suggests that the presence of multiple enzymes necessitates heuristics solutions for large networks. Also, collaboration among enzymes requires relatively higher execution time as compared to that of the substitution between enzymes in majority of the experiments.
3. Comparison of the two methods: Recall that the binary method introduces additional binary variables to linearize the constraints. The continuous method only generates additional continuous variables. However, it requires additional constraints. Our experiments (see Figures 3 and 4) demonstrate that the Binary method executes twice or more faster than the continuous method for the case when all multiple enzymes cases are substitutions. When the multiple enzymes collaborate with each other, the gap between

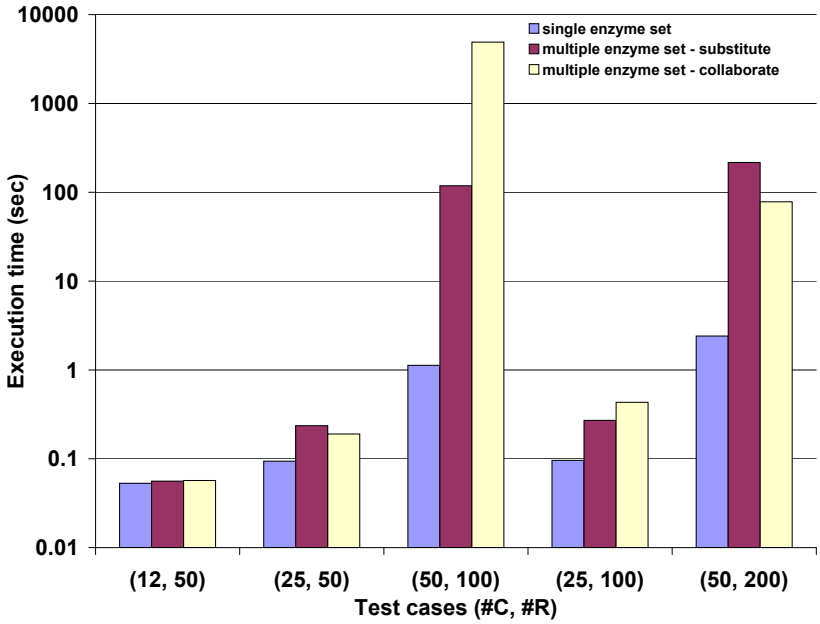

Fig. 4. The average execution time (in seconds) for the networks with single and multiple enzymes. All multiple enzymes cases are either all substitutions or all collaborations. For multiple enzymes set, we use continuous method. As the network size increases, the gap between the execution times of the multiple enzymes set and the single enzyme set increases exponentially.

the running time of the binary and continuous method increases further. Therefore, for the large networks, binary method is the preferred choice.

5.2.2 Evaluation on the real dataset

In this section, we evaluate the performance of our algorithm on real metabolic networks taken from the KEGG database. We use the metabolisms of *H. sapiens*. Given, the superior performance of the binary method over continuous method (as described in the previous subsection), we limit ourselves to the binary method on the real dataset. We execute the binary method for purine metabolism, metabolism of cofactors and vitamins, amino acid metabolism and the entire metabolism. However, the KEGG database does not provide the details of enzyme association information. Thus, we consider two alternative cases: a) all the enzymes are collaborations, b) all the enzymes are substitutions. Table 1 demonstrates the running time using the binary method. These results show that our method requires less than one second of execution time and hence, are scalable to practical network sizes for both cases. Even for the entire metabolism of *H. sapiens*, the execution time is less than half a second. This makes our methods of great practical importance.

It is worth mentioning that the execution times on the real datasets are substantially lower than that of the synthetic datasets. This is because, the topology of the real networks is much

sparser than the ones we used for our synthetic experiments. Therefore, less time is required to find the flux distribution on the real networks.

Pathway	#E	#R	#C	Collaborative	Substitute
Purine metabolism	52	92	65	0.07	0.13
Metabolism of cofactors and vitamins	90	132	122	0.04	0.05
Amino acid metabolism	195	317	305	0.05	0.06
the entire metabolism	640	1176	1067	0.38	0.28

Table 1. Execution time in seconds of our binary method for the metabolisms of *H. sapiens* from KEGG. #E, #R and #C denote the number of enzymes, reactions and compounds respectively in the metabolism. The results demonstrate that our method requires less than one second of execution time. Hence it is scalable to practical network sizes for both the cases.

6. Conclusions

Given a metabolic network and a goal, such as maximizing or minimizing the production of a set of compounds, we considered the problem of computationally determining the optimal enzyme knockouts to modify the production of compounds using the Flux Balance Analysis (FBA) model. We proved that the problem of finding the optimal enzyme set to knockout is NP-hard even when only one enzyme catalyzes a reaction.

We developed two strategies to identify the enzymes to knockout, when multiple enzymes catalyze a single reaction. We allowed multiple substitute and collaborative enzymes. In the proposed solutions, we eliminate this limitation of single enzyme. Our first solution uses a small number of binary variables in the underlying MILP formulation. The second method increases the number of binary variables but requires a smaller number of constraints.

Our experiments using synthetic and real datasets demonstrated that adding extra binary variables is significantly superior to adding additional constraints in terms of execution time. For the metabolism consisting of all the pathways of H. sapiens, our binary method requires less than one second. This makes our methods of great practical importance.

