A Gene Ranking Framework Enhances the Design Efficiency of Genome-Scale Constraint-Based Metabolic Networks

Yier Ma[1,2], Takeyuki Tamura[1,2]

[1]Bioinformatics Center, Institute for Chemical Research, Kyoto University, Kyoto, Japan
[2]Graduate School of Informatics, Kyoto University, Kyoto, Japan
Email: `mayier@kuicr.kyoto-u.ac.jp`, `tamura@kuicr.kyoto-u.ac.jp`

Abstract

The design of genome-scale constraint-based metabolic networks has steadily advanced, with an increasing number of successful cases achieving growth-coupled production, in which the biosynthesis of key metabolites is linked to cell growth. However, a major cause of design failures is the inability to find solutions within realistic time limits. Therefore, it is essential to develop methods that achieve a high success rate within the specified computation time. In this study, we propose a framework for ranking the importance of individual genes to accelerate the solution of the original mixed-integer linear programming (MILP) problems in the design of constraint-based models. In the proposed method, after pre-assigning values to highly important genes, the MILPs are solved in parallel as a series of mutually exclusive subproblems. It is found that our framework was able to recover most of the successful cases identified by the original approach and achieved a 37% to 186% increase in success rate compared to the original method within the same time limits. Analysis of the MILP solution process revealed that the proposed method reduced the sizes of subproblems and decreased the number of nodes in the branch-and-bound tree. This framework for ranking gene importance can be directly applicable to a range of MILP-based algorithms for the design of constraint-based metabolic networks.

1 Introduction

Mathematical modeling of metabolism is crucial for quantifying key features of metabolic systems and has significantly advanced metabolic engineering. It provides a quantitative framework to describe cell physiology and estimate metabolic pathway usage, allowing a clearer understanding of metabolic behavior. Unlike heuristic methods, it can explain

complex regulatory mechanisms and make predictions beyond tested conditions while ensuring robust and reproducible results[1]. Metabolic modeling is mainly divided into two approaches: kinetic modeling and constraint-based modeling[1, 2, 3]. Kinetic modeling captures dynamic changes in metabolite concentrations by incorporating parameters such as enzyme expression into nonlinear ordinary differential equations (ODEs)[3]. While this approach provides a detailed characterization of metabolic dynamics, it is typically restricted to small-scale systems because of computational complexity and limited experimental data[2, 4, 5]. To address this, the non-equilibrium steady-state assumption is often employed[6, 7]. Both experiments and mathematical analyses support its validity under constant growth conditions in batch cultures[8, 9, 10].

Constraint-based modeling applies the steady-state assumption to represent metabolic reactions linearly, greatly reducing complexity[11, 12]. The steady-state assumption maintains the concentration of a compound constant based on the stoichiometry. Stoichiometry defines the coefficient relationships between reactants and products in chemical equations, revealing the number of molecules each compound contributes to a reaction. When presented in matrix form, these coefficients constitute the stoichiometric matrix, which encodes the structure of metabolic networks[2]. This simplification allows scaling up to the genome level and the integration of gene-protein-reaction (GPR) associations[2]. In the past few decades, constraint-based models have also been developed for various viral species and compiled in public databases, expanding their potential use in designing metabolic networks[13, 14, 15].

Flux balance analysis (FBA) provides a central framework for designing constraint-based metabolic networks[16]. Cell growth is sustained through biomass synthesis, as revealed by analysis of carbon flux distribution in metabolic networks. This identifies biomass synthesis as the principal cellular target for maintaining balance (homeostasis)[1]. In contrast, the production of key metabolites is usually a secondary metabolic activity[1]. This insight forms the basis of FBA, which estimates metabolic flux distributions using constraint-based models by assuming that cells follow specific biological objectives such as cell growth[16]. However, this method has limitations, as the key idea in designing constraint-based metabolic networks is to couple cell growth with the production of desired metabolites[17]. As a result, many algorithms have been developed in recent decades to design constraint-based metabolic networks for growth-coupled production. The details are described in the Discussion section.

GPR associations describe the relationships among genes, proteins, and reactions within metabolic networks. Specifically, they represent the mechanism by which genes encode enzymes (proteins) responsible for catalyzing biochemical reactions. These associations are typically described using logical operators such as "AND" and "OR," where "AND" denotes that multiple genes collectively form an enzyme complex, whereas "OR" indicates that alternative genes can independently catalyze the same reaction[2]. GPR associations integrate genetic information with metabolic reaction networks, enabling the design of metabolic networks at the gene level.

Mixed-integer linear programming (MILP) plays an important role in the design of constraint-based metabolic networks. Constraint-based models represent systems through linear constraints that capture biological limitations[16, 18]. MILP enables

identifying optimal solutions while ensuring all constraints are satisfied. Such problems often involve both continuous variables (e.g., reaction fluxes) and discrete variables (e.g., gene activation or repression)[19]. MILP can handle both variable types simultaneously, allowing the modeling of complex decision structures. Many design tasks also involve logical relationships. These logical (Boolean) conditions can be formulated as integer constraints, making MILP a natural choice for modeling hybrid systems that integrate logic and continuous dynamics[20]. Moreover, design tasks frequently require balancing multiple objectives[21, 22]. MILP supports this need through multi-objective formulations, enabling systematic exploration of trade-offs within the constraint space.

Despite ongoing progress in the design of genome-scale constraint-based metabolic networks, the overall success rate of these computational methods remains limited, mainly due to the need to complete calculations within practical time constraints. Although previous work introduced a gene deletion database for growth-coupled production in constraint-based metabolic networks[23], our goal is to develop pre-screening strategies to identify important genes across diverse networks and reduce computational complexity.

In this study, we propose a framework that ranks gene importance using multiple strategies in genome-scale constraint-based metabolic networks. Ranking gene importance is critical as it enables researchers to prioritize genes that are most likely to be relevant to a specific condition. We ranked gene importance according to GPR associations and network topological information. This simplifies the large-scale datasets and effectively reduces analytical complexity. Through analysis of the solving process of the constructed MILP problems, we verified that pre-assigning values to key genes can further decrease the computational complexity of the MILPs. Moreover, these ranked important genes can be directly integrated with constraint-based modeling algorithms to enhance overall computational efficiency.

In the computational experiments, we evaluated these ranking strategies by embedding the selected genes into the existing algorithm RatGene[24]. We assessed the performance of Gene-Ranked RatGene using three different gene set sizes within a fixed time limit. The results showed that the proposed strategies substantially enhanced success rates across three datasets, including two large-scale datasets. By effectively reducing the complexity of the MILP problems, the framework simplified the computation cost of the solving process.

2　Preliminaries

2.1　Constraint-based Metabolic Network

A constraint-based metabolic network is defined as $N = \{R, M, S, G, F, LB, UB\}$ consisting of three fundamental components and their interactions, including **reactions** R, **metabolites** M, and **genes** G. In this context, S and F represent a **stoichiometric matrix** and a set of **GPR** associations, respectively. LB and UB correspond to sets of **lower bounds** and **upper bounds** for rates of reactions R. The set of reactions R is categorized into reversible reactions and irreversible reactions. The stoichiometric information of the network is encoded within S and the size of S is $m \times n$, where m

and n are the number of metabolites and reactions, respectively. Each entity $s_{j,i}$ of the matrix represents the coefficient of metabolite m_j in reaction r_i. A positive $s_{j,i}$ indicates m_j is produced by r_i, and a negative $s_{j,i}$ corresponds to consumption of m_j, whereas a zero value denotes m_j is not involved in r_i. The biomass reaction $r_{biomass}$ is the reaction that produces all necessary metabolites synthesized for cell growth. A target production reaction r_{target} is a reaction that produces the target metabolite.

