Bi-level Optimization: Optimistic and Pessimistic Algorithm for Conflicting Objectives in Metabolic Engineering

Jose Alexander Vindel Garduno
Registration Number: 201991298

Primary Supervisor: Dr. Ashwin Arulselvan
Secondary Supervisor: Dr. Mahdi Doostmohammadi

Department of Management Science
Strathclyde Business School
November 2024

Contents

List of Figures

List of Tables

1 Introduction

In recent years, bi-level programming has emerged as a pivotal concept in the realm of optimization, addressing the inherent complexity found in hierarchical decision-making environments. This intricate framework involves two levels of decision-makers, termed the leader and the follower, each with distinct objectives and constraints that are interdependent. The leader's decisions influence the follower's feasible set, while the follower's decisions, in turn, impact the leader's outcomes. Such a dynamic interplay naturally lends itself to modeling via bi-level programming, capturing the essence of interactions as seen in Stackelberg games (Shi, G. Zhang, and Lu 2005; Stackelberg 2011; Dempe 2015).

The significance of bi-level programming is further exemplified by its wide array of applications across various domains. However, the structure of bi-level problems invariably renders them NP-hard, posing significant computational challenges (Jeroslow 1985; J. F. Bard 1991). This complexity necessitates the development of efficient solution methodologies, such as reformulation techniques and sophisticated algorithmic strategies, to render these problems tractable (Deng 1998; Pineda, Bylling, and Morales 2017; Fischetti 2018).

One domain where bi-level programming has been extensively applied is network optimization, exemplified by bi-level flow and routing problems. These problems often involve scenarios where a leader, such as a toll authority, seeks to optimize toll pricing strategies, while the follower, representing network users, optimizes their routing decisions based on the imposed tolls (P. Marcotte 1986; Labbé, Patrice Marcotte, and Gilles Savard 1998a). The iterative interplay between pricing strategies and user responses in congested networks aptly demonstrates the power of bi-level models to capture and solve real-world problems with hierarchical decision structures.

Another noteworthy application is found in the field of metabolic engineering, where bi-level optimization frameworks are employed to redesign metabolic networks for enhanced biochemical production. In this context, the leader (engineer) aims to maximize the production of specific chemicals through genetic modifications, while the follower (microbe) naturally seeks to maximize growth, creating a complex bi-objective landscape (Burgard, Pharkya, and Maranas 2003; Tepper and Shlomi 2010). The innovative application of bi-level programming in this arena underscores its potential to drive advancements in biotechnology and industrial bioprocesses. Furthermore, the versatility of bi-level programming extends to network interdiction problems, where an attacker (leader) seeks to disrupt a network optimized by a defender (follower) to minimize losses or maximize flow (Smith, Prince, and Geunes 2013). These models are integral to understanding and mitigating vulnerabilities in critical infrastructure and defense systems.

In light of the challenging nature of bi-level problems, our research focuses on

developing efficient solution techniques, anchored in cutting-plane and branch-and-bound methodologies. By exploring both strong duality and KKT-based reformulations, we strive to advance the state-of-the-art in solving bi-level optimization problems, offering novel insights and solutions to enhancing chemical production in metabolic networks.

this dissertation contributes to the broader understanding and application of bi-level programming by elucidating its theoretical underpinnings, computational strategies, and real-world implications. As it delves deeper into solution methodologies and their applications, it aims to bridge the gap between theoretical constructs and practical implementation, Ultimately, providing a novel algorithm that will facilitate more informed decision-making processes across disciplines characterized by hierarchical interactions with special attention to metabolic engineering and chemical production.

2 Literature Review

2.1 Bi-level Programing I

Bi-level problems represent a nested hierarchical structure for two decision makers. Usually referred as leader and follower. This structure best describes a Stackelberg game where players try to optimize their objective respectively. Stackelberg first formulated this relationship on his work on market economy (Stackelberg 2011) to describe when a player's decision has the ability to influence over the second player's objective, but neither player completely dominates the other (Dempe 2015).In other words, both players objectives rely on the other player's decision (Shi, G. Zhang, and Lu 2005). Generally both objectives are in a competing state, meaning that leader and follower have their own independent objective with some shared variables. Such characteristics make the bi-level problem NP-hard (Jeroslow 1985; J. F. Bard 1991).In other words, not solvable in polynomial time and require enormous computational iterations (Deng 1998; Pineda, Bylling, and Morales 2017).

Interestingly, researchers have given more attention to Bi-level problems mainly because Bi-level models are convenient when representing structures owned by one party but controlled by another (competing entities) (DeNegre 2011). Scenarios where this type of relationship between decision makers can be found in different areas. Areas such as toll pricing (Labbé, Patrice Marcotte, and Gilles Savard 1998b), gas distribution(Dempe et al. 2011), electrical grids (Delgadillo, Arroyo, and Alguacil 2010), (Morton, Pan, and Saeger 2007), police enforcement (Ajay, Thomas, and Chase 2010), metabolic engineering (Burgard, Pharkya, and Maranas 2003; SunXu et al. 2013; Tepper and Shlomi 2010), and the list keeps growing. The reader is directed to the work of (Dempe 2015; Smith, Prince, and Geunes 2013; Benoít Colson, Patrice Marcotte, and Gilles Savard 2005) to find a more comprehensive details on the areas of application. These models fall into the category of linear bi-level problems (LBLP).The commonality between the models mentioned previously is that the constraints are linear functions and a subset of the variables belongs in $\mathbb{Z}_2$.

A LBLP is generally formulated as (1) according to (J. F. Bard 1991):

$$\min_{x \in X} \quad F(x, y) = c_1 x + d_1 y \tag{1a}$$

$$\text{subject to} \quad A_1 x + B_1 y \leq b_1 \tag{1b}$$

$$\min_{y \in Y} \quad f(x, y) = c_2 x + d_2 y \tag{1c}$$

$$\text{subject to} \quad A_2 x + B_2 y \leq b_2 \tag{1d}$$

Where equation (1a) represents the leader's objective function, equation 1c is the follower's objective function. Let's note that both decision makers have

their own set of variables (x, y), in this case the leader's decision variable is x and the follower's decision variable is y. Such variables can either be present in both levels or just in one. Let's also note that the set of the leader's constraints (1b) also includes the follower's entire optimization problem. From the model in (1) (J. F. Bard 1991) defines the following:

1. The constraint region of the LBLP as:

$$S = \{(x, y) : x \in X, y \in Y, A_1x + B_1y \leq b_1, A_2x + B_2y \leq b_2\}$$

2. For each fixed $x \in X$ the follower's feasible set is then:

$$S(x) = \{y \in Y : B_2y \leq b_2 - A_2x\}$$

   Which is all the $y^s \in Y$ that satisfy the follower's constraints when a fixed x is evaluated.

3. The projection of S onto the leader's decision space:

$$S(X) = \{x \in X : \exists y \in Y, A_1x + B_1y \leq b_1, A_2x + B_2y \leq b_2\}$$

   In other words, the set of x for which there exists y from the follower's feasible set that satisfies both the leader's and follower's sets of constraints.

4. The follower's rational reaction set for $x \in S(X)$:

$$P(x) = \{y \in Y : y \in \text{argmin}[f(x, \hat{y}) : \hat{y} \in S(x)]\}$$

   Where $\text{argmin}[f(x, \hat{y}) : \hat{y} \in S(x)] = \{y \in S(x) : f(x, y) \leq f(x, \hat{y}), \hat{y} \in S(x)\}$ basically, this defines the follower's response where there is a y from the follower's feasible that provides the best solution.

5. The inducible region:

$$IR = \{(x, y) : (x, y) \in S, y \in P(x)\}$$

   IR represents the set of the leader's decision variables and corresponding follower's optimal solution(s) that belong to a feasible constraint region.

Usually there are some assumptions made for models such as (1). S is nonempty and compact, $P(x) \neq \emptyset$, in other words, for a decision by the leader, the follower has possible responses. Under these assumptions the leader can take two postures regarding the follower's response.

(a) The leader assumes total cooperation from the follower, therefore the follower will choose an option that involves a higher leader's objective function

(b) The follower cannot be influenced, so the leader can assume the follower will choose a solution that does not lead to the best leader's objective function.

The case described in (a) is defined in the literature as the *optimistic approach*, the *pessimistic approach* (b). Throughout the literature (a) and (b) are the main alternatives for the LBLP. There is a wider exploration to (a) in the literature. Its assumption allows for single level reformulations with proper equilibrium constraints replacing the follower's problem (Dempe 2003).

2.2 Flow and Routing Problems under a Bi-level framework

The work of bi-level models in networks had been introduced by (P. Marcotte 1986) in a toll pricing setting.Commonly, toll pricing problems describe a situation where the follower decides over routing options from an origin node to a sink node based on costs for using a path while the leader prices the different possible paths. The problem consists in finding the optimal cost balance for a congested network under traffic equilibrium. Later expanded to consider the case of a deterministic toll-pricing problem in (Labbé, Patrice Marcotte, and Gilles Savard 1998a) applied in a taxation problem where the leader is capable of taxing products or services all the while the follower optimizes its own objective by considering the taxation planned by the leader. Usually, these problems are solved by a reformulation as mixed integer program and then using short path algorithms to find its solution. In some special cases, the problem can be solved in polynomial time. The method consists in reformulating the problem using binary variables. However, most models lack on incorporating some real-life features. The toll pricing model serves as a foundation for further works with different considerations and assumptions. Such is the case of (Dokka et al. 2016) who employed a two-point heuristics and robust optimization to solve the toll pricing problem under uncertainty, with a closing note suggesting that the simple nature does not fully show that the solution is in fact optimal for a non-deterministic case. The case for a multi-commodity transportation network under uncertainty was explored by (Alizadeh, P. Marcotte, and G. Savard 2013) where the goal of the model is to estimate the revenue loss when randomness is not taken into consideration. More recently, a bi-level model designed to solve optimal toll rates in port traffic network design was solved with a method based on memetic heuristics, where the upper level describes waiting times and the lower levels describes user equilibrium (H. Zhang, Q. Zhang, and Chen 2019). The process generates a random initial population and through genetic operations obtain an offspring population, then the second population is optimized using tabu search, the updated population is kept after a selection operation, only keeping those individuals presenting high fitness and abandoning those who do not meet the criterion, the process is repeated until reaching end conditions.