We believe that the approach presented in this chapter is not limited to MILP based strategies. It should also be applicable to other linear constraint strategies, e.g. quadratic programming, where the objective function is non-linear but the constraints are linear.

7. Acknowledgment

This work was supported partially by NSF under grants CCF-0829867 and IIS-0845439.

8. References

Alper, H., Jin, Y., Moxley, J. & Stephanopoulos, G. (2005). Identifying gene targets for the metabolic engineering of lycopene biosynthesis in E. coli, *Metab. Eng.* 7(3).

Bonarius, H. P. J., Schmid, G. & Tramper, J. (1997). Flux analysis of underdetermined metabolic networks: The quest for the missing constraints, *Trends Biotechnology* 15.

Burgard, A. P., Pharkya, P. & Maranas, C. D. (2003). Optknock: A bilevel programming framework for identifying gene knockout strategies for microbial strain optimization, *Biotechnology and Bioengineering* 84.

Covert, M. W., Schilling, C. H. & Palsson, B. (2001). Regulation of Gene Expression in Flux Balance Models of Metabolism, *Journal of Theoretical Biology* 213(1).

Edwards, J. S. & Palsson, B. O. (2000a). Metabolic flux balance analysis and the in silico analysis of Escherichia coli K-12 gene deletions, *BMC Bioinformatics* 1(1).

Edwards, J. S. & Palsson, B. O. (2000b). The Escherichia coli MG1655 in silico metabolic genotype: Its definition, characteristics, and capabilities, *Proc Natl Acad Sci U S A* 97.

Forster, J., Famili, I., Fu, P., Palsson, B. O. & Nielsen, J. (2003). Genome-scale reconstruction of the saccharomyces cerevisiae metabolic network, *Genome Research* 13.

Kanehisa, M. & Goto, S. (2000). KEGG: Kyoto encyclopedia of genes and genomes, *Nucleic Acids Res.* 28(1): 27–30.

Kauffman, K. J., Prakash, P. & Edwards, J. S. (2003). Advances in flux balance analysis, *Current opinion in biotechnology* 14(5).

Klamt, S. & Gilles, E. D. (2004). Minimal cut sets in biochemical reaction networks, *Bioinformatics* 20(2).

Ng, P. H. & Rardin, R. L. (1996). Commodity family extended formulations of uncapacitated fixed charge network flow problems, *Networks* 30(1).

Palsson, B. O. (2000). The challenges of in silico biology, *Nature Biotechnology* 18.

Patil, K. R., Rocha, I., Forster, J. & Nielsen, J. (2005). Evolutionary programming as a platform for in silico metabolic engineering, *BMC Bioinformatics* 6(308).

Peschel, M. & Mende, W. (1986). *The predator-prey model: do we live in a volterra world?*, Akademie-Verlag, Berlin.

Pharkya, P., Burgard, A. P. & Maranas, C. D. (2004). OptStrain: A computational framework for redesign of microbial production systems, *Genome Res.* 14.

Pharkya, P. & Maranas, C. D. (2006). An optimization framework for identifying reaction activation/inhibition or elimination candidates for overproduction in microbial systems, *Metab. Eng.* 8(1).

Reed, J. L., Vo, T. D., Schilling, C. H. & Palsson, B. O. (2003). An expanded genomescale model of escherichia coli k-12 (ijr904 gsm/gpr), *Genome Biology* 4(R54).

Savageau, M. & Voit, E. (1987). Recasting nonlinear differential equations as S-systems: a canonical nonlinear form, *Math. Biosci* 87.

Segre, D., Vitkup, D. & Church, G. (2002). Analysis of optimality in natural and perturbed metabolic networks, *Proc. Natl. Acad. Sci. USA* 99(23).

Shlomi, T., Berkman, O. & Ruppin, E. (2005). Regulatory on/off minimization of metabolic flux changes after genetic perturbations, *Proc. Natl. Acad. Sci. USA* 102.

Shlomi, T., Eisenberg, Y., Sharan, R. & Ruppin, E. (2007). A genome-scale computational study of the interplay between transcriptional regulation and metabolism, *Mol Syst Biol* 3.

Song, B., Buyuktahtakin, I. E., Kahveci, T. & Ranka, S. (2011). Manipulating the steady state of metabolic pathways, , *IEEE/ACM Transactions on Computational Biology and Bioinformatics (IEEE TCBB)*, 8(3).

Song, B., Sridhar, P., Kahveci, T. & Ranka, S. (2007). Double iterative optimization for metabolic network-based drug target identification, *International Journal of Data Mining and Bioinformatics*, 3(2).

Sridhar, P., Kahveci, T. & Ranka, S. (2007). An iterative algorithm for metabolic network-based drug target identification, *Pacific Symposium on Biocomputing* .

Sridhar, P., Song, B., Kahveci, T. & Ranka, S. (2008). OPMET: A metabolic network-based algorithm for optimal drug target identification, *Pacific Symposium on Biocomputing* .

Voit, E. O. (2000). *Computational analysis of biochemical systems: a practical guide for biochemists and molecular biologists*, Cambridge University Press.