An example network is shown in Fig. 1. $\{m_1, m_2, \ldots, m_9\}$ are nine metabolites and $\{r_1, r_2, \ldots, r_{12}\}$ are twelve reactions. r_1, r_9, r_{11}, and r_{12} are reversible reactions that can proceed in both directions, and the rest are irreversible reactions. The intervals on reactions are the lower and upper bounds for reaction rates. A negative lower bound indicates a reversible reaction. Here, r_1 corresponds to the substrate uptake reaction, m_8 to the target metabolite, and m_9 to the metabolite for cell growth. Thus, the reaction r_{11} transfers the target metabolite m_8 is the target production reaction, and r_{12} transfers m_9 is the biomass reaction. The coefficients for each metabolite in all reactions are 1 except for m_7 in r_7. The coefficient of m_7 in r_7 is 3. Based on the above definitions, the stoichiometric matrix S for the example network in Fig. 1 is constructed as:

	r_1	r_2	r_3	r_4	r_5	r_6	r_7	r_8	r_9	r_{10}	r_{11}	r_{12}
m_1	1	−1	−1	0	0	0	0	0	0	0	0	0
m_2	0	1	0	−1	−1	0	0	0	0	0	0	0
m_3	0	0	1	0	0	−1	0	0	0	0	0	0
m_4	0	0	0	1	0	0	−1	0	0	0	0	0
m_5	0	0	0	0	1	0	−1	−1	0	0	0	0
m_6	0	0	0	0	0	1	0	0	−1	0	0	0
m_7	0	0	0	0	0	0	3	0	0	−1	0	0
m_8	0	0	0	0	0	0	0	1	1	0	−1	0
m_9	0	0	0	0	0	0	0	1	0	1	0	−1

Each row represents the relationship between a single metabolite and all reactions. For example, the second row indicates that metabolite m_2 is generated by reaction r_2 but consumed by reactions r_4 and r_5. Each column represents the relationship between a reaction and all the network's metabolites. For instance, reaction r_3 is shown to produce metabolites m_3 from the consumption of metabolite m_1 in the third column.

Furthermore, GPR associations can be categorized into three major types[2]:

$$f_i = \begin{cases} g \\ \bigwedge_{h=1}^{\lambda_i} C_h \\ \bigvee_{h=1}^{\lambda_i} C_h \end{cases} \tag{1}$$

$$C_h = \{g, f\}, \quad g \in G \quad f \in F$$

F represents typically three types: a single gene, the AND connection, and the OR connection. C is a clause in f. C can be either a single gene or an expression following

the same form as an f. Let N be the metabolic network in Fig. 1. Assume it includes five genes and three GPR associations:

$$f_7 = g_1 \wedge g_2 \wedge g_3$$
$$f_9 = (g_2 \wedge g_3) \vee (g_4 \wedge g_5)$$
$$f_{10} = g_1 \vee (g_1 \wedge g_3) \vee g_4 \vee g_5$$

f_7, f_9, and f_{10} correspond to r_7, r_9, and r_{10}, respectively. Their lower bounds and upper bounds are: $lb_7 = 0$, $lb_9 = -10$, $lb_{10} = 0$, $ub_7 = 10$, $ub_9 = 10$, and $ub_{10} = 10$.
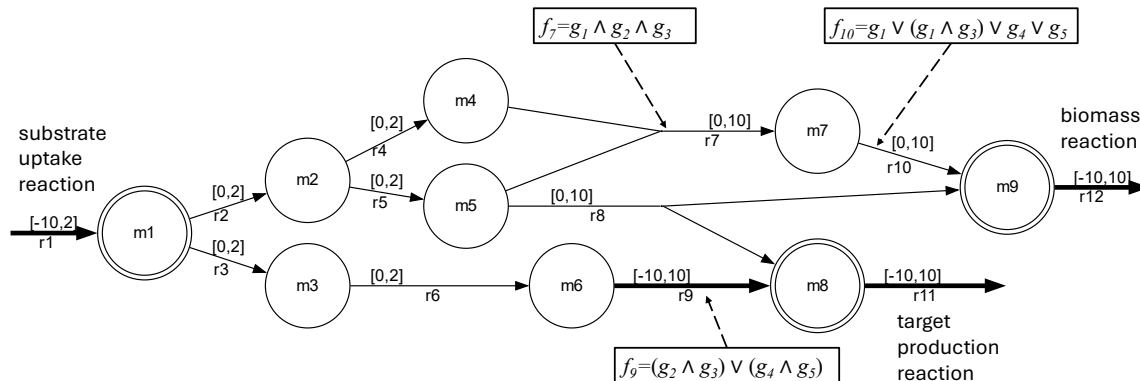

Figure 1. An example of a constraint-based metabolic network. r_1 to r_{12} are reactions, m_1 to m_9 are metabolites and g_1 to g_5 are genes. The intervals marked next to reactions are the lower and upper bounds for reactions. A negative lower bound indicates a reversible reaction. Three valid GPR associations are embedded in this network. r_1 corresponds to the substrate uptake reaction, r_{12} to the biomass reaction. m_8 is defined as the key metabolite, and r_{11} is the target production reaction for its production.

2.2 Growth-coupled Production

The steady-state assumption refers to a condition in which the concentration of a metabolite is dynamically balanced over time. For example, a flux distribution $[2, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0]^T$ confirms that the change in concentration of m_1 is zero in the example network N because the inner product of this vector and the first row of matrix S results 0. The steady-state assumption of the whole metabolic network ensures that concentrations of all metabolites are stable. $S \cdot x = 0$ formulates this steady-state constraint. x is a vector of variables that correspond to the rates of all reactions.

Growth-coupled production has two most common types by definition: strong-coupled production and weak-coupled production. The strong-coupled production is that the production of the target metabolite is strictly possible in all non-zero fluxes with substrate uptake. The weak-coupled production means that the production of the target metabolite is possible in all fluxes with the maximal biomass reaction rate[25]. In this context, we focus on weak-coupled production.

Let x_i be the continuous variable representing the rates of reaction r_i. In particular, $x_{biomass}$ and x_{target} are the two variables that represent rates of biomass reaction and target production reaction, respectively. Let y_k be the binary variable indicating the existence of the gene g_k. A knockout strategy $K \in G$ is a set of genes. K satisfies { $y_\gamma = 0 \,|\, \gamma \in K$ } and { $y_\gamma = 1 \,|\, \gamma \notin K$ }. Form the following linear programming (LP) problem P_1 based on K:

$$max \quad x_{biomass} \qquad\qquad (2)$$
$$s.t. \quad S \cdot x = 0$$
$$p_i \cdot lb_i \leq x_i \leq p_i \cdot ub_i$$
$$p_i = f_i(G)$$

Let $v_{biomass}$ denote the solution of problem P_1, which is the optimal value of the biomass reaction rate as well. Then construct the following LP problem P_2:

$$min \quad x_{target} \qquad\qquad (3)$$
$$s.t. \quad S \cdot x = 0$$
$$p_i \cdot lb_i \leq x_i \leq p_i \cdot ub_i, \quad i \neq biomass$$
$$x_{biomass} = v_{biomass}$$
$$p_i = f_i(G)$$

Let v_{target} denote the solution of problem P_2. If both $v_{biomass}$ and v_{target} are greater than their individual preset thresholds, we call that the deletion strategy K achieves growth-coupled production.

As a further illustration, we again consider the example of N in Fig. 1. The thresholds for production rates of target metabolite m_8 and biomass m_9 are preset at 1 and 1, respectively. Table 1 shows the flux distributions obtained from the above problem P_1 and P_2 under different gene deletion strategies. When the genes are not deleted, or when genes g_4 and g_5 are deleted, the flux distribution reveals that the maximum rate of the biomass reaction v_{12} is 3. By constraining the rate of this reaction to its maximum value, the flux distribution shows that the value of the target production rate v_{11} is 0. This is the growth-coupled rate for the target production reaction in this case. Although the biomass reaction rate is greater than the required threshold, the target production reaction rate is unable to achieve the preset threshold. Consequently, the growth-coupled production is not satisfied in such conditions. When the deletion strategies g_1 and $\{g_1, g_4, g_5\}$ are adopted, maximal value for v_{12} is 2. Then the growth-coupled production rate v_{11} for the target metabolite is 2. Both reaction rates satisfy the thresholds. Thus, the growth-coupled production is achieved under the condition of such deletion strategies.

3   Method

3.1   Gene Ranking Strategies

To evaluate the importance of genes based on a priori information, nine distinct strategies for scoring genes on the basis of GPR associations and topological structures are proposed

Table 1. Deletion strategies and Flux distributions

Deletions	Problem	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}	v_{11}	v_{12}
$\varnothing$	$P_1(flux)$	2	2	0	1	1	0	1	0	0	3	0	3
$\varnothing$	$P_2(flux)$	2	2	0	1	1	0	1	0	0	3	0	3
g_4, g_5	$P_1(flux)$	2	2	0	1	1	0	1	0	0	3	0	3
g_4, g_5	$P_2(flux)$	2	2	0	1	1	0	1	0	0	3	0	3
g_1	$P_1(flux)$	2	2	0	0	2	0	0	2	0	0	2	2
g_1	$P_2(flux)$	2	2	0	0	2	0	0	2	0	0	2	2
g_1, g_4, g_5	$P_1(flux)$	2	2	0	0	2	0	0	2	0	0	2	2
g_1, g_4, g_5	$P_2(flux)$	2	2	0	0	2	0	0	2	0	0	2	2

in this study. GPR associations represent fundamental knowledge for describing the associations between genes and metabolic reactions. These relationships are typically formulated as Boolean functions, and they cover the combinatorial logic by which gene products, primarily enzymes encoded, govern reaction activities within metabolic networks. Such formulations not only enable integrating genome information into metabolic networks but also provide a basis for assessing the individual contributions of each gene to the network. Table 2 provides an overview of all nine strategies, which will be discussed in detail from subsection 3.1.1 to 3.1.7.