Different methodologies have been used to solve the bi-level case on networks. Cutting planes where introduced in every iteration of the algorithm to solve the model proposed in (Hearn et al. 2001) to solve for fixed demand toll pricing problems. This method becomes increasingly more complex accordingly to the network complexity. In the work by (X. Zhang and Wee 2012), where the objective is to optimize network capacity by introducing toll pricing (congestion prices). The network capacity improvement is calculated through the implementation of mixed integer variables to linearize constraints by introducing new variables for each type of constraint. This method leads to single level problems that can determine a global optimum at the expense of increasing the number of variables to consider. Simulation has also played a part on the efforts to

deal with the dynamic nature of the network itself (He 2017), however, this approach comes with higher computational times as the process involves an extra step for transport simulation. More recently, complementary methods have been developed to account for both deterministic and stochastic cases based on local search method derived from the global optimum and a bi-level trust region, in the work of (Gilbert, Patrice Marcotte, and Gilles Savard 2015) a method is proposed to consider when users are discretely allocated to paths in the network. LP duality has also been used to solve the bi-level case (Burgard, Pharkya, and Maranas 2003), in the metabolic engineering framework, to optimize the individual flows in the network which ultimately will lead to chemical overproduction. In the same framework, a multi-objective approach has been developed and solved through an enumeration technique (Andrade et al. 2020).

A branch and bound algorithm was expanded during the 1990's to include MIBLP (Moore and Jonathan F. Bard 1990). This work was an early attempt to develop an all-around algorithm to solve MIBLP. These algorithm comes with its computational costs.The method will require better bounds as the number of variables increases. To work around, heuristics can then be employed trading between speed and accuracy, and occasionally optimality may never be confirmed. Work on cutting planes has been developed to complement the all-around algorithm.The work in (Denegre and Ralphs 2009) discusses a way to work with branch and cut and generate valid inequalities. The idea is to solve a linear relaxation, then generate valid inequalities to improve bounds and branch when necessary.

As of 2007 the most common solution approaches for different cases of bi-level problems were: Vertex enumeration, where all functions involved are linear and the set of variables are polyhedral and Branch and Bound, usually for problems where the lower level is convex and regular and can be reformulated into a single level problem, applications of these can be found with Mathematical Programs with Equilibrium Constraints (MEPCS), which sometimes is another word for bi-level models, (Benoît Colson, Patrice Marcotte, and Gilles Savard 2007).

Bounds are extremely important in the work of bi-level problems, they provide useful information to compute feasible solutions. Specially, this is true for bi-level knapsack problems, where the use of good lower and upper bound can be used to fix the variables of the leader to their optimal values forming a set of different solutions later categorize is a set of different best solutions to then solve different feasible problems (Raid et al. 2012).

Contemporary work, where both the leader's and the follower's variables are mixed integer, have produced algorithms based on the use of intersection cuts, which rely on defining convex-feasible sets that cut off the infeasible points in the problem relaxation (Fischetti 2018), a key assumption is that the relaxation is compact and feasible. Another assumption is that the leader's variables influencing the follower's decisions are integer and bounded. Some problems under such assumptions can be seen in knapsack problems. Some variants of the bi-level knapsack problem cannot be solved in polynomial time, rather a

polynomial time approximation is designed (Caprara et al. 2013), this work also includes description of variants on the bi-level knapsack problem.

Most of the linear reformulations are KKT based, this means, the introduction of a new binary variable for each of the complementary slackness conditions at the lower level. It easily becomes impractical for some bi-level problems. Strong Duality based reformulations can be the solution to such impracticality. SD-based reformulations assume that KKT conditions can be replaced by a single quadratic constraint, where the latter can be then linearized considering a binary decomposition, only if the upper-level variables are integer. This method can result in adding more linear constraints and discrete variables. (Zare et al. 2019), This method can be particularly useful when reformulations based on KKT introduce more discrete variables than reformulations based on SD, usually problems where the lower level is larger than the upper counterpart.

2.3 Network Interdiction

Possibly one of the best examples for the LBLP comes from the network interdiction problems. In this setting leader could also be called *interdictor, adversary, attacker* and the follower *operator, owner, defender* (Smith, Prince, and Geunes 2013), the follower optimizes its objective over the network and the leader will alter the owner's network, usually limiting the possible arcs available. Consider the example of a pipeline network where the owner (follower) is interested in optimizing the flow from one node to a particular node over the network, typically maximum flow problems. And an interdictor (leader) who may try to restrict a strategic set of links in the network for its own advantage, thus limiting the owner's overall utility from the network, *minimax*. In this example the follower is aware of the links which have been limited and adjusts the flow accordingly, that is the flow distribution on the pipes will be different.
The general formulation of the minimax network interdiction is

$$\begin{aligned} \min_{y \in Y} \max \quad & p(y)^t v \\ \text{s.t.} \quad & Dv \leq r(y) \\ & v \geq 0 \end{aligned}$$

Where y and v are the leader and follower variables, respectively, and where $p(y)$ and $r(y)$ represent the follower's profits and available resources, in other words $r(y)$ is the right hand side in the feasible set. The constraint $Dv \leq r(y)$ usually consist of flow conservation and capacity constraints. Oftentimes Y, the leader's feasible region, consists of a single knapsack constraint, thus restricting the leader to make binary decisions.

2.4 Metabolic Engineering and Interdiction Problems

Metabolic networks are a representation of the relationships between the processes carried out inside an organism (Lee et al. 2012). These relationships are similar to the relationships found in network optimization, where the interest is to calculate flow between nodes that conform the structure. For some networks, the flow defines goods being moved from one place to another. In the case of metabolic engineering, the flow is defined as the amount of mass that is being moved between metabolites (nodes) over reactions (links, arcs). The network usually is described by a $N \times M$ where N is the number of metabolites and M is the number of reactions. Usually M is larger in comparison to N, describing an undetermined system.

An example of network interdiction problems, expressed as bi-level linear problems, where the lower-level is a linear program and the upper-level is an integer program comes from the metabolic engineering paradigm, explored in the paper (Burgard, Pharkya, and Maranas 2003) where the leader (interdictor) tries to disrupt a network by deleting arcs (reactions) to improve the production of biochemicals. While the follower (network owner) tries to optimize its own flow through the restricted network. The model for Flux Balance Analysis (FBA) is the network owner in the LBLP.

2.4.1 Flux Balance Analysis as lower-level decision maker

This subsection is optional for the reader. However, its presence in this work is to support and contextualize the ideas behind metabolic networks. The finer details of metabolic network engineering are not in the scope if this section. The work in (Edwards, Covert, and B. Palsson 2002), (Orth, Thiele, and Bernhard O Palsson 2010), (Vital-Lopez, Memišević, and Dutta 2012), (Zomorrodi et al. 2012), (Raman and Chandra 2009), and (Anand, Mukherjee, and Padmanabhan 2020) complement each other to offer a rich source for understanding the principles of FBA and biological networks.
FBA provides a methodology to estimate the flow of metabolites through the network.This methodology makes it possible to predict production rates (the amount of mass that results from a particular reaction) and compute the flow distribution in the network. Mathematically, FBA is expressed as model 3.

$$\max \nu_{biomass} \tag{3a}$$

$$\sum S_{ij} * \nu_j = 0 \tag{3b}$$

$$\nu_j^{min} \leq \nu_j \leq \nu_j^{max} \tag{3c}$$

The Stoichiometric coefficients S_{ij} are the amount of mass that is needed from one arc to produce a metabolite, much like the coefficients on the arcs in the max flow problems. Thus, the S coefficient helps to construct the conservation constraints under steady state, the total flow in a particular node that is not the

source or sink is zero, ie the flow that enters a node is the same flow that exits the node. Each metabolite is governed by a number of reactions. $S_{ij}\ \nu_j = 0$ ensures that every metabolite (node) is balanced to a steady state and also in line with mass conservation laws.
To close the search space for the FBA every reaction (link,arc) has a capacity. The flow on each arc is denoted by ν_j, each arc is bounded by 3c where ν^{min} and ν^{max} are the lower and upper bound respectively.

2.4.2 Gene Deletion and Bi-level Problems

Is a rational metabolic engineering strategy to predict allowable gene deletions that could lead to the overproduction of a desirable compound (reaction product). In this context, gene deletion refers to restricting the flow through a particular reaction to zero (knockout) (Edwards, Covert, and B. Palsson 2002). This modifies the flow to a more desirable outcome. The question then shifts to finding the best strategies for overproduction without hindering the cell's growth. One possible way to address this issue is answered in the work where the bi-level OptKnock was first proposed (Burgard, Pharkya, and Maranas 2003). This is mathematically expressed as model 4, where the objective is to identify the best gene deletion that leads to the chemical maximization. Model 4 clearly reflects the bi-level nature of the problem in gene deletion, as the leader's problem has a nested problem enclosed in the inner box. The model in the inner box is no other than the FBA model reacting to the $\hat{y}$ solution from the master problem. Model 5 is a simplified version of model 4.

$$
\begin{aligned}
&\max_{y_j}\ \nu_{chemical} \\
&\text{subject to} \\
&\sum_{j \in M} (1 - y_j) \leq K \\
&\quad \boxed{\begin{aligned}
&\max_{\nu_j}\ \nu_{biomass} \\
&\sum_{j=1}^{M} S_{ij} * v_j = 0 \\
&\nu_j^{min} \,.\, y_j \leq \nu_j \leq \nu_j^{max} \,.\, y_j, \quad \forall j \in M
\end{aligned}} \qquad y_j = \{0, 1\}, \quad \forall j \in M
\end{aligned}
\tag{4}
$$

$$\max\ \nu_{chemical} \tag{5a}$$

$$\sum (1 - y_j) = K \tag{5b}$$

(3) FBA

Models 2 and 5 share some similitude. In both models the leader is an external entity whose decision making is based on binary variables restricted to a knapsack whose interest is to shape the network for profit. The follower is an independent entity owner of the network whose variables are, in essence,

continuous and adapts its overall flow based on the leader's decision. Equation $\nu \geq 0$ in model 2 becomes $LB \leq \nu \leq UB$ in model 4 to completely describe the transport fluxes in the metabolic network. Positive values correlates with the uptake of a particular metabolite, inversely negative values conform to secretions (Burgard, Pharkya, and Maranas 2003).

3 Bi-level Solving Methods

In this section two methods to solve the bi-level problem in (ibid.) are described. The first method corresponds to a single level reformulation, the second method correspond to the proposed algorithm in this work. The method relies on cutting off infeasible solutions and improving the bounds until a solution is found.

3.1 Single Level Reformulation

The MILP reformulation to solve a bi-level linear problem requires the dualization of the follower's problem. This can be done under certain circumstances, one of them being under the assumption of strong duality. Where the solution of the primal model is the same as the solution of its dual model. It is important to keep in mind that the solution to the inner problem is in itself a convex set, and the maximal objective function value can be estimated by different feasible flow distributions in the network (Tepper and Shlomi 2010).