Table 2. Gene Ranking strategies Summary

ID	Name	Definition	Equality ID
St1	multiplicity	$\frac{m_N(g_k)}{|F|}$	(6)
St2	frequency	$\frac{\sum_{i=1}^n 1_{U(f_i)}(g_k)}{|F|}$	(7)
St3	logic	$\sum_{i=1}^n Score_{logic}^{f_i}(g_k)$	(11)
St4	degree	$\sum_{i=1}^n Score_{logic}^{f_i}(g_k) \cdot deg(r_i)$	(14)
St5	revdegree	$\sum_{i=1}^n Score_{logic}^{f_i}(g_k) \cdot deg(r_i) \cdot rev(r_i)$	(18)
St6	flux	$\sum_{i=1}^n Score_{logic}^{f_i}(g_k) \cdot weight(r_i)$	(16)
St7	revflux	$\sum_{i=1}^n Score_{logic}^{f_i}(g_k) \cdot weight(r_i) \cdot rev(r_i)$	(19)
St8	fluxdegree	$\sum_{i=1}^n Score_{logic}^{f_i}(g_k) \cdot deg(r_i) \cdot weight(r_i)$	(20)
St9	revfluxdegree	$\sum_{i=1}^n Score_{logic}^{f_i}(g_k) \cdot deg(r_i) \cdot weight(r_i) \cdot rev(r_i)$	(21)

3.1.1 Strategy of Multiplicity

We firstly introduce a metric of gene importance derived from the multiplicity of a gene appearing across all GPR associations in a network, and introduce a quantitative way to evaluate this metric. To better illustrate definitions, the example N in Fig.1 is taken here. N is a metabolic network with five genes and three GPR associations: $f_7 = g_1 \wedge g_2 \wedge g_3$, $f_9 = (g_2 \wedge g_3) \vee (g_4 \wedge g_5)$, and $f_{10} = g_1 \vee (g_1 \wedge g_3) \vee g_4 \vee g_5$. Define a multi-set $U(f_i) = \{g_t \mid g_t \in f_i\}$ for a GPR association as a multi-set including a finite number of

genes which appear in f_i. Then, we have:

$$U(f_7) = \{g_1, g_2, g_3\}$$
$$U(f_9) = \{g_2, g_3, g_4, g_5\}$$
$$U(f_{10}) = \{g_1, g_1, g_3, g_4, g_5\}$$

Define $SQ(f_i) = \{f_i^1, f_i^2, ..., f_i^\epsilon\}$ as the multi-set of sequence for a GPR association f_i. f_i^ϵ is the identity number of the ϵ-th gene that appears in f_i. Then, we have:

$$SQ(f_7) = \{1, 2, 3\}$$
$$SQ(f_9) = \{2, 3, 4, 5\}$$
$$SQ(f_{10}) = \{1, 1, 3, 4, 5\}$$

Define the frequency of occurrence of a gene g_k in a multi-set $U(f_i)$ for a GPR association as:

$$m_{U(f_i)}(g_k) := \#\{t \in SQ(f_i) \mid g_t = g_k\} \tag{4}$$

g_1 appears once and twice in f_7 and f_{10}, respectively. But it does not occur in f_9. The m_U for the example is:

$$m_{U(f_7)}(g_1) = 1$$
$$m_{U(f_9)}(g_1) = 0$$
$$m_{U(f_{10})}(g_1) = 2$$

Therefore, the total multiplicity of a gene g_k in a metabolic network N is defined as:

$$m_N(g_k) := \sum_{i=1}^{n} m_{U(f_i)}(g_k) \tag{5}$$

The m_N for the example is:

$$m_N(g_1) = m_{U(f_7)}(g_1) + m_{U(f_9)}(g_1) + m_{U(f_{10})}(g_1)$$

And the final definition of the score for gene importance in the first strategy $Score_{multiplicity}$ is provided as:

$$Score_{multiplicity}(g_k) := \frac{m_N(g_k)}{|F|} \tag{6}$$

where $|F|$ denotes the total number of GPR associations in a metabolic network. $|F| = 3$ for the example. And $m_N(g_1) = 3$. Finally, $Score_{multiplicity}(g_1) = 3/3 = 1$ for g_1 in the example.

3.1.2   Strategy of Frequency

Similarly, define the second strategy based on the frequency of a gene. We measure whether any gene exists in a given GPR rule. Firstly, define an indicator function $1_S(x) : S \rightarrow \{0, 1\}$ as:

$$1_{U(f_i)}(g_k) := \begin{cases} 1 & if\ g_k \in U(f_i) \\ 0 & if\ g_k \notin U(f_i) \end{cases}$$

S is a multi-set that might include x. Subsequently, we quantify the number of GPR associations in which a gene g_k appears and determine the second strategy for gene importance $Score_{frequency}$ as:

$$Score_{frequency}(g_k) := \frac{\sum_{i=1}^{n} 1_{U(f_i)}(g_k)}{|F|} \tag{7}$$

As a further illustration, we again consider the example of N. Since g_1 only exists in f_7 and f_{10}, we have:

$$1_{U(f_7)}(g_1) = 1$$
$$1_{U(f_9)}(g_1) = 0$$
$$1_{U(f_{10})}(g_1) = 1$$
$$|F| = 3$$

Then $Score_{frequency}(g_1) = \frac{2}{3}$ for g_1.

3.1.3   Strategy of Boolean Logic

The two strategies discussed so far regard the genes associated with each GPR rule merely as a simple set, thereby neglecting the Boolean logic embedded in the GPR representation in (2). To address this limitation, we propose a third strategy for gene importance that explicitly incorporates the underlying Boolean relationships. Assign a basis value β equally to each GPR rule. For a GPR rule f_i with an AND connection, the evaluation of the entire expression becomes 0 if any individual clause is 0. As a consequence, each clause is regarded as having an equal impact on the overall GPR rule f_i. Similarly, for clauses connected by OR relationships within a GPR rule f_i, $f_i = 0$ only holds in the case when all the clauses are 0. Thus, all clauses combined could affect the entire f_i and share the value β. Denote the number of clauses in a OR connection as λ_i. For f_i controlled by a single gene, such a gene will share the whole β:

$$Score_{logic}^{f_i}(C_h) := \beta \quad if\ f_i = \bigwedge_{h=1}^{\lambda_i} C_h \tag{8}$$

$$Score_{logic}^{f_i}(C_h) := \frac{\beta}{\lambda_i} \quad if\ f_i = \bigvee_{h=1}^{\lambda_i} C_h \tag{9}$$

$$Score_{logic}^{f_i}(g_k) := \beta \quad if\ f_i = g_k \tag{10}$$

However, f_i in a metabolic network exhibits hierarchical nesting of multiple clauses, resulting in more complex logical structures for literals in clauses. It is not obvious to derive gene importance from primary clauses. Therefore, RECURSCORE(f_i, g_k, β) function in (12) is developed to calculate $Score_{logic}^{f_i}(g_k)$ for g_k recursively. $Score_{logic}^{f_i}(g_k)$ is defined as the importance score of a gene g_k in a GPR rule f_i given a basis value β based on the above principles (8) to (10). The third strategy for determining the importance of a gene is to sum all $Score_{logic}^{f_i}(g_k)$ in the entire metabolic network as:

$$Score_{logic}(g_k) := \sum_{i=1}^{n} Score_{logic}^{f_i}(g_k) = \sum_{i=1}^{n} \text{RECURSCORE}(f_i, g_k, \beta) \qquad (11)$$

$$\text{RECURSCORE}(f_i, g_k, \beta) = \begin{cases} 0 & if\ g_k \notin U(f_i), \\ \beta & if\ f_i = g_k, \\ \sum_{h=1}^{\lambda_i} \text{RECURSCORE}(C_h, g_k, \beta) & if\ f_i = \bigwedge_{h=1}^{\lambda_i} C_h \\ \sum_{h=1}^{\lambda_i} \text{RECURSCORE}(C_h, g_k, \frac{\beta}{\lambda_i}) & if\ f_i = \bigvee_{h=1}^{\lambda_i} C_h \end{cases} \qquad (12)$$

where λ_i is the number of clauses in f_i.

Take g_1 in the case of N as an example. Assume a basis value is 1. g_1 contributes equally compared as to other clauses in $f_7 = g_1 \wedge g_2 \wedge g_3$, thus:

$$Score_{logic}^{f_7}(g_1) = 1$$

$f_{10} = g_1 \vee (g_1 \wedge g_3) \vee g_4 \vee g_5$ is a OR connection. The first two clauses that include g_1 contribute equally to f_{10}. Therefore, we have:

$$Score_{logic}^{f_{10}}(C_1) = \frac{1}{4} \quad C_1 = g_1$$

$$Score_{logic}^{f_{10}}(C_2) = \frac{1}{4} \quad C_2 = g_1 \wedge g_3$$

g_1 monopolizes all weights in C_1 and shares the same weight as other literals in C_2 which is $(g_1 \wedge g_3)$. Then we have:

$$Score_{logic}^{C_1}(g_1) = \frac{1}{4}$$

$$Score_{logic}^{C_2}(g_1) = \frac{1}{4}$$

$$Score_{logic}^{f_{10}}(g_1) = Score_{logic}^{C_1}(g_1) + Score_{logic}^{C_2}(g_1) = \frac{1}{2}$$

g_1 does not exist in f_9, thus $Score_{logic}^{f_9}(g_1) = 0$. Finally $Score_{logic}$ of g_1 should be:

$$Score_{logic}(g_1) = Score_{logic}^{f_7}(g_1) + Score_{logic}^{f_9}(g_1) + Score_{logic}^{f_{10}}(g_1) = \frac{3}{2}$$

3.1.4 Strategy of Degree

The fourth strategy accounts for the degree of each reaction node. The higher degree of a reaction node indicates a larger set of potential metabolic pathways involved. Consequently, the perturbation to such a node may compromise the integrity of the network, thereby indicating its importance within the network.

Define in-degree, out-degree, and degree of reaction r_i as:

$$
\begin{aligned}
deg^+(r_i) &:= |\{s_{i,j} \mid s_{i,j} < 0,\ i = 1, 2, ..., q\}| \\
deg^-(r_i) &:= |\{s_{i,j} \mid s_{i,j} > 0,\ i = 1, 2, ..., q\}| \\
deg(r_i) &:= deg^+(r_i) + deg^-(r_i)
\end{aligned}
\tag{13}
$$

where $s_{i,j}$ is the entity of the stoichiometric matrix S. Then the sum of weighted scores of $Score_{degree}$ is defined as the gene importance for g_k:

$$
Score_{degree}(g_k) := \sum_{i=1}^{n} Score_{logic}^{f_i}(g_k) \cdot deg(r_i)
\tag{14}
$$

We now return to the example of N previously discussed. $Score_{logic}^{f_7}(g_1) = 1$, $Score_{logic}^{f_9}(g_1) = 0$, and $Score_{logic}^{f_{10}}(g_1) = \frac{1}{2}$. f_7, f_9, and f_{10} correspond to r_7, r_9, and r_{10} in Fig. 1, respectively. Then we have:

$$
\begin{aligned}
deg(r_7) &= 3 \\
deg(r_9) &= 2 \\
deg(r_{10}) &= 2
\end{aligned}
$$

It is derived that:

$$
\begin{aligned}
Score_{degree}^{f_7}(g_1) &= 3 \\
Score_{degree}^{f_9}(g_1) &= 0 \\
Score_{degree}^{f_{10}}(g_1) &= 1
\end{aligned}
$$

Finally, $Score_{degree}(g_1) = 4$ for g_1.

3.1.5 Strategy of Flux Bounds

In the fifth strategy, the difference between the upper and lower bounds of each reaction rate given by the constraint-based model is regarded as the weight of the reaction:

$$
weight(r_i) := \ln\left(ub_i - lb_i + 1\right) + 1
\tag{15}
$$

As with the previous strategy, we define the fifth strategy for evaluating gene importance as:

$$
Score_{flux}(g_k) := \sum_{i=1}^{n} Score_{logic}^{f_i}(g_k) \cdot weight(r_i)
\tag{16}
$$

Recall the lower bounds and upper bounds for r_7, r_9, and r_{10} in the example network N as shown in Fig. 1. $lb_7 = 0, lb_9 = -10, lb_{10} = 0, ub_7 = 10, ub_9 = 10$, and $ub_{10} = 10$. Then we have:

$$weight(r_7) = 1 + \ln 11$$
$$weight(r_9) = 1 + \ln 21$$
$$weight(r_{10}) = 1 + \ln 11$$

Additionally, $Score_{logic}^{f_7}(g_1) = 1$, $Score_{logic}^{f_9}(g_1) = 0$, and $Score_{logic}^{f_{10}}(g_1) = \frac{1}{2}$. Thus, we have:

$$Score_{flux}^{f_7}(g_1) = Score_{logic}^{f_7}(g_1) \cdot (1 + \ln 11) = (1 + \ln 11)$$
$$Score_{flux}^{f_9}(g_1) = Score_{logic}^{f_9}(g_1) \cdot (1 + \ln 21) = 0$$
$$Score_{flux}^{f_{10}}(g_1) = Score_{logic}^{f_{10}}(g_1) \cdot (1 + \ln 11) = \frac{(1 + \ln 11)}{2}$$

Finally, $Score_{flux}(g_1) = \frac{3}{2} + \frac{3 \ln 11}{2}$ for g_1.

3.1.6 Strategies of Combined Reversibility with Degree and Flux Bounds

As illustrated in the Fig. 1, reactions are classified into two categories according to their reversibility. Reversible reactions, which can proceed in both forward and backward directions, are associated with a larger number of potential metabolic pathways compared with irreversible ones. They are therefore considered to be of greater importance. Based on this distinction, we introduce the following definitions:

$$rev(r_i) := \begin{cases} 1 & if \ r_i \in irreversible \ reactions \\ 2 & if \ r_i \in reversible \ reactions \end{cases} \tag{17}$$

In the case of N, reaction r_7 and r_{10} are irreversible reactions, while r_9 is a reversible reaction. Then it could be derived:

$$rev(r_7) = 1$$
$$rev(r_9) = 2$$
$$rev(r_{10}) = 1$$

By combining the property of reversibility with the two structural characteristics mentioned above, namely degree and flux gap, we can derive the following sixth and seventh evaluation strategies for gene importance:

$$Score_{revdegree}(g_k) := \sum_{i=1}^{n} Score_{logic}^{f_i}(g_k) \cdot deg(r_i) \cdot rev(r_i) \tag{18}$$

$$Score_{revflux}(g_k) := \sum_{i=1}^{n} Score_{logic}^{f_i}(g_k) \cdot weight(r_i) \cdot rev(r_i) \tag{19}$$

As for the same case of g_1 in the example network N, substituting the above relevant numbers yields:

$$Score_{logic}^{f_7}(g_1) \cdot deg(r_7) \cdot rev(r_7) = 3$$

$$Score_{logic}^{f_9}(g_1) \cdot deg(r_9) \cdot rev(r_9) = 0$$

$$Score_{logic}^{f_{10}}(g_1) \cdot deg(r_{10}) \cdot rev(r_{10}) = 1$$

$$Score_{logic}^{f_7}(g_1) \cdot weight(r_7) \cdot rev(r_7) = 1 + \ln 11$$

$$Score_{logic}^{f_9}(g_1) \cdot weight(r_9) \cdot rev(r_9) = 0$$

$$Score_{logic}^{f_{10}}(g_1) \cdot weight(r_{10}) \cdot rev(r_{10}) = \frac{1 + \ln 11}{2}$$

Then $Score_{revdegree}(g_1) = 4$ and $Score_{revflux}(g_1) = \frac{3}{2} + \frac{3\ln 11}{2}$ for g_1, respectively.

3.1.7 Strategies of Fluxdegree and Revfluxdegree

In addition, we integrate the information for degree and flux gap to formulate the eighth strategy, and composite all three local topological information to derive the ninth strategy:

$$Score_{fluxdegree}(g_k) := \sum_{i=1}^{n} Score_{logic}^{f_i}(g_k) \cdot deg(r_i) \cdot weight(r_i) \tag{20}$$

$$Score_{revfluxdegree}(g_k) := \sum_{i=1}^{n} Score_{logic}^{f_i}(g_k) \cdot deg(r_i) \cdot weight(r_i) \cdot rev(r_i) \tag{21}$$

Building on the previous example network N as well, we introduce specific numbers of degree, flux bound, and reversibility to further illustrate this point:

$$Score_{logic}^{f_7}(g_1) \cdot deg(r_7) \cdot weight(r_7) = 3 + 3 \ln 11$$

$$Score_{logic}^{f_9}(g_1) \cdot deg(r_9) \cdot weight(r_9) = 0$$

$$Score_{logic}^{f_{10}}(g_1) \cdot deg(r_{10}) \cdot weight(r_{10}) = 1 + \ln 11$$

$$Score_{logic}^{f_7}(g_1) \cdot deg(r_7) \cdot weight(r_7) \cdot rev(r_7) = 3 + 3 \ln 11$$

$$Score_{logic}^{f_9}(g_1) \cdot deg(r_9) \cdot weight(r_9) \cdot rev(r_9) = 0$$

$$Score_{logic}^{f_{10}}(g_1) \cdot deg(r_{10}) \cdot weight(r_{10}) \cdot rev(r_{10}) = 1 + \ln 11$$

Finally, both $Score_{fluxdegree}(g_1) = 4 + 4 \ln 11$ and $Score_{revfluxdegree}(g_1) = 4 + 4 \ln 11$ hold for g_1.