The MILP from 1 mathematically is expressed as model 6, note the inclusion of the follower's constraints into leader's set of constraints (equation 1d)

$$\min_{x \in X} \quad F(x, y) = c_1 x + d_1 y \tag{6a}$$

$$\text{subject to}$$

$$A_1 x + B_1 y \leq b_1 \tag{6b}$$

$$A_2 x + B_2 y \leq b_2 \tag{6c}$$

$$b2\ \lambda \geq 0 \tag{6d}$$

$$\lambda(A_2 + B_2) >= c_2 + d_2 \tag{6e}$$

This method, as stated before, introduces λ as a dual variable from the follower's problem. By introducing a new variable, in some cases, there will be non linear terms in the objective function. The non linear terms can be expressed in a linear form by introducing additional variables and a system of linear inequalities.

3.1.1 OptKnock

The bi-level model proposed in (Burgard, Pharkya, and Maranas 2003) reformulates into a single level MILP. The follower's model proposed in (ibid.) is model 7.

$$\max \nu_{biomass} \tag{7a}$$

$$\sum_{j=1}^{M} S_{ij} * \nu_j = b_i \tag{7b}$$

$$\nu_{pts} + \nu_{glk} = \nu_{glc-uptake} \tag{7c}$$

$$\nu_{atp} \geq \nu_{atp-main} \tag{7d}$$

$$\nu_{biomass} \geq \nu_{target} \tag{7e}$$

$$\nu_j^{min} * y_j \leq \nu_j \leq \nu_j^{max} * y_j \tag{7f}$$

Equation 7f includes the term y_j which is the leader's binary variable performing the gene deletion (Knockout). Equations 7c and 7d constitute biological assumptions to fuel the network while equation 7e assures a minimum growth and cell viability. The dual of model 7 is expressed as model 8.

$$\min \nu_{atp-main} * \mu_{atp} + \nu_{target} * \mu_{biomas} + \nu_{glc-uptake} * glc \tag{8a}$$

$$\sum \lambda_i^{stoich} S_{i,glk} + \mu_{glk} + glc = 0 \tag{8b}$$

$$\sum \lambda_i^{stoich} S_{i,pts} + \mu_{pts} + glc = 0 \tag{8c}$$

$$\sum \lambda_i^{stoich} S_{i,biomas} + \mu_{biomas} = 1 \tag{8d}$$

$$\sum \lambda_i^{stoich} S_{i,j} + \mu_j = 0 \tag{8e}$$

$$\mu_j^{min} * (1 - y_j) \leq \mu_j \leq \mu_j^{max} * (1 - y_j) \tag{8f}$$

$$\mu_j \geq \mu_j^{min} * (1 - y_j) \tag{8g}$$

$$\mu_j \leq \mu_j^{max} * (1 - y_j) \tag{8h}$$

$$\mu_j \in \Re \tag{8i}$$

$$\lambda_i^{stoich} \in \Re \tag{8j}$$

$$glc \in \Re \tag{8k}$$

Where λ_i, μ_j, are the dual variables associated with constraints 7b and 7f respectively. And equation 8a describes the objective function.
Note that some variables in the objective function are non linear. Specifically, the dual variables associated with the constraint 7f.
The linearization of the product of a continuous variable, μ_j, and a binary variable, y_j, when $\mu_j \in \left[\mu_j^{min}, \mu_j^{max}\right]$ can be easily performed. However, in the original paper it's not really clear how to estimate the parameters μ^{min} and μ^{max}.Moreover, the methodology did not include the follower's primal variables in the dual objective. The work published in ReacKnock (SunXu et al. 2013) points out this issue and offers a better explanation for the MILP reformulation.

3.1.2 ReacKnock

The models in OptKnock and ReacKnock are reformulated essentially with the same method. However, in ReacKnock the constraints 7d, 7c and 7e are included as part of the constraint 7f. In other words, all reaction requirements are now part of the lower bounds. Thus, making some lower-bound constraints redundant. The inner formulation (model 9) turns out to be more compact and easier to trace.

$$\max \nu_{biomass} \tag{9a}$$

$$\sum S_{ij} * \nu_j = 0 \tag{9b}$$

$$\nu_j^{min} \leq \nu_j \leq \nu_j^{max} \tag{9c}$$

$$\nu_j^{min} y_j \leq \nu_j \leq \nu_j^{max} y_j \tag{9d}$$

Its dual then can be written as model 10. Where $\alpha 1$, $\beta 1$, α, β are the dual variables associated with constraints 9c and 9d. And BM is a sufficiently large number.

$$\min \sum(\alpha 1_j * UB_j - \beta 1_j * LB_j) + \sum(\alpha 2_j * UB_j - \beta 2 * LB_j) \tag{10a}$$

$$\sum \lambda_i * S_{i,j} - \beta_i + \alpha_i - \beta 2_i + \alpha 2_i = 0 \tag{10b}$$

$$\lambda_{biomas} * S_{biomas,j} - \beta_{biomas} + \alpha_{biomas} - \beta 2_{biomas} + \alpha 2_{biomas} = 1 \tag{10c}$$

$$\alpha 1 \leq BM * (y_j) \tag{10d}$$

$$\alpha 1 \geq -BM * (y_j) \tag{10e}$$

$$\alpha 1 \leq \alpha + BM * (1 - y_j) \tag{10f}$$

$$\alpha 1 \geq \alpha - BM * (1 - y_j) \tag{10g}$$

$$\beta 1 \leq BM * (y_j) \tag{10h}$$

$$\beta 1 \geq -BM * (y_j) \tag{10i}$$

$$\beta 1 \leq \beta + BM * (1 - y_j) \tag{10j}$$

$$\beta 1 \geq \beta - BM * (1 - y_j) \tag{10k}$$

The single-level MILP reformulation of the bi-level problem, model 11, proved to be efficient when transcribed into a working programming language. And, the best for direct comparison in terms of computing time and solution quality.

$$\max \nu_{chemical} \tag{11a}$$

$$\sum S_{i,j} * \nu_j = 0 \tag{11b}$$

$$\sum(1 - y_j) = k \tag{11c}$$

$$\nu_j^{min} \leq \nu_j \leq \nu_j^{max} \tag{11d}$$

$$\nu_j^{min} * y_j \leq \nu_j \leq \nu_j^{max} * y_j \tag{11e}$$

$$\nu_{biomas} = (10a) \tag{11f}$$

$$\vdots \qquad \text{Dual Constraints}$$

$$(10k)$$

MILP reformulations in both cases need good bounds and a sufficiently large BM to work properly and can be solved using commercial solvers.

3.1.3 RobustKnock

The linearization followed by (Tepper and Shlomi 2010) in their (RobustKnock) model to guarantee more robust chemical rate production is, essentially, the same as those of OptKnock and ReacKnock. However, a crucial difference is the setup of the leader's objective. In RobustKnock the leader's objective is defined as, $\max \min \nu_{chemical}$, this is to maximize the minimal possible rate of the bioengineering objective achieved by performing gene knockouts. While, the follower's objective remains identical as in the original, see model 4. The linearization leads to adding dual variables and the boolean complements to resolve the bi-linear terms. This method acknowledges that there might be solutions that lead to unbounded solutions, eg. when the knockout doesn't enable a biomass production larger than the estimated threshold. To overcome this issue Robustknock introduces additional decision variables to enforce feasibility of the primal minimization problem.

4 A unified branch and cut framework for metabolic engineering

4.1 Introduction

The idea of modifying a microbe's structure to increase the production of a desired chemical is not new. This describes a hierarchical relationship between the engineer, interested in optimizing chemical production, and the microbe, whose interest lies in cellular growth. Metabolic engineers rely on a mathematical approach to analyze the flow of metabolites through a metabolic network and compute chemical production rates called Flux Balance Analysis (FBA) (Edwards, Covert, and B. Palsson 2002; Orth, Thiele, and Bernhard O Palsson 2010). A metabolic network is a complex grid of biochemical reactions that occur within an organism. In this grid, the nodes are the metabolites and the connecting edges (arcs) are the reactions of the organism. When organisms are represented as metabolic networks, it provides a convenient framework to study the results of engineered modifications often via gene knockouts (arc intervention). Modified organisms have a positive impact on optimized chemical production, better strains for agriculture, and better-designed medications in the pharmaceutical industry (Volk et al. 2023). Gene deletion (knockout) is a popular technique to modify an organism into a more economically valuable entity that removes or activates specific genes within an organism (Griffiths 2020). Switching subsets of arcs on and off would redistribute the network flow yielding different production levels of a desired chemical. Mathematically, this can be modeled by mapping the arcs in the metabolic network to binary variables and finding a suitable assignment of binary values to them that maximizes chemical yield. Every organism's growth response is different according to the genes that are activated resulting in a combinatorial problem that can be efficiently modeled using integer programming.

A substantial body of literature provides optimistic gene deletion strategies, suggesting that organisms with modified networks will adapt to favor chemical production (Pharkya and Maranas 2006; Pharkya, Burgard, and Maranas 2004; Tamura et al. 2018; Kim and Reed 2010; Patil et al. 2005; Burgard, Pharkya, and Maranas 2003; SunXu et al. 2013; Brochado and Patil 2014). However, most organisms are set to follow their evolutionary goal that favors cellular growth, thus the intervened organism will not necessarily yield a maximum chemical production.

To handle this, typically, pessimistic gene deletion strategies are designed to expect the worst outcome (Apaydin et al. 2017; Merkert, Orlinskaya, and Weninger 2022). The literature points to mixed integer bi-level programming and its single-level reformulations as the predominant methodology for both optimistic and pessimistic approaches. However, to the best of our knowledge, there are very few pessimistic approaches proposed in the literature. Our contribution is to present a unified framework where a decision maker can choose either approach and efficiently compute gene deletion strategies. We also in-

troduce cuts that are stronger than no-good cuts (Balas and Jeroslow 1980) and demonstrate that our algorithm is competent when compared with state-of-the-art. Hierarchical structures with competing objectives can be found in most research fields. Therefore, presenting our framework to solve optimistic and pessimistic bi-level gene deletion has a positive impact as it expands both the literature and the tools available. In addition, we also demonstrate that our method is computationally competent when compared with the most prevalent techniques.

Bacteria such as E. coli offer a possible alternative for commercial production of bioproducts and biofuels (Rosales-Calderon and Arantes 2019; Na D et al. 2012). To validate our approach, we tested our proposed algorithm for the overproduction of different chemicals while maximizing cellular growth from different E. Coli reconstructions. Table 1 holds the different metabolic E. Coli reconstructions, i.e. iJO1366 (Orth, Conrad, et al. 2011), iAF1260 (Bernhard O Palsson et al. 2007), and iJR904 (Reed et al. 2003) and the chemicals to optimize.