3.2 Gene-Ranked RatGene

RatGene[24] is an algorithm that models the quantitative relationship between the reaction rates of the two most critical reactions in a growth-coupled production state by

utilizing the growth-to-production ratio and fully integrating the Boolean function of GPR associations. In this study, a priori knowledge enables us to rank gene importance according to nine strategies. Leveraging this ranking, the MILP problems constructed by RatGene can be decomposed into subproblems by fixing the values of binary variables corresponding to a subset of the most important genes. Assuming that a set D of κ genes is selected, we can construct 2^κ subproblems by assigning values of 0 and 1 to each variable separately. Let ξ be a natural number such that $\xi = 1, 2, ..., 2^\kappa$. Define the ξ-th subproblem P_ξ as:

$$
\begin{aligned}
min \quad & -x_{biomass} + TMGR \cdot \|x_Q\|_0 \quad\quad\quad (22)\\
s.t.\, & S \cdot x = 0 \\
& p_i \cdot lb_i \leq x_i \leq p_i \cdot ub_i \\
& p_i = f_i(y) \\
& x_{biomass} \geq lb^{min}_{biomass} \\
& \frac{v_{target}}{v_{biomass}} = \alpha \\
& 0 \leq \alpha \leq \frac{TMPR}{lb^{min}_{biomass}} \\
& Q := \{i \mid \exists f_i\} \\
& y_{d_k} = \lfloor \frac{\xi - 1}{2^{\kappa-1}} \rfloor \; mod \; 2, \; d_k \in D, \; k = 1, 2, ..., \kappa
\end{aligned}
$$

where the objective function is constructed as minimizing the sum of l_0-Norm of the reactions scaled by the theoretical maximum growth rate (TMGR) and the negative biomass reaction rate. The coefficient TMGR is to normalize the two objectives in the objective function to the same level of extent. Upper or lower bounds are imposed on the three reactions to simulate the actual cell growth process. α is a fixed value in each MILP. In RatGene, the appropriate α is obtained by iteratively assigning different values to form a series of MILPs. Then, solve those MILPs. In addition, each binary variable y_{d_k} corresponding to the gene d_k in the selected set D is assigned a value of 1 or 0. From this, the following inference can be readily drawn. In the MILP problem $P_{RatGene}$ formulated by RatGene, it is possible to decompose the problem into 2^κ subproblems. The complexity of each subproblem is less than $P_{RatGene}$. Therefore, we parallelize the computation of these subproblems, which can greatly save time compared to solving the original problem.

4   Computational Experiments

In this study, we present nine strategies for quantifying gene importance by utilizing GPR associations and the topological features of constraint-based metabolic networks. These strategies allow for the prior ranking of genes, enabling the systematic identification of those most involved in key biological processes. The binary variables associated with the selected genes are then assigned values of 0 or 1 and integrated into our modified

Gene-Ranked RatGene framework. We conducted computational experiments on three datasets from the BiGG database[15] to evaluate the effectiveness of the proposed strategies. The datasets iML1515 and iMM904 represent the genome-scale metabolic networks of *E. coli* and *S. cerevisiae*, respectively. The e_coli_core dataset represents a small-scale *E. coli* network. All experiments were performed on an Ubuntu 20.04 system equipped with an AMD Ryzen Threadripper3 3970X CPU (3.70 GHz, 32 cores / 64 threads). The computational environment included IBM ILOG CPLEX 12.10, the COBRA Toolbox v3.0 [26], and MATLAB R2019b.

Method I denotes the proposed Gene-Ranked RatGene framework by this study, and the original RatGene is denoted as **Method II** in the following context. Two main metrics were used for evaluation: the number of successful cases and the average runtime. A successful case is defined as one in which the computational method can successfully identify a valid gene knockout strategy enabling growth-coupled production for a target metabolite.

Table 3 summarizes the performance of Method I using three different sizes of ranked gene sets and Method II used as the benchmark on the iMM904 dataset under a fixed time limit. The results are divided into three categories: (1) only Method I succeeded, (2) both Methods succeeded, and (3) only Method II succeeded. As shown in Table 3, Method I incorporating gene importance rankings consistently outperformed Method II in terms of both the number of successful cases and average runtime. When the most important single gene was selected and assigned 0 and 1, treating each as a mutually exclusive parallel process, Method I achieved 36.84%–52.63% more successful cases than Method II, indicating a notable improvement. It also reproduced over 80% of success cases of Method II while reducing runtime, demonstrating both reliability and efficiency. Similarly, when the two most important genes were selected, Method I yielded up to 113.68% additional successful cases, while still recovering approximately 90% of Method II's results and reducing runtime. This confirms a clear and significant advantage over Method II. When the three most important genes were used, Method I achieved a 118%-186% increase in successful cases compared to Method II. It also matched up to 95% of success cases of Method II, while continuing to reduce computation time, highlighting both robustness and efficiency. Among the successful cases obtained only by Method II, the average runtime is greater than the average runtime of all successful cases by Method II. This suggests the potential complexity of these cases, which may explain why Method I failed to produce results within the time limit. Furthermore, comparison across different gene set sizes shows that selecting more important genes steadily improved performance in both the (1) and (2) categories. Notably, the results of cases in (1) showed a more than threefold increase, demonstrating the framework's effectiveness in identifying additional, previously undetected successful cases. These findings highlight the scalability, efficiency, and effectiveness of the proposed framework. For further details, refer to Table S1 in the supplementary file.

Table 4 presents the performance comparison between Method I and Method II, each using three different sizes of ranked gene sets, on the iML1515 dataset under a specified time limit. Notably, the top three genes were ranked identically by all nine strategies for this dataset. When one important gene was included, Method I outperformed Method

Table 3. Performance comparison on iMM904 dataset

κ [d]	Dataset iMM904 Evaluation Strategies[a,b]	(1)Only Method I Succeeded Succ. Case[c]	Avg. Time	(2)Both Methods Succeeded Succ. Case[c]	Avg. Time	(3)Only Method II Succeeded Succ. Case[c]	Avg. Time
1	St1,St2,St4,St8	50	371.28	82	194.57	13	386.79
	St3,St5,St6,St7,St9	35	402.56	77	175.80	18	342.60
2	St1,St2	108	343.02	87	179.62	8	276.11
	St3,St6	65	427.48	82	191.22	13	316.76
	St4,St8	86	417.86	82	198.57	13	347.70
	St5,St7,St9	66	382.55	84	183.82	11	304.20
3	St1,St2	177	322.64	90	157.29	5	279.17
	St3,St4,St6,St8	112	379.86	86	197.36	9	337.96
	St5,St7,St9	151	335.54	84	191.61	11	339.98
	Origin. RatGene	95	210.92	95	210.92	95	210.92

[a] Maximum loops for RatGene is 200, and the time limit for each metabolite is 500 seconds.
[b] Strategy IDs refer to Table 2
[c] Number of success cases.
[d] Selected κ important genes associating with Gene-Ranked RatGene.

II by producing 36.90% additional successful cases, indicating a notable improvement. It also preserved 88.69% of successful cases of Method II while reducing runtime, demonstrating a clear advantage in both effectiveness and efficiency. With two important genes, Method I achieved a 48.21% increase in successful cases compared to Method II and reliably reproduced nearly 90% of Method II's results, further confirming its robustness and superior performance. Using three important genes, Method I generated almost 50% more successful cases than Method II. It also maintained over 87.50% overlap with Method II's success cases while achieving a comparable average runtime, highlighting both its computational efficiency and reliability. Detailed information can be found in Table S2 provided in the supplementary file.

Table 4. Performance comparison on iML1515 dataset

κ [d]	Dataset iML1515 Evaluation Strategy[a,b]	(1)Only Method I Succeeded Succ. Case[c]	Avg. Time	(2)Both Methods Succeeded Succ. Case[c]	Avg. Time	(3)Only Method II Succeeded Succ. Case[c]	Avg. Time
1	St1 to St9	62	354.71	149	271.97	19	246.71
2	St1 to St9	81	351.31	147	287.27	21	306.37
3	St1 to St9	81	326.22	147	292.92	21	231.15
	Origin. RatGene	168	283.76	168	283.76	168	283.76

[a] Maximum loops for RatGene is 200, and the time limit for each metabolite is 500 seconds.
[b] Strategy IDs refer to Table 2
[c] Number of success cases.
[d] Selected κ important genes associating with Gene-Ranked RatGene.

Table 5 shows the results on the e_coli_core dataset, a small-scale constraint-based metabolic network comprising only 72 metabolites and 95 reactions. On this dataset, Method I yielded few additional success cases. However, they were still able to recover nearly all the success cases identified by Method II. This outcome suggests that, for small-scale models, the primary bottleneck is not computation time, and Method II may already operate at full efficiency. Furthermore, the observed increase in average runtime

indicates that the proposed framework does not improve computational efficiency in small-scale models. For the few successful cases obtained solely through Method II, their average runtime largely exceeded the mean runtime of all Method II's successful cases. Detailed information is available in Table S3 presented in the supplementary file.

Table 5. Performance comparison on e_coli_core dataset

κ [d]	Evaluation Strategy[a,b]	(1)Only Method I Succeeded Succ. Case[c]	Avg. Time	(2)Both Methods Succeeded Succ. Case[c]	Avg. Time	(3)Only Method II Succeeded Succ. Case[c]	Avg. Time
	St1,St2,St3,St6,St7	0	-	45	236.89	1	422.08
1	St4,St8	0	-	44	288.01	2	461.04
	St5,St9	2	365.49	44	329.37	2	295.55
	St1,St2,St3,St6,	0	-	45	250.35	1	422.08
2	St4,St8	0	-	45	366.25	1	422.08
	St5,St7,St9	0	-	45	347.03	1	422.08
	St1,St2,St3,St4,St6,St8	1	187.10	45	364.95	1	422.08
3	St5,St9	0	-	45	371.67	1	422.08
	St7	1	96.05	45	375.94	1	422.08
	Origin. RatGene	46	266.87	46	266.87	46	266.87

[a] Maximum loops for RatGene is 200, and the time limit for each metabolite is 500 seconds.
[b] Strategy IDs refer to Table 2
[c] Number of success cases.
[d] Selected κ important genes associating with Gene-Ranked RatGene.

5   Discussion and Conclusion

As previously discussed, obtaining gene deletion strategies within limited time constraints remains a significant challenge, particularly in genome-scale constraint-based metabolic networks, where the success rates of existing methods are relatively low. In this study, we introduced nine novel strategies for evaluating gene importance by integrating GPR associations with topological properties of metabolic networks. Based on these strategies, we developed a Gene-Ranked framework (Method I) extending the existing RatGene method (Method II). Note that a Gene-Ranked framework refers to the parallel computation of mutually exclusive subproblems that have been decomposed. Regardless of whether Method I or Method II is applied, the solver exploits internal parallel computing automatically as well when solving MILP problems.

To ensure a fair and consistent comparison between Method I and Method II, all analyses were conducted under standardized conditions. Both methods were applied to identical datasets, following the same preprocessing procedures, parameter configurations, and evaluation metrics. This rigorous and consistent evaluation provides a reliable basis for comparing the effectiveness and robustness of the two approaches. In addition, according to the definition of RatGene, the number of iterations corresponds to the number of uniformly sampled points in the ratio constraint space[24]. Increasing the number of samples improves the likelihood of identifying a feasible solution, but also leads to higher computational costs. To achieve a trade-off between solution feasibility and computational efficiency, the number of iterations is set to 200.

5.1 Investigation of the MILP Solving Process

Computational results in Tables 3 and 4 show that Method I consistently achieved at least 80% of the successful cases found by Method II, but with reduced runtime. Furthermore, we identified up to twice as many additional successful cases within the same time limits on genome-scale models for two species. These improvements can be attributed to the assignment of binary values to important gene variables, which effectively altered the structure of the MILP problems. Modern MILP solvers employ several powerful techniques to efficiently solve such problems, including presolving, branch and bound, and cutting planes. Presolving simplifies the problem by eliminating duplicate rows, removing redundant columns, and fixing the values of certain variables prior to solving. Solvers also store and process only non-zero coefficients to optimize memory and computation [27]. The branch-and-bound technique iteratively constructs a search tree, solving LP relaxations at each node, and applies bounding and pruning strategies to reduce the search space [28]. The cutting plane method improves the formulation by progressively adding valid constraints that tighten the feasible region, thereby guiding the solver more efficiently toward the optimal solution [29].

To better understand the impact of our Method I on the optimization process, we further examined how the MILP problems reformulated by Method I differed from those formulated by Method II during the solution process, using the iMM904 dataset as a case study.

Table 6. Solving Process of MILPs for the Target Metabolites where Both Methods Succeeded

Dataset iMM904		Summary of Solving Process (Cases where Both Methods were successful)					
κ [a]	Evaluation Strategy	Avg. Nodes[b]	Avg. Rows[c]	Avg. Columns[c]	Avg. Non-zeros[d]	Avg. Binary[d]	Avg. Cuts[e]
1	St1,St2,St4,St8	1363.91	1141.04	1037.25	4593.85	448.85	19.02
	St3,St5,St6,St7,St9	1477.21	1171.94	1062.24	4750.59	447.71	18.32
2	St1,St2	647.03	1145.42	1040.50	4628.47	450.87	14.32
	St3,St6	1439.94	1158.94	1053.74	4718.00	439.87	12.61
	St4,St8	3412.80	1119.90	1023.30	4478.70	441.42	42.03
	St5,St7,St9	3143.09	1109.16	1019.81	4450.63	432.74	26.37
3	St1,St2	3543.06	1138.86	1037.22	4599.61	448.81	16.61
	St3,St4,St6,St8	1476.85	1117.49	1023.95	4502.34	436.34	17.73
	St5,St7,St9	1646.30	1120.87	1030.34	4451.09	432.34	12.70
	Origin. RatGene	5006.71	1162.71	1046.49	4613.01	447.9	56.36

[a] Selected κ important genes associating with Gene-Ranked RatGene.
[b] The average number of nodes searched in branch-and-bound trees, and the smaller the better.
[c] The average number of constraints and variables after pro-solving processes, and the smaller the better. Rows correspond to constraints, and columns correspond to variables in a constraint matrix.
[d] The average number of non-zero counts and binary variables, and the smaller the better.
[e] The average number of cutting planes, and the smaller the better.

Table 6 presents key statistics from the solution process of MILP problems constructed by Method I, using cases where both methods were successful on the iMM904 dataset. Across all three configurations: assigning values to one, two, or three important genes, Method I consistently traversed fewer branch-and-bound nodes than Method II. This reduction in the number of LP relaxations was a major factor contributing to the

improved computational efficiency. Additionally, over 80% of Method I effectively reduced problem size in terms of the average number of variables, constraints, and non-zero counts. At the configuration with three important genes, all three of these metrics were lower than Method II averages and also outperformed the corresponding results from configurations using one or two genes. This suggests that assigning fixed binary values to a larger number of key gene variables simplifies the problem structure, making it easier to solve. Similarly, the number of cutting planes generated by the solvers was significantly lower for Method I compared to Method II. This further supports the conclusion that our Method I reduced MILP problem complexity and can achieve faster solutions within the time limit. These findings align with the performance results reported for the set of (2) in Table 3, providing insight into the underlying causes of improved performance. However, the relationship between MILP complexity and runtime was not strictly linear. For instance, in the multiplicity (St1) and frequency (St2) strategies using two important genes, the number of nodes explored was markedly less than that of Method II. Nonetheless, due to the internal behavior of MILP solvers, including heuristic strategies, secondary branch-and-bound processes, and other internal procedures, the reduction in nodes did not always lead to a proportional decrease in average runtime, as shown in Table 3. Comprehensive details for the solution process of cases where both methods were successful are provided in Table S4 within the supplementary file.

Method I essentially partitioned the problem constructed by Method II into mutually exclusive subproblems for parallel computation, and then solved each subproblem independently. In principle, the successful results obtained by Method II should be fully reproduced by combining the solutions of all subproblems by Method I. However, due to the computation time limit, the ratios reported in Table 3 did not reach 100%. The following discussion examines the causes of these non-reproducible cases.