Chemical	E. coli Strain		
	iJO1366	iAF1260	iJR904
Succinate	•	•	
Formate	•	•	•
Acetate	•	•	•
Lactate	•	•	
Ethanol	•	•	
Fumarate		•	•

Table 1: Chemical byproducts from E. coli for validating the algorithm

The rest of the paper is organized as follows. In Section 4.2, we formally introduce bi-level programming and discuss several solution approaches. Section 4.3 presents the fundamental concepts in metabolic engineering and gene deletion. In Section 4.4, we elaborate on our solution approach, which involves an algorithmic framework that generates and adds cuts based on the decision maker's chosen strategy. Section 4.5 details the biological assumptions and computational experiments conducted to evaluate our algorithms and presents our findings. Finally, our conclusions are summarized in Section 4.7.

4.2 Bi-level programming preliminaries

Bi-level problems represent a nested hierarchical structure for two decision-makers, usually called leader and follower, and their respective problems, upper and lower levels. Each level has its variables and constraints. And, there is at least one set of variables that can appear in the constraints of both levels known as *linking variables*. It is through the linking variables that the lower-level problem is parameterized. In other words, both players' objectives rely on the other player's decision (Shi, G. Zhang, and Lu 2005). This structure best describes

a Stackelberg game where players try to optimize their objectives respectively. Stackelberg first formulated this relationship in his work on market economy (Stackelberg 2011) to describe when a player's decision can influence the second player's objective, but neither player completely dominates the other (Dempe 2015). Such characteristics make the bi-level problem NP-hard (Jeroslow 1985; J. F. Bard 1991). In other words, bi-level problems are usually not tractable and require enormous computational effort (Deng 1998; Pineda, Bylling, and Morales 2017).

A Bi-Level Programming Problem (BLPP), represented as problem (12), can be formulated as follows (see Dempe 2015; Schmidt and Beck 2023; Bracken and McGill 1973):

$$\min_{\boldsymbol{y} \in Y} f(\boldsymbol{\nu}, \boldsymbol{y}) \tag{12a}$$

$$\text{s.t. } F(\boldsymbol{\nu}, \boldsymbol{y}) \geq 0 \tag{12b}$$

$$\boldsymbol{\nu} \in \arg\min_{\boldsymbol{\nu'} \in \mathcal{V}} \{g(\boldsymbol{\nu'}, \boldsymbol{y}) : G(\boldsymbol{\nu'}, \boldsymbol{y}) \geq 0\} \tag{12c}$$

A hierarchical structure can be observed where the set of decision variables of the upper level (leader) and lower level (follower) are given by $\boldsymbol{y} \in Y \subset \mathbb{R}^n$ and $\boldsymbol{\nu} \in \mathcal{V} \subset \mathbb{R}^m$ and their respective objective functions by $f : \mathcal{V} \times Y \to \mathbb{R}$ and $g : \mathcal{V} \times Y \to \mathbb{R}$. The constraint functions of the leader (resp. follower) are given by $F : \mathcal{V} \times Y \to \mathbb{R}^p$ (resp. $G : \mathcal{V} \times Y \to \mathbb{R}^q$). Decisions are sequentially made. First, the leader makes a decision on $\boldsymbol{y}$ and the follower solves the parameterised problem given in (12c).

The *optimal value function* $\psi(\boldsymbol{y}) := \min\{f(\boldsymbol{\nu}, \boldsymbol{y}) : G(\boldsymbol{\nu}, \boldsymbol{y}) \geq 0\}$ is used to re-write the BLPP as follows:

$$\min_{\boldsymbol{\nu} \in \mathcal{V}, \boldsymbol{y} \in Y} f(\boldsymbol{\nu}, \boldsymbol{y}) \tag{13a}$$

$$\text{s.t.} \quad F(\boldsymbol{\nu}, \boldsymbol{y}) \geq 0 \tag{13b}$$

$$G(\boldsymbol{\nu}, \boldsymbol{y}) \geq 0 \tag{13c}$$

$$g(\boldsymbol{\nu}, \boldsymbol{y}) \leq \psi(\boldsymbol{y}) \tag{13d}$$

The constraint expressed by inequality (13d) models that any solution with $g(\boldsymbol{\nu}, \boldsymbol{y})$ bounded by $\psi(\boldsymbol{y})$ will be a solution for the BLPP. Our approach is based on the optimal value function and the High-Point Relaxation (HPR). The HPR is given by (13a)-(13c), where the leader's objective (13a) is optimized over the shared constraint set (13b) and (13c), with the non-convex constraint (13d) dropped (see Fischetti 2018). Solution procedures to solve the HPR involve in integrating cutting planes and branch-and-bound methods.

We will now briefly discuss the other predominant technique to solve bi-level optimization problems, which involves in reformulating the problem as a single level program using strong duality when the follower's problem is convex (Burgard, Pharkya, and Maranas 2003; SunXu et al. 2013; Tepper and Shlomi 2010; Zare et al. 2019). We will discuss the details of reformulation here when

the follower's problem is a linear program as this is the structure that is relevant to our work. Suppose $\mathcal{V} := \{\boldsymbol{\nu} : \boldsymbol{\nu} \geq 0\}$ and the constraints and objective of follower's problem given in (12c) are given by the following functions:

$$g(\boldsymbol{\nu}, \boldsymbol{y}) := \boldsymbol{c}^t\boldsymbol{\nu} + \boldsymbol{d}^t\boldsymbol{y}$$
$$G(\boldsymbol{\nu}, \boldsymbol{y}) := C\boldsymbol{\nu} + D\boldsymbol{y} - \boldsymbol{b}$$

Associating dual variables $\boldsymbol{\pi}$ with the constraint functions $G(\boldsymbol{\nu}, \boldsymbol{y}) \geq 0$, the dual of the parameterized follower problem can then be written as

$$\boldsymbol{d}^t\boldsymbol{y} + \max_{\boldsymbol{\pi} \in \mathbb{R}^q_+} (\boldsymbol{b} - D\boldsymbol{y})^t\boldsymbol{\pi} \tag{14a}$$
$$C^t\boldsymbol{\pi} \leq \boldsymbol{c} \tag{14b}$$

Assuming both primal and dual problems have finite optimal objective values and using the fact that only at optimality, these solutions are equal, we can reformulate BLPP, problem (12), as follows:

$$\min_{\boldsymbol{\nu} \in \mathcal{V}, \boldsymbol{y} \in Y, \boldsymbol{\pi} \in \mathbb{R}^q_+} \quad f(\boldsymbol{\nu}, \boldsymbol{y}) \tag{15a}$$
$$\text{s.t.} \quad F(\boldsymbol{\nu}, \boldsymbol{y}) \geq 0 \tag{15b}$$
$$G(\boldsymbol{\nu}, \boldsymbol{y}) \geq 0 \tag{15c}$$
$$\boldsymbol{c}^t\boldsymbol{\nu} = (\boldsymbol{b} - D\boldsymbol{y})^t\boldsymbol{\pi} \tag{15d}$$
$$C^t\boldsymbol{\pi} \leq \boldsymbol{c} \tag{15e}$$

It is worth noting that *strong-duality* and *KKT* reformulations, using Complementary slackness conditions, are equivalent and both introduce bi-linear terms that require linearization (Schmidt and Beck 2023). Assuming each bi-linear term involves in the product of a binary and a continuous variable, it can be linearized by adding a new variable corresponding to each such bi-linear term. However, this method might become cumbersome as the model increases in size (Zare et al. 2019).

4.3 From flux balance analysis to bi-level gene deletions

In this section, we briefly discuss the basic aspects of flow (flux) estimations in metabolic engineering and how it is modeled as a bi-level program. We also describe the OptKnock model (Burgard, Pharkya, and Maranas 2003), as this is the most popular optimization-based approach among researchers and practitioners and helps motivate our methodology.

Metabolic networks have been constructed with genomics and biochemical information to capture the relationship between the chemical processes inside an organism, allowing the study of these networks regarding flow distribution (Edwards, Covert, and B. Palsson 2002; Lee et al. 2012). The most popular method to estimate the flow distribution in a single-cell organism is the FBA (Orth, Thiele, and Bernhard O Palsson 2010). We denote $M = \{1, \ldots, m\}$ to

be the set of all reactions and $N = \{1, \ldots, N\}$ to be the set of all metabolites in the metabolic network. The estimation of flow rates uses a mathematical representation of the metabolic network where the metabolic reactions are captured in S, a stoichiometric matrix of size $n \times m$. The entries of this sparse matrix are the stoichiometric coefficients of the metabolites (rows) participating in a reaction (columns). The mass flow through the reactions is denoted using the vector $\boldsymbol{\nu} \in \mathbb{R}^m$ and the individual component of the vector corresponds to a specific reaction. FBA is bound by the mass balance constraints $S\boldsymbol{\nu} = 0$. The vector $\boldsymbol{\nu}$ is also constrained by its lower and upper bounds, $\boldsymbol{\nu}_{min}$ and $\boldsymbol{\nu}_{max}$ respectively, that define the allowable flow distribution of the system. We define the feasible set for the FBA as

$$\Phi := \{\boldsymbol{\nu} : S\boldsymbol{\nu} = 0, \boldsymbol{\nu}_{\min} \leq \boldsymbol{\nu} \leq \boldsymbol{\nu}_{\max}, \}$$

The complete FBA model is then given by

$$\max_{\boldsymbol{\nu} \in \Phi} \quad \nu_b \tag{16}$$

where ν_b is the component of $\boldsymbol{\nu}$ corresponding to the biomass. The objective function (16) in this model is cellular growth rate (biomass), which is defined with respect to the organism, but from an application perspective one wishes to increase the yield of any desired chemical within the metabolic network. This requires experimentation to predict chemical production. The challenge to develop engineered single-cell organisms is an ongoing effort in the field of biotechnology for the optimal production of chemicals (Stephanopoulos 1998) and bio-fuel factories (Yadav and Saxena 2013), contributing to different industries (Volk et al. 2023; Anand, Mukherjee, and Padmanabhan 2020), and development of toolboxes such as COBRA (Heirendt et al. 2019).