Table 7 presents key information about the solving processes of different constructed MILP problems by Method I, focusing on unsuccessful cases from target metabolites in the set of (1) on the iMM904 dataset. The set of (1) refers to cases for which knockout strategies were successfully identified by Method II but not by Method I. In terms of average node search, Method I required larger searches compared to Method II, suggesting that the problems became more difficult to solve when certain gene variables were fixed. At the same time, Method I consistently produced smaller MILP instances than Method II across four metrics: average number of rows, columns, non-zero counts, and binary variables. For cutting planes, the problems generated by Method I with three fixed important gene values were also less complex than those produced by Method II. These findings indicate that the main reason some deletion strategies of several target metabolites could not be reproduced by Method I within the time limit was the increased complexity of the branch-and-bound trees. Furthermore, comparing problems generated with different numbers of fixed important gene values showed that all measures of problem complexity improved as more gene values were fixed. This result is consistent with Table 6, confirming that fixing additional gene variables makes the problems easier to solve. For additional information, reference is made to Table S5 included in the supplementary file.

Table 7. Solving Process of MILPs for the Target Metabolites where Only Method II Succeeded

Dataset iMM904		Summary of Solving Process (Cases where Only Method I were successful)					
κ [a]	Evaluation Strategy	Avg. Nodes[b]	Avg. Rows[c]	Avg. Columns[c]	Avg. Non-zeros[d]	Avg. Binary[d]	Avg. Cuts[e]
1	St1,St2,St4,St8	7564.88	1117.76	1017.68	4390.54	435.73	60.54
	St3,St5,St6,St7,St9	6482.92	1115.34	1020.15	4416.29	423.25	45.85
2	St1,St2	7145.48	1102.37	1007.38	4336.15	433.26	61.98
	St3,St6	6972.36	1113.88	1015.23	4367.59	422.08	55.22
	St4,St8	7585.80	1090.00	997.19	4277.60	429.46	55.29
	St5,St7,St9	7706.75	1070.42	990.98	4203.45	415.71	63.32
3	St1,St2	6647.53	1105.34	1010.02	4349.61	433.43	47.90
	St3,St4,St6,St8	6685.82	1095.92	1005.43	4307.69	423.82	50.40
	St5,St7,St9	6392.79	1091.18	1005.17	4286.86	421.46	45.61
	Origin. RatGene	5006.71	1162.71	1046.49	4613.01	447.9	56.36

[a] Selected κ important genes associating with Gene-Ranked RatGene.
[b] The average number of nodes searched in branch-and-bound trees, and the smaller the better.
[c] The average number of constraints and variables after pro-solving processes, and the smaller the better. Rows correspond to constraints, and columns correspond to variables in a constraint matrix.
[d] The average number of non-zero counts and binary variables, and the smaller the better.
[e] The average number of cutting planes, and the smaller the better.

According to the Table 6 and Table 7, it is particularly important to note that even when two genes of equal importance, as identified by the framework proposed in this study, are assigned the same fixed value independently, the resulting structures of the two MILP problems will not necessarily be identical. This arises because our proposed strategy does not fully integrate all information from the network. Moreover, even in extremely rare cases where both constructed problems are identical, the reduction in search nodes from assigning values to both genes is not simply twice that achieved by assigning a single gene, since problem construction also depends on other aspects of the network. While correlations exist between gene variables and constructed problems, there is no explicit linear relationship between them. Furthermore, due to the internal workflows of modern MILP solvers integrating multiple mechanisms, a proportional reduction in search nodes does not always yield a proportional reduction in average runtime.

5.2 Related Work of Constraint-based Modeling

Many algorithms have been developed in recent decades to design constraint-based metabolic networks for growth-coupled production. Some of these methods use metabolic pathway analysis based on elementary modes (EMs), which are defined as the minimal sets of reactions that satisfy mass balance under steady-state conditions[30, 31]. EM-based approaches allow for an unbiased identification of all possible reaction pathways[32, 33], aiming to find reaction deletion strategies that enable growth-coupled overproduction[1]. To improve efficiency, a metabolic design algorithm called minL1-FMDL was introduced. This method uses L1-norm minimal modes to reduce the number of candidate reaction deletions compared to traditional EM-based approaches[34]. Additionally, some studies have explored deletion strategies under anaerobic conditions, and these strategies have

shown success on the *E. coli* dataset[35, 36].

Despite these advances, enumerating EMs in large-scale metabolic models remains computationally intensive[1]. To address this limitation, alternative constraint-based programming frameworks have been developed. Some studies use complex Boolean representations of GPR associations to identify gene deletion or addition strategies[20, 24, 37]. OptKnock introduced a bilevel optimization approach that couples cell growth with target metabolite production. It successfully achieved chemical overproduction in *E. coli* through gene deletions[19]. To improve on OptKnock, RobustKnock was developed to enforce stronger growth coupling by ensuring that the target product is an essential by-product of cell growth[38]. OptGene extended the OptKnock framework by applying a genetic algorithm to efficiently search for gene deletion strategies that optimize a phenotypic objective function[39]. gcOpt used a multi-level optimization framework to maximize the minimum guaranteed production rate at a moderate growth rate[40]. A software library was later created to integrate these algorithms[41]. In addition, methods involving subspace partitioning have also shown improved success rates on *E. coli* and yeast models under anaerobic conditions[42, 43, 44]. OptForce used metabolic flux data from wild-type strains to classify reactions based on changes in flux, identifying a minimal set of flux interventions that led to a successful increase in succinate production[21]. OptStrain achieved a balance between high product yield and cell growth by identifying and removing non-native metabolic functions from a universal reaction database[45].

5.3 Comparison with gMCS

The gMCS approach integrates GPR associations into the framework of the MCSEnumerator method[46]. MCSEnumerator is designed for the efficient enumeration of the smallest Minimal Cut Sets (MCSs) in genome-scale metabolic networks[33]. An MCS represents the minimal set of reaction deletions required to block specific target reactions at steady state. Here, minimal means that removing any subset of these reactions would no longer lead to the inhibition of the target reaction. Previous studies have demonstrated that each MCS corresponds to an EM in the dual problem[32]. To identify EMs in the dual space, MCSEnumerator constructs a dual stoichiometric matrix derived from the primal system. gMCS incorporates gene expression data into the dual stoichiometric matrix to identify gene minimal cut sets, building upon the MCSEnumerator framework.

Specifically, gMCS introduces a Γ matrix as part of the dual stoichiometric matrix, which utilizes GPR associations instead of simple one-to-one gene–reaction mappings between gene sets and reaction deletions. In this method, GPR associations are treated as a sufficient condition but not a necessary condition for the construction of the Γ matrix. This effectively serves as an approximation of the true GPR associations. In contrast, our study employs GPR associations directly as constraints by representing them through explicit Boolean functions. Moreover, in gMCS, a flux is constrained to be non-zero if the reaction is not repressed by its associated genes. Conversely, in our model, even when a reaction is not repressed by genes, its flux may still take a value of zero. This allows for a more flexible and accurate representation of metabolic networks.

5.4 Conclusion

In this study, we proposed nine strategies for scoring and ranking gene importance by leveraging a priori knowledge from constraint-based metabolic networks, including GPR associations and network topology. The highly important genes identified through these strategies can be seamlessly integrated into existing algorithms in a plug-and-play manner. Building on this, we developed a Gene-Ranked framework based on RatGene that assigns values for different numbers of important gene variables prior to computation. This framework effectively reduced the complexity of the MILP problems to be solved. For most target metabolites with knockout strategies successfully addressed by the benchmark, our framework significantly reduced computational runtime. They identified many successful cases that the benchmark method could not resolve within the same time limit. Although problem complexity increased in a few instances, it was consistently reduced as more gene variable values were fixed. Overall, this study shows considerable promise for application in other time-sensitive constraint-based algorithms, as it highlights how leveraging a priori information to assign gene variable values without extensive algorithmic modification can substantially reduce problem complexity and accelerate solution processes.

References

1. Mohammadreza Yasemi and Mario Jolicoeur. Modelling cell metabolism: a review on constraint-based steady-state and kinetic approaches. *Processes*, 9(2):322, 2021.

2. Costas D Maranas and Ali R Zomorrodi. *Optimization methods in metabolic networks*. John Wiley & Sons, 2016.

3. Jens Nielsen. Systems biology of metabolism. *Annual review of biochemistry*, 86(1):245–275, 2017.

4. Tuty Asmawaty Abdul Kadir, Ahmad A Mannan, Andrzej M Kierzek, Johnjoe McFadden, and Kazuyuki Shimizu. Modeling and simulation of the main metabolism in escherichia coli and its several single-gene knockout mutants with experimental verification. *Microbial cell factories*, 9(1):88, 2010.

5. Ali Khodayari, Ali R Zomorrodi, James C Liao, and Costas D Maranas. A kinetic model of escherichia coli core metabolism satisfying multiple sets of mutant flux data. *Metabolic engineering*, 25:50–62, 2014.

6. Andreas Karoly Gombert and Jens Nielsen. Mathematical modelling of metabolism. *Current opinion in biotechnology*, 11(2):180–186, 2000.

7. Ronan MT Fleming and Ines Thiele. Mass conserved elementary kinetics is sufficient for the existence of a non-equilibrium steady state concentration. *Journal of Theoretical Biology*, 314:173–181, 2012.

8. George Stephanopoulos, Aristos A Aristidou, and Jens Nielsen. *Metabolic engineering: principles and methodologies*. Elsevier, 1998.

9. M Schauer and R Heinrich. Quasi-steady-state approximation in the mathematical modeling of biochemical reaction networks. *Mathematical biosciences*, 65(2):155–170, 1983.

10. Willi Gottstein, Brett G Olivier, Frank J Bruggeman, and Bas Teusink. Constraint-based stoichiometric modelling from single organisms to microbial communities. *Journal of the Royal Society Interface*, 13(124):20160627, 2016.

11. Tae Yong Kim, Seung Bum Sohn, Yu Bin Kim, Won Jun Kim, and Sang Yup Lee. Recent advances in reconstruction and applications of genome-scale metabolic models. *Current opinion in biotechnology*, 23(4):617–623, 2012.

12. 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*, 3(1):121, 2007.

13. Jeffrey D Orth, Ronan MT Fleming, and Bernhard O Palsson. Reconstruction and use of microbial metabolic networks: the core *escherichia coli* metabolic model as an educational guide. *EcoSal plus*, 2010.

14. Jeffrey D Orth, Tom M Conrad, Jessica Na, Joshua A Lerman, Hojung Nam, Adam M Feist, and Bernhard Ø Palsson. A comprehensive genome-scale reconstruction of *escherichia coli* metabolism—2011. *Molecular systems biology*, 7(1):535, 2011.

15. Charles J Norsigian, Neha Pusarla, John Luke McConn, James T Yurkovich, Andreas Dräger, Bernhard O Palsson, and Zachary King. Bigg models 2020: multi-strain genome-scale models and expansion across the phylogenetic tree. *Nucleic acids research*, 48(D1):D402–D406, 2020.

16. Jeffrey D Orth, Ines Thiele, and Bernhard Ø Palsson. What is flux balance analysis? *Nature biotechnology*, 28(3):245–248, 2010.

17. Axel von Kamp and Steffen Klamt. Growth-coupled overproduction is feasible for almost all metabolites in five major production organisms. *Nature communications*, 8:15956, 2017.

18. Amit Varma and Bernhard O Palsson. Metabolic flux balancing: basic concepts, scientific and practical use. *Bio/technology*, 12(10):994–998, 1994.

19. 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*, 84(6):647–657, 2003.

20. Takeyuki Tamura, Ai Muto-Fujita, Yukako Tohsato, and Tomoyuki Kosaka. Gene deletion algorithms for minimum reaction network design by mixed-integer linear programming for metabolite production in constraint-based models: gdel_minrn. *Journal of Computational Biology*, 2023.

21. 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*, 6(4):e1000744, 2010.

22. Ali R Zomorrodi and Costas D Maranas. Optcom: a multi-level optimization framework for the metabolic modeling and analysis of microbial communities. *PLoS computational biology*, 8(2):e1002363, 2012.

23. Takeyuki Tamura. Metnetcomp: Database for minimal and maximal gene-deletion strategies for growth-coupled production of genome-scale metabolic networks. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2023.

24. Yier Ma and Takeyuki Tamura. Ratgene: Gene deletion-addition algorithms using growth to production ratio for growth-coupled production in constraint-based metabolic networks. *IEEE Transactions on Computational Biology and Bioinformatics*, 2025.

25. Philipp Schneider, Radhakrishnan Mahadevan, and Steffen Klamt. Systematizing the different notions of growth-coupled product synthesis and a single framework for computing corresponding strain designs. *Biotechnology Journal*, 16(12):2100236, 2021.

26. Laurent Heirendt, Sylvain Arreckx, Thomas Pfau, Sebastián N Mendoza, Anne Richelle, Almut Heinken, Hulda S Haraldsdóttir, Jacek Wachowiak, Sarah M Keating, Vanja Vlasov, et al. Creation and analysis of biochemical constraint-based models using the cobra toolbox v. 3.0. *Nature protocols*, 14(3):639–702, 2019.

27. E Robert Bixby, Mary Fenelon, Zonghao Gu, Ed Rothberg, and Roland Wunderling. Mip: Theory and practice—closing the gap. In *IFIP Conference on System Modeling and Optimization*, pages 19–49. Springer, 1999.

28. Ailsa H Land and Alison G Doig. An automatic method for solving discrete programming problems. In *50 Years of Integer Programming 1958-2008: From the Early Years to the State-of-the-Art*, pages 105–132. Springer, 2009.

29. James E Kelley, Jr. The cutting-plane method for solving convex programs. *Journal of the society for Industrial and Applied Mathematics*, 8(4):703–712, 1960.

30. Jürgen Zanghellini, David E Ruckerbauer, Michael Hanscho, and Christian Jungreuthmayer. Elementary flux modes in a nutshell: properties, calculation and applications. *Biotechnology journal*, 8(9):1009–1016, 2013.

31. Predrag Horvat, Martin Koller, and Gerhart Braunegg. Recent advances in elementary flux modes and yield space analysis as useful tools in metabolic network studies. *World Journal of Microbiology and Biotechnology*, 31(9):1315–1328, 2015.

32. 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*, 28(3):381–387, 2012.

33. Axel von Kamp and Steffen Klamt. Enumeration of smallest intervention strategies in genome-scale metabolic networks. *PLoS computational biology*, 10(1):e1003378, 2014.

34. Takeyuki Tamura. L1 norm minimal mode-based methods for listing reaction network designs for metabolite production. *IEICE TRANSACTIONS on Information and Systems*, 104(5):679–687, 2021.

35. Steffen Klamt, Georg Regensburger, Matthias P Gerstl, Christian Jungreuthmayer, Stefan Schuster, Radhakrishnan Mahadevan, Jürgen Zanghellini, and Stefan Müller. From elementary flux modes to elementary flux vectors: Metabolic pathway analysis with arbitrary linear flux constraints. *PLoS computational biology*, 13(4), 2017.

36. 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*, 12(3):173–186, 2010.

37. Takeyuki Tamura. Trimming gene deletion strategies for growth-coupled production in constraint-based metabolic networks: Trimgdel. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2022.

38. Naama Tepper and Tomer Shlomi. Predicting metabolic engineering knockout strategies for chemical production: accounting for competing pathways. *Bioinformatics*, 26(4):536–543, 2010.

39. Kiran Raosaheb Patil, Isabel Rocha, Jochen Förster, and Jens Nielsen. Evolutionary programming as a platform for in silico metabolic engineering. *BMC bioinformatics*, 6(1):308, 2005.

40. Tobias B Alter and Birgitta E Ebert. Determination of growth-coupling strategies and their underlying principles. *BMC bioinformatics*, 20(1):447, 2019.

41. Philipp Schneider, Pavlos Stephanos Bekiaris, Axel von Kamp, and Steffen Klamt. Straindesign: a comprehensive python package for computational design of metabolic networks. *Bioinformatics*, 38(21):4981–4983, 2022.

42. Takeyuki Tamura. Grid-based computational methods for the design of constraint-based parsimonious chemical reaction networks to simulate metabolite production: Gridprod. *BMC bioinformatics*, 19(1):325, 2018.

43. Yier Ma and Takeyuki Tamura. Dynamic solution space division-based methods for calculating reaction deletion strategies for constraint-based metabolic networks for substance production: Dyncubeprod. *Frontiers in Bioinformatics*, 1:716112, 2021.

44. Takeyuki Tamura. Efficient reaction deletion algorithms for redesign of constraint-based metabolic networks for metabolite production with weak coupling. *IPSJ Transactions on Bioinformatics*, 14:12–21, 2021.

45. Priti Pharkya, Anthony P Burgard, and Costas D Maranas. Optstrain: a computational framework for redesign of microbial production systems. *Genome research*, 14(11):2367–2376, 2004.

46. Iñigo Apaolaza, Edurne San José-Eneriz, Luis Tobalina, Estíbaliz Miranda, Leire Garate, Xabier Agirre, Felipe Prósper, and Francisco J Planes. An in-silico approach to predict and exploit synthetic lethality in cancer metabolism. *Nature communications*, 8(1):459, 2017.