The network's flow distribution can be modified to a more desirable outcome via gene knockouts. Gene deletion (knockouts) refers to restricting the flow through a particular reaction to zero. We can model this by associating a binary variable to each arc in the network (Edwards, Covert, and B. Palsson 2002). The binary variables assume a value of one if the reaction is active and a value of zero in the case the reaction is not active, thus "*knocked out*".

$$y_i = \begin{cases} 1 \ , & \text{if reaction } i \text{ is present in the metabolic network} \\ 0 \ , & \text{“knocked out”.} \end{cases}$$

For a given binary vector, $\boldsymbol{y} \in \{0, 1\}^m$, we define the parametrized FBA feasible set as

$$\Phi(\boldsymbol{y}) = \{\boldsymbol{\nu} : S\boldsymbol{\nu} = 0, \nu^{min} y_j \leq \nu_j \leq \nu^{max} y_j, \forall j \in M\} \tag{17}$$

Chemical predictions using only FBA seem to fail as the model doesn't capture the single-cell organism's evolutionary objective (Bernhard O. Palsson, Ibarra, and Edwards 2002; Raman and Chandra 2009). Generally, the organism's objective and the desired chemical production are two independent, often competing, objectives. Mathematical optimization offers frameworks to deal with such

cases. On the one hand, the work by Andrade et al. 2020, Daud et al. 2019 and Maia et al. 2008 frame gene-knockouts as a bi-objective optimization problem where the solution of the objectives, $z_1 = \max\ \nu_c$ and $z_2 = \max\ \nu_b$, are found in the Pareto frontier. On the other hand, this problem can be modelled as a bi-level programming problem, wherein the leader will be the *engineer* and the follower is the *microbe* (Smith, Prince, and Geunes 2013). The follower optimizes its objective over the network while the leader alters the network by limiting the possible reactions available. The leader makes binary decisions $\boldsymbol{y}$, a given number of reactions to suppress, denoted by K. We can then define the leader's feasible set as

$$Y = \{\boldsymbol{y} \in \{0,1\}^m : \sum_{i=1}^{m}(1 - y_i) = K\} \tag{18}$$

The bi-level gene knockout model, which was first formulated in Burgard, Pharkya, and Maranas 2003, can then be written as

$$\begin{aligned} &\max_{\boldsymbol{y} \in Y}\ \nu_c \\ &\quad \text{s.t. } \boldsymbol{\nu} \in \arg\max\{\nu_b : \boldsymbol{\nu} \in \Phi(\boldsymbol{y})\} \end{aligned} \tag{19}$$

where ν_c denotes some component in $\boldsymbol{\nu}$ corresponding to the desired chemical to optimize the leader's objective. It is one of the indices from the subset of desired chemicals, $C \subset M$, of the leader. More particularly, problem (19) has three components: (i) *leader's objective* to maximize the production rate of a determined chemical ν_c present in the metabolic network; (ii) *knapsack constraint* to select the allowed number of reactions to knock-out; and (iii) *follower's problem* which is a parameterized FBA trying to optimise the growth of the organism.

The first bi-level gene knockout model, OptKnock, assumes an optimistic bi-level approach where among the follower's possible solutions the solution that allows the best results for the leader is picked. In this paper, the solution method employes the strong duality to reformulate the bi-level problem into a single-level mixed integer problem. This method adds more linear constraints and discrete variables to the problem (Zare et al. 2019). The inclusion of the new variables results in non-linear terms in the objective function of the dual problem, though the non-linearity is ignored. SunXu et al. 2013 developed a subsequent model, called ReacKnock, where the non-linear terms are linearized using McCormick envelopes exploiting the bounds of the continuous variable and a sufficiently large number as a threshold (McCormick 1976; Asghari et al. 2022), increasing the number of constraints from the original model. The models presented in Burgard, Pharkya, and Maranas 2003; SunXu et al. 2013 only solve for optimistic gene deletions. In Tepper and Shlomi 2010, RobustKnock model was developed considering the pessimistic approach to suggest robust gene deletions. This model is approached as a max-min bi-level model and linearised using a similar technique as in SunXu et al. 2013.

4.4 Algorithm

In this section, we introduce our unified solution approach to identifying gene deletion strategies, encompassing both optimistic and pessimistic approaches. This integrated methodology allows decision-makers to explore gene knockouts with a balanced perspective, considering both ideal outcomes, where the microbial response is aligned with maximum production goals, and worst-case scenarios, where cellular growth objectives dominate, potentially reducing chemical yields. By incorporating both strategies, our approach enhances the robustness and flexibility of metabolic engineering predictions, facilitating optimized gene deletion strategies across varied biological objectives.

Our solution method iteratively solves the HPR problem (20), which is a relaxation of problem (19). In each iteration, cuts are added to restrict the search to bi-level feasible solutions (feasibility cuts) and eliminate solutions that are not optimal (optimality cuts).

$$\max_{\substack{\boldsymbol{y}\in Y \\ \boldsymbol{\nu}\in\boldsymbol{\Phi}(\boldsymbol{y})}} \quad \nu_c \tag{20}$$

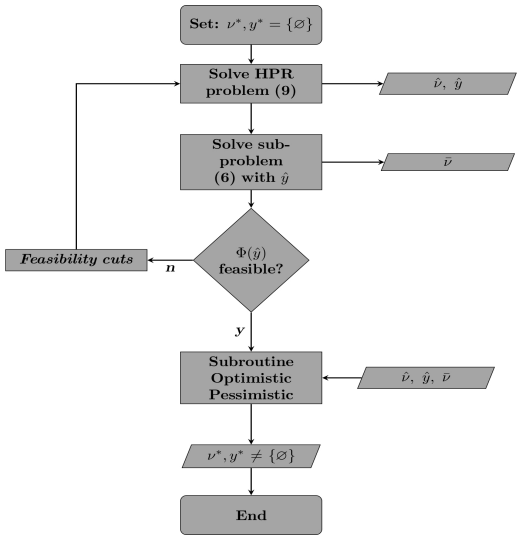


Figure 1: Framework to solve bi-level gene knockouts

The proposed algorithm to identify gene-knockout strategies is provided both in Figure 1 and Algorithm 1. The algorithm accommodates both optimistic or pessimistic strategy (depending on the choice of the decision maker), which we will discuss shortly. A primal bound, denoted by LB, is set initially as a sufficiently large negative number and will be updated as the algorithm progresses. In each iteration, we solve the current HPR (20) to optimality which returns the solution $\hat{\boldsymbol{\nu}}$ and $\hat{\boldsymbol{y}}$. We then solve the parameterized problem $FBA(\alpha, \hat{\boldsymbol{y}})$

$$\min M\nu_b + \alpha\nu_c : \boldsymbol{\nu} \in \Phi(\hat{\boldsymbol{y}}) \tag{21}$$

where α is chosen depending on the decision maker's strategy. The choice of M, the objective coefficient of ν_b, will be explained shortly. If the parameterized problem is infeasible or current incumbent solution is better than the FBA's solution corresponding to the chemical value for the suggested knockout set $\hat{\boldsymbol{y}}$, we then update the HPR by adding the cut

$$\sum_{i \in \mathbb{I}(\hat{\boldsymbol{y}})} y_i \geq 1, \tag{22}$$

where $\mathbb{I}(\hat{\boldsymbol{y}}) := \{j : \hat{y}_j = 0, \forall j \in M\}$. This will restrict the solution $\hat{\boldsymbol{y}}$ from being generated in subsequent iterations. Otherwise, the parametrised problem will return a solution $\hat{\boldsymbol{\nu}}$.

Algorithm 1 Bi-level Algorithm Framework

Input: gap, α ▷ $\alpha = 1$ is optimistic and $\alpha = -1$ is pessimistic
Initialise: $LB = -\infty$, $UB = \infty$
while $\frac{UB-LB}{LB} \leq gap$ **do**
    $(\hat{\boldsymbol{\nu}}, \hat{\boldsymbol{y}}) \leftarrow HPR(\boldsymbol{\nu}, \boldsymbol{y})$
    $UB = \hat{\nu}_c$
    $\bar{\boldsymbol{\nu}} \leftarrow FBA(\alpha, \hat{\boldsymbol{y}})$
    if $\Phi(\hat{\boldsymbol{y}}) = \emptyset$ or $LB \geq \bar{\nu}_c$ **then**
        Add cut $(\sum_{j \in \mathbb{I}(\hat{\boldsymbol{y}})} y_j \geq 1)$ to HPR
        ▷ Either $\hat{\boldsymbol{y}}$ infeasible or incumbent is better
    else if $\bar{\nu}_b > \hat{\nu}_b$ **then**
        Add cut $(\bar{\nu}_b \leq \nu_b + \bar{\nu}_b(\sum_{j \in \mathbb{I}(\hat{\boldsymbol{y}})} y_j))$ to HPR
        ▷ $(\hat{\boldsymbol{\nu}}, \hat{\boldsymbol{y}})$ is not optimal to FBA
    else if $\alpha = -1$ and $\hat{\nu}_c > \bar{\nu}_c$ **then**
        Add cut $(\bar{\nu}_c \geq \nu_c - \bar{\nu}_c(\sum_{j \in \mathbb{I}(\hat{\boldsymbol{y}})} y_j))$ to HPR
        ▷ Pessimistic cut to lower bound ν_c
    else
        $LB = \bar{\nu}_c$
        Set $(\bar{\boldsymbol{\nu}}, \hat{\boldsymbol{y}})$ as incumbent
    end if
end while

An optimistic approach is based on the assumption that among the alternative optimal solutions to the follower's problem, the follower will choose the solution that is most beneficial for the leader with respect to the chemical mass flow in the network. This subroutine sets objective function of the parametrised problem (21) to max $M\nu_b + \nu_c$, so the follower's problem is optimised on ν_b and among the alternate optimal solutions, the one with the maximum value ν_c is chosen. It is essential to select the value of M carefully. This parameter must be sufficiently large to ensure that the optimisation is performed on biomass ν_b rather than on ν_v. However, M should not be excessively large, as this would result in $\frac{1}{M}$ exceeding the numerical tolerance of the algorithm. Identifying an

appropriate value for M is generally straightforward, as numerical tolerance is typically set to a (very) small value.

A pessimistic approach is based on the assumption that the follower's response to the leader's strategy is going to be detrimental to the leader's solution. In other words, among the alternative optimal solutions, the follower will choose the solution that has the least value for ν_c resulting in the worst scenario for the leader. The objective function in the parametrized problem (21) is set to max $M\nu_b - \nu_c$. The optimisation is first performed on ν_b and then for the worst outcomes with respect to ν_c.

The algorithm generates cuts according to the following specific conditions. When we add in the cut $\sum_{j\in\mathbb{I}(\hat{\boldsymbol{y}})} y_j \geq 1$, in both pessimistic and optimistic approaches, we are simply discarding the suggested knockout $\hat{y}$ from the HPR problem, either because the solution leads to worse lower-level outcome or the lower-lever becomes infeasible. We add the cut

$$\bar{\nu}_b \leq \nu_b + \bar{\nu}_b(\sum_{j\in\mathbb{I}(\hat{\boldsymbol{y}})} y_j) \tag{23}$$

for both approaches, when the biomass flow value in the follower's optimal solution ($\bar{\nu}_b$) is greater than the biomass value in the HPR solution ($\hat{\nu}_b$). In the pessimistic approach, we add

$$\bar{\nu}_c \geq \nu_c - \bar{\nu}_c(\sum_{j\in\mathbb{I}(\hat{\boldsymbol{y}})} y_j) \tag{24}$$

to restrict the value of ν_c being higher than the worst case chemical production value obtained in the follower's optimal solution. If no cuts are added, then we have obtained a better incumbent. We update the bounds in this case and resolve the HPR. We now show the correctness of the above algorithm in the following theorem.

Theorem 1. *Algorithm 1 terminates at the optimal solution to* (19) *for both optimistic and pessimistic cases.*

Proof. We focus on the optimistic case as the pessimistic case follows similar lines of reasoning. First, notice that the solution of Algorithm 1 is feasible to (19) as cuts of type (23) will only accept optimal solutions to FBA. Furthermore, notice that the solution vector of the HPR, i.e. $(\hat{\boldsymbol{\nu}}, \hat{\mathbf{y}})$, changes at least in one component compared with the solutions in all previous iterations. Since there are only finitely many of these, the algorithm will terminate. For an optimal solution $(\boldsymbol{\nu}^*, \mathbf{y}^*)$ to (19), it is enough if we focus on cuts generated with respect to $(\tilde{\boldsymbol{\nu}}, \mathbf{y}^*)$, for any $\tilde{\boldsymbol{\nu}}$. Cuts of type (22) are added only when current primal solution is better than ν_c^* as $\Phi(\mathbf{y}^*) \neq \emptyset$. This solution is pruned from further search only if there exist some incumbent solution (that is bi-level feasible) with the optimal value greater than ν_c^*, unless the incumbent is already optimal but corresponding to some other knockout strategy. □

4.5 Computational experiments

The algorithms described in the previous section were implemented in Python 3.10 and used to solve some instances using the Gurobi 10 solver on a personal computer with an Intel® Core™ i7 processor running at 1.3 GHz to 1.5 GHz. They were tested on three established E. coli metabolic networks: iJO1366, iJR904, and iAF1260. The key criteria for comparing our algorithms with the benchmark include the solving time (latency) for each method and the validity of the solutions in terms of chemical production. The number of allowed knockouts for the experiments was $K = 1, 2, 3$. Moreover, we test our solving methods under biological assumptions for each metabolic network.

4.5.1 Biological assumptions

The biological assumptions activate the metabolic network for its growth by setting specific flows on key reactions. The biological assumptions for the E. coli metabolic network are set similarly through the different metabolic networks used for the computations, as outlined in Table 2. The benchmarks were taken from (Tepper and Shlomi 2010) and (Mendoza 2018) for the pessimistic and optimistic algorithms respectively. The growth in each strain is allowed through glucose uptake. A minimum growth should be set before computing. In most cases this is set to expect growth of at least 50% of the wild strain, this is $\nu_b^{min} = 50\% WT_b$ or $\nu_b = \nu_b^{tgt}$. Wild strain refers to the unmodified network's flow estimation. However, each metabolic network is different, and special adjustments have to be made on each metabolic network employing the solving methods.

4.5.2 Assumptions

- Allowing glucose uptake.
- Unconstraining uptake routes for inorganic phosphate, sulfate, and ammonia.
- Enabling secretion routes for acetate, carbon dioxide, ethanol, formate, lactate and succinate.
- Constraining the phosphotransferase system.
- Constraining the O_2 (oxygen) uptake.

The number of reactions translates to the number of continuous variables in the follower's problem and the number of binary variables in the master's problem. The KO size is a fraction of potential reactions for gene deletion, the smaller the KO size the less computing is expected for the solving methods. Finally, the algorithms were tested under their biological assumptions, different K values, and different growth percentages. The characteristics of the E. coli metabolic networks utilized in the computational experiment are summarized in Table 3, where S stands for the stoichiometric matrix.

Rxn	ν^{min}	ν^{max}	Strain
ATPM	8.39	8.39	iAF1260
O2	-18.5	1e4	iAF1260
glc	-10	-10	iAF1260
ATPM	7.6	7.6	iJR904
O2	-20	1e4	iJR904
glc	-10	-10	iJR904
ATPM	3.5	1e4	iJO1366
O2	0	1e4	iJO1366
glc	-10	-10	iJO1366

Table 2: Biological assumptions, all rates unit is $mmol/g(Dw)h$

Model	S size		KO set		Bi-level Objective	Bacteria
	Metabolites	Reactions	Size	%		
iJO1366	1805	2583	37	1.43	Succinate	E. coli
iJR904	761	1075	32	2.98	Acetate	E. Coli
iAF1260	1668	2382	37	1.55	Succinate	E.Coli

Table 3: Metabolic network characteristics and KO proportions

4.6 Results

The performance of our algorithms is compared against benchmarks that employ the MILP reformulation, with a focus on solving time and growth outcomes. A good solution is defined as one where the knockout strategy is computed within a reasonable time. The solving time is extracted from the solver's information of computing time as wall-clock times.

4.6.1 Validation

To validate our optimistic algorithm we compared the results computed with our method and the chemical production from the MILP benchmark (Mendoza 2018). Table 4 shows the biomass flux (ν_b), chemical flux (ν_c), and knockouts produced by the various approaches for different levels of minimum growth. Our method computes same values, for biomass and chemical production, as the values from the MILP benchmark across varying percentages of minimum biomass production (*mbp*). However, our method returns a different knockout strategy when $mbp >= 80\%$, this is because of the follower's problem structure where different strategies (flow distribution) could lead to the same objectives.

4.6.2 Performance

Following algorithm validation, we assess the performance of each method in terms of solution time, measured in seconds, and presented in Table 5. We compare the performance (solution time) between the optimistic benchmark

	Benchmark			Optimistic Algorithm		
mbp	ν_b	ν_c	Knockouts	ν_b	ν_c	Knockouts
10%	0.0531	8.9797	GLCptspp,PYK	0.0531	8.9797	GLCptspp,PYK
20%	0.0531	9.9797	GLCptspp,PYK	0.0531	8.9797	GLCptspp,PYK
30%	0.1173	2.3132	PFL,RPI	0.1173	2.3132	PFL,RPI
40%	0.1173	2.3132	PFL,RPI	0.1173	2.3132	PFL,RPI
50%	0.1831	1.4148	PFL,TKT2	0.1831	1.4148	PFL,TKT2
60%	0.1831	1.4148	PFL,TKT2	0.1831	1.4148	PFL,TKT2
70%	0.1831	1.4148	PFL,TKT2	0.1831	1.4148	PFL,TKT2
80%	0.2401	0.2504	FUM,MDH	0.2401	0.2504	FUM,GND
90%	0.2401	0.2504	FUM,SUCOAS	0.2401	0.2504	FUM,G6PDH2r

Table 4: IJO1366 with double knockouts

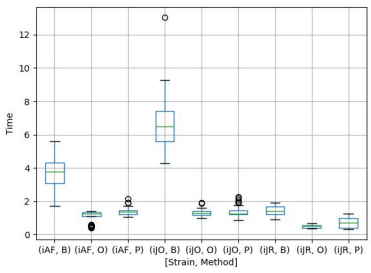
(B) and our optimistic algorithm (O). We could not compare the solution time for the pessimistic benchmark as this information is not available. However, a comparison of our pessimistic algorithm (P) against the optimistic algorithms would help one to assess its performance.

	Max			Mean			Min		
mbp	B	O	P	B	O	P	B	O	P
10%	28.265	10.473	6.633	8.4565	3.4337	2.6868	0.955	0.383	0.380
20%	23.559	9.921	6.625	8.8517	2.8867	2.6355	0.946	0.451	0.400
30%	29.639	10.083	6.615	9.8855	2.8805	2.6533	1.137	0.411	0.378
40%	15.594	9.684	7.728	7.0123	3.8932	2.7815	1.707	0.377	0.364
50%	18.432	9.535	6.008	7.0322	3.1038	2.5038	0.995	0.369	0.347
60%	18.345	8.402	8.102	8.2117	2.7713	2.7485	1.234	0.377	0.343
70%	12.301	7.187	5.692	6.5250	2.8377	2.1925	1.169	0.415	0.382
80%	13.292	5.632	4.721	6.0622	2.4868	1.8395	1.406	0.363	0.351
90%	21.059	4.886	4.502	7.9472	2.0978	1.5968	1.040	0.378	0.395

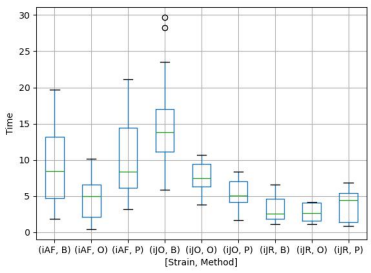
Table 5: Average solving time (seconds) for $K = 2$

In Figure 2, we present the box plot of solution time by strain and solution method for various chemicals and target levels. We have a noticeable improvement in the solution time our framework. Our optimistic solving time is consistently outperforms the single level reformulation for all values of k. Our pessimistic performance is competent for low values of k and deteriorates when $k = 3$. This is an expected behaviour for the pessimistic approach.

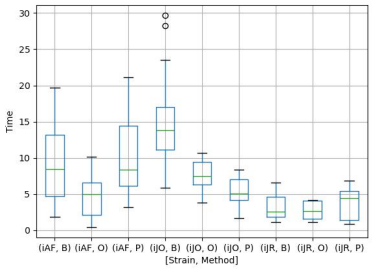
We also present the solution time target levels and k values in Figure 3. The vertical bars represent the variation across the three strains and chemical targets. We make a similar observation here. The optimistic approach either outperforms or is as good as the benchmark for all values of k. The pessimistic approach works well for low values of k and deteriorates when $k = 3$.



(a) $k = 1$



(b) $k = 2$



(c) $k = 3$

Figure 2: Time comparison by method and strain for various target levels and chemical targets.

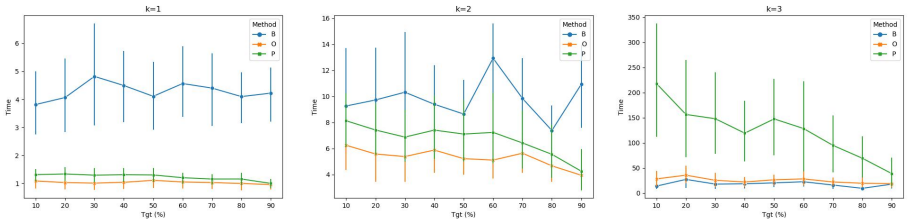


Figure 3: Overall Performance of the three methods for various k and biomass targets. The bars indicate the variation corresponding to the three strains and various chemical targets.

4.6.3 Pessimistic vs optimistic

In general, the chemical production computed under the pessimistic approach tends to be smaller or in the best scenario equal to the chemical values computed using the optimistic approach. We present the ratio of optimal pessimistic and optimistic solutions for various chemicals, k and strains in 4. There is generally an increasing pattern seen by target levels for all chemicals for iAF strain and it tends to stay the same for iJO strain. There are no such clear patterns with the iJR strain.

In essence, the optimistic approach will give a theoretical maximum that can be achieved whereas the pessimistic approach will provide a robust solution where the risk of selecting a strategy that delivers the worst outcome is minimized and the solution is not only bi-level feasible but also microbial feasible.

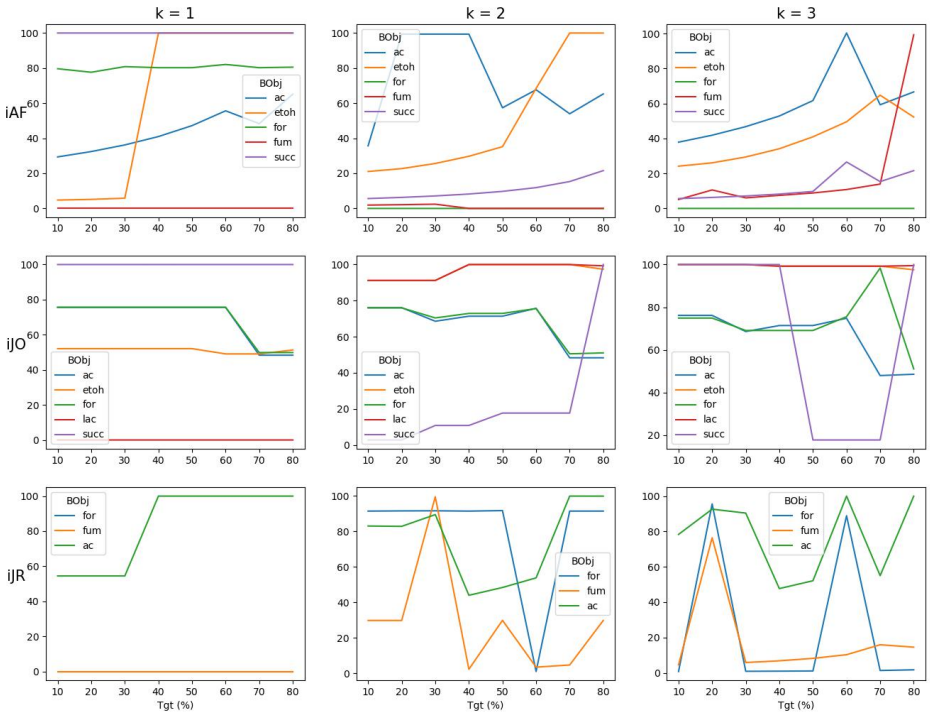


Figure 4: Ratio of pessimistic to optimistic (in percentage) chemical production

4.7 Conclusion

The framework we provide permits the user to pick either an optimistic or pessimistic solution and we have demonstrated that it consistently outperforms the benchmark (see Figure 2 and Table 4). Our algorithm in contrast with the single-level reformulations do not deal with non-linear terms and extra dual variables. It is a branch-and-cut algorithm where the optimality and feasibility cuts are added to eliminate bilevel infeasible and suboptimal solutions from the search space.

While there is no direct performance comparison with the pessimistic algorithm and its benchmark. We compared our optimistic and pessimistic approach as in our methodology these algorithms share similarities. These algorithms have a similar performance (see Figure 3). However, the performance of the pessimistic approach deteriorates more in comparison as the number of allowable knockouts increases. The optimistic and pessimistic algorithms when $K = 1$ outperform the optimistic single-level reformulation by more than 70% in running time. When $K = 2$ the algorithms' performance is improved by at least 50%. However, when K increases the pessimistic approach explodes in its solving time due to the additional steps this procedure goes through (see Algorithm 1), and the combinatorial nature of finding the components in both subroutines.

5 Final Remarks

Our exploration of bi-level programming underscores its significance and versatility in tackling complex hierarchical decision-making problems across various domains such as network optimization and metabolic engineering. Bi-level programming's inherent NP-hardness challenges researchers to develop innovative and computationally efficient methods to render these problems tractable. Throughout our work, we have demonstrated that solution techniques, including reformulating bi-level problems into single-level problems via strong duality and employing cutting-edge methodologies like branch-and-bound and cutting planes, play a pivotal role in optimizing decision-making processes when nested decisions are involved.

Our specific application of bi-level modeling in metabolic engineering highlights the promise and potential of these frameworks in biotechnology. By emulating the interaction between an engineer and a microbe in modifying metabolic networks through gene knockouts, we provide a methodology not only to maximize chemical production but also to align biological and engineering objectives in a coherent, structured manner.

Moreover, through the development of algorithms adopting both optimistic and pessimistic strategies, this study provides pathways towards achieving maximum feasible solutions and robustness in decision-making processes. Our computational results exhibit not only improvements in solution times but also highlight the flexibility and applicability of these models to predictively simulate and

optimize biological systems.

Ultimately, bi-level programming represents a profound tool for researchers and practitioners seeking to disentangle and optimize complex interdependent systems characterized by hierarchy and sequential decision-making. As advancements in computational methods continue to evolve, so too will the efficacy and applicability of bi-level programming across an ever-expanding array of scientific and industrial domains. Future research and development in this area promise to unlock even greater capabilities, supporting more efficient and impactful decision-making in systems governed by hierarchical structure and interdependent objectives.

References

Ajay, Malaviya, C. Sharkey Thomas, and Rainwater Chase (2010). "Multi-Period Network Interdiction Models with Applications to City-Level Drug Enforcement". In: *IIE Annual Conference. Proceedings*, p. 1.

Alizadeh, S. M., P. Marcotte, and G. Savard (2013). "Two-stage stochastic bilevel programming over a transportation network". In: *Transportation Research Part B: Methodological* 58, p. 92. ISSN: 0191-2615.

Anand, Shreya, Koel Mukherjee, and Padmini Padmanabhan (2020). "An insight to flux-balance analysis for biochemical networks". In: *Biotechnol Genet Eng Rev* 36.1, pp. 32–55. ISSN: 0264-8725. DOI: 10.1080/02648725.2020.1847440.

Andrade, Ricardo et al. (2020). "MOMO - multi-objective metabolic mixed integer optimization: application to yeast strain engineering". In: *BMC bioinformatics*.

Apaydin, Meltem et al. (2017). "Robust mutant strain design by pessimistic optimization". In: *BMC Genomics* 18.Suppl 6, pp. 677–677. ISSN: 1471-2164. DOI: 10.1186/s12864-017-4025-7.

Asghari, Mohammad et al. (Jan. 2022). "Transformation and Linearization Techniques in Optimization: A State-of-the-Art Survey". In: *Mathematics* 10, p. 283. DOI: 10.3390/math10020283.

Balas, Egon and Robert G. Jeroslow (1980). "Strengthening cuts for mixed integer programs". In: *EUR J OPER RES* 4.4, pp. 224–234. ISSN: 0377-2217. DOI: 10.1016/0377-2217(80)90106-X.

Bard, J. F. (1991). "Some properties of the bilevel programming problem". In: *Journal of optimization theory and applications* 68.2, pp. 371–378. ISSN: 0022-3239. DOI: 10.1007BF00941574.

Bracken, Jerome and James T. McGill (1973). "Mathematical Programs with Optimization Problems in the Constraints". In: *OPER RES* 21.1, pp. 37–44. ISSN: 0030-364X. DOI: 10.1287/opre.21.1.37.

Brochado, Ana Rita and Kiran Raosaheb Patil (2014). "Model-Guided Identification of Gene Deletion Targets for Metabolic Engineering in Saccharomyces cerevisiae". In: *Yeast Metabolic Engineering: Methods and Protocols*. Ed. by Valeria Mapelli. New York, NY: Springer New York, pp. 281–294. ISBN: 978-1-4939-0563-8. DOI: 10.1007/978-1-4939-0563-8_17. URL: https://doi.org/10.1007/978-1-4939-0563-8_17.

Burgard, Anthony P., Priti Pharkya, and Costas D. Maranas (2003). "Optknock: A bilevel programming framework for identifying gene knockout strategies for microbial strain optimization". In: *Biotechnology and Bioengineering* 84.6, pp. 647–657. ISSN: 0006-3592. DOI: 10.1002/bit.10803.

Caprara, A. et al. (2013). "A complexity and approximability study of the bilevel knapsack problem". In: vol. 7801. Berlin, Heidelberg: Berlin, Heidelberg: Springer, pp. 98–109. DOI: 10.1007/978-3-642-36694-9_9.

Colson, Benoít, Patrice Marcotte, and Gilles Savard (2005). "Bilevel programming: A survey". In: *4OR* 3.2, pp. 87–107. ISSN: 16194500. DOI: 10.1007/s10288-005-0071-0.

Colson, Benoît, Patrice Marcotte, and Gilles Savard (2007). "An overview of bilevel optimization". In: *Annals of Operations Research* 153.1, p. 235. ISSN: 02545330. DOI: 10.1007/s10479-007-0176-2.

Daud, Kauthar Mohd et al. (2019). "A non-dominated sorting Differential Search Algorithm Flux Balance Analysis (ndsDSAFBA) for in silico multiobjective optimization in identifying reactions knockout". In: *Comput Biol Med* 113, pp. 103390–103390. ISSN: 0010-4825 1879-0534. DOI: 10.1016/j.compbiomed.2019.103390.

Delgadillo, Andres, Jose Manuel Arroyo, and Natalia Alguacil (2010). "Analysis of Electric Grid Interdiction With Line Switching". In: *TPWRS* 25.2, pp. 633–641. ISSN: 0885-8950. DOI: 10.1109/TPWRS.2009.2032232.

Dempe, Stephan (2003). "Annotated Bibliography on Bilevel Programming and Mathematical Programs with Equilibrium Constraints". In: *Optimization* 52.3, pp. 333–359. ISSN: 0233-1934. DOI: 10.1080/0233193031000149894.

— (2015). *Bilevel programming problems: theory, algorithms, and applications to energy networks*. 1st ed. 2015.. New York: Springer.

Dempe, Stephan et al. (2011). "Natural gas bilevel cash-out problem: Convergence of a penalty function method". In: *European journal of operational research* 215.3, pp. 532–538. ISSN: 0377-2217. DOI: 10.1016/j.ejor.2011.07.003.

Denegre, S. T. and T. K. Ralphs (2009). *A Branch-and-cut Algorithm for Integer Bilevel Linear Programs*. Vol. 47. Boston, MA: Boston, MA: Springer US, pp. 65–78. DOI: 10.1007/978-0-387-88843-9_4.

DeNegre, Scott (2011). "Interdiction and Discrete Bilevel Linear Programming". PhD thesis. Lehigh University.

Deng, Xiaotie (1998). "Complexity Issues in Bilevel Linear Programming". In: *Multilevel Optimization: Algorithms and Applications*. Vol. 20. Nonconvex Optimization and Its Applications. Dordrecht: Kluwer Academic, pp. 149–164.

Dokka, Trivikram et al. (Aug. 2016). "Pricing toll roads under uncertainty". In.

Edwards, Jeremy S., Markus Covert, and Bernhard Palsson (2002). "Metabolic modelling of microbes: the flux[U+2010]balance approach". In: *Environ Microbiol* 4.3, pp. 133–140. ISSN: 1462-2912. DOI: 10.1046/j.1462-2920.2002.00282.x.

Fischetti, Matteo (2018). "On the use of intersection cuts for bilevel optimization". In: *Mathematical Programming* 172.1/2, pp. 77–104. ISSN: 00255610. DOI: 10.1007/s10107-017-1189-5.

Gilbert, Francois, Patrice Marcotte, and Gilles Savard (2015). "A numerical study of the logit network pricing problem". In: *Transportation Science* 49.3, p. 706. ISSN: 0041-1655. DOI: 10.1287/trsc.2014.0560.

Griffiths, Anthony J. F. (2020). *Introduction to genetic analysis*. NY: Mcmillan Learning.

He, et al. (2017). "Optimal Time-Varying Pricing for Toll Roads Under Multiple Objectives". In: *Transportation Science* 51, pp. 412–426.

Hearn, D. W. et al. (2001). "Computational methods for congestion toll pricing models". In: IEEE, pp. 257–262. DOI: 10.1109/ITSC.2001.948665.

Heirendt, Laurent et al. (2019). "Creation and analysis of biochemical constraint-based models using the COBRA Toolbox v.3.0". In: *NAT PROTOC* 14.3, pp. 639–702. ISSN: 1754-2189 1750-2799. DOI: 10.1038/s41596-018-0098-2.

Jeroslow, Robert G. (1985). "The polynomial hierarchy and a simple model for competitive analysis". In: *Mathematical programming* 32.2, pp. 146–164. ISSN: 0025-5610. DOI: 10.1007BF01586088.

Kim, Joonhoon and Jennifer L. Reed (2010). "OptORF: Optimal metabolic and regulatory perturbations for metabolic engineering of microbial strains". In: *BMC Syst Biol* 4.1, pp. 53–53. ISSN: 1752-0509. DOI: 10.1186/1752-0509-4-53.

Labbé, Martine, Patrice Marcotte, and Gilles Savard (1998a). "A Bilevel Model of Taxation and Its Application to Optimal Highway Pricing". In: *Management Science* 44.12. ISSN: 00251909. DOI: 10.1287mnsc.44.12.1608.

— (1998b). "A Bilevel Model of Taxation and Its Application to Optimal Highway Pricing". In: *Management Science* 44.12, pp. 1608–1622. ISSN: 00251909. DOI: 10.1287/mnsc.44.12.1608.

Lee, Sang Yup et al. (2012). "Genome-Scale Network Modeling". In: *Systems Metabolic Engineering*. Ed. by Christoph Wittmann and Sang Yup Lee. Dordrecht: Springer. Chap. 1, pp. 1–23. ISBN: 978-94-007-4534-6. DOI: https://doi-org.proxy.lib.strath.ac.uk/10.1007/978-94-007-4534-6_1.

Maia, P. et al. (2008). "Evaluating evolutionary multiobjective algorithms for the in silico optimization of mutant strains". In: *BIBE* 8, pp. 1–6. DOI: 10.1109/BIBE.2008.4696733.

Marcotte, P. (1986). "Network design problem with congestion effects: A case of bilevel programming". In: *Mathematical Programming* 34.2, pp. 142–162. ISSN: 0025-5610. DOI: 10.1007BF01580580.

McCormick, Garth P. (1976). "Computability of global solutions to factorable nonconvex programs: Part I - Convex underestimating problems". In: *Mathematical programming* 10.1, pp. 147–175. ISSN: 0025-5610. DOI: 10.1007/BF01580665.

Mendoza, Sebastian (2018). *OptKnock COBRA Tutorial*. en. https://opencobra.github.io/cobratoolbox/latest/tutorials/tutorialOptKnock.html.

Merkert, Maximilian, Galina Orlinskaya, and Dieter Weninger (2022). "An exact projection-based algorithm for bilevel mixed-integer problems with nonlinearities". In: *J Glob Optim* 84.3, pp. 607–650. ISSN: 0925-5001. DOI: 10.1007/s10898-022-01172-w.

Moore, James T. and Jonathan F. Bard (1990). "The Mixed Integer Linear Bilevel Programming Problem". In: *Operations Research* 38.5, pp. 911–921. ISSN: 0030364X. DOI: 10.1287/opre.38.5.911.

Morton, David P., Feng Pan, and Kevin J. Saeger (2007). "Models for nuclear smuggling interdiction". In: *IIE transactions* 39.1, pp. 3–14. ISSN: 0740-817X. DOI: 10.1080/07408170500488956.

Na D, Park J.H et al. (2012). "Systems Metabolic Engineering". In: ed. by Christoph Wittmann. Springer, pp. 117–149. ISBN: 978-94-007-4534-6.

Orth, Jeffrey D., Tom M. Conrad, et al. (2011). "A comprehensive genome [U+2010] scale reconstruction of Escherichia coli metabolism—2011". In: *Mol Syst Biol* 7.1, 535–n/a. ISSN: 1744-4292. DOI: 10.1038/msb.2011.65.

Orth, Jeffrey D., Ines Thiele, and Bernhard O Palsson (2010). "What is flux balance analysis?" In: *Nature biotechnology* 28.3, pp. 245–248. ISSN: 10870156. DOI: 10.1038/nbt.1614.

Palsson, Bernhard O et al. (2007). "A genome-scale metabolic reconstruction for Escherichia coli K-12 MG1655 that accounts for 1260 ORFs and thermodynamic information". In: *Mol Syst Biol* 3.1. ISSN: 1744-4292. DOI: 10.1038/msb4100155.

Palsson, Bernhard O., Rafael U. Ibarra, and Jeremy S. Edwards (2002). "Escherichia coli K-12 undergoes adaptive evolution to achieve in silico predicted optimal growth". In: *Nature* 420.6912, pp. 186–189. ISSN: 0028-0836. DOI: 10.1038/nature01149.

Patil, Kiran Raosaheb et al. (2005). "Evolutionary programming as a platform for in silico metabolic engineering". In: *BMC Bioinformatics* 6.1, pp. 308–308. ISSN: 1471-2105. DOI: 10.1186/1471-2105-6-308.

Pharkya, Priti, Anthony P. Burgard, and Costas D. Maranas (2004). "OptStrain: A computational framework for redesign of microbial production systems". In: *Genome Res* 14.11, pp. 2367–2376. ISSN: 1088-9051. DOI: 10.1101/gr.2872004.

Pharkya, Priti and Costas D. Maranas (2006). "An optimization framework for identifying reaction activation/inhibition or elimination candidates for overproduction in microbial systems". In: *Metab Eng* 8.1, pp. 1–13. ISSN: 1096-7176. DOI: 10.1016/j.ymben.2005.08.003.

Pineda, S., H. Bylling, and J. M. Morales (2017). "Efficiently solving linear bilevel programming problems using off-the-shelf optimization software". In: *Optim Eng* 19.1, pp. 187–211. ISSN: 1389-4420. DOI: 10.1007/s11081-017-9369-y.

Raid, Mansi et al. (2012). "An Exact Algorithm for Bilevel 0-1 Knapsack Problems". In: *Mathematical problems in engineering* 2012.2012. ISSN: 1024-123X. DOI: 10.1155/2012/504713.

Raman, Karthik and Nagasuma Chandra (2009). "Flux balance analysis of biological systems: applications and challenges". In: *Brief Bioinform* 10.4, pp. 435–449. ISSN: 1467-5463. DOI: 10.1093/bib/bbp011.

Reed, Jennifer L. et al. (2003). "An expanded genome-scale model of Escherichia coli K-12 (iJR904 GSM/GPR)". In: *Genome Biol* 4.9, R54–R54. ISSN: 1474-760X. DOI: 10.1186/gb-2003-4-9-r54.

Rosales-Calderon, Oscar and Valdeir Arantes (2019). "A review on commercial-scale high-value products that can be produced alongside cellulosic ethanol". In: *Biotechnology for biofuels* 12.1, pp. 240–240. ISSN: 1754-6834. DOI: 10.1186/s13068-019-1529-1.

Schmidt, Martin and Yasmine Beck (2023). *A Gentle and Incomplete Introduction to Bilevel Optimization*. Optimization Online, TRR154 Preprint Server. Lecture Notes.

Shi, Chenggen, Guangquan Zhang, and Jie Lu (2005). "On the definition of linear bilevel programming solution". In: *Applied mathematics and computation* 160.1, pp. 169–176. ISSN: 0096-3003. DOI: 10.1016j.amc.2003.10.031.

Smith, J. Cole, Mike Prince, and Joseph Geunes (2013). "Modern Network Interdiction Problems and Algorithms". In: vol. 3-5. New York, NY: New York, NY: Springer New York, pp. 1949–1987. DOI: 10.1007/978-1-4419-7997-1_61.

Stackelberg, Heinrich von (2011). *Market structure and equilibrium*. 1st 2011. Berlin, Springer.

Stephanopoulos, G. (1998). *Metabolic engineering [internet resource]: principles and methodologies*.

SunXu, Zixiang et al. (2013). "ReacKnock: Identifying reaction deletion strategies for microbial strain optimization based on genome-scale metabolic network". In: *PLoS One* 8.12, e72150–e72150. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0072150.

Tamura, Takeyuki et al. (2018). "Computing Minimum Reaction Modifications in a Boolean Metabolic Network". In: *TCBB* 15.6, pp. 1853–1862. ISSN: 1545-5963 1557-9964. DOI: 10.1109/TCBB.2017.2777456.

Tepper, Naama and Tomer Shlomi (2010). "Predicting metabolic engineering knockout strategies for chemical production: accounting for competing pathways". In: *Bioinformatics* 26.4, pp. 536–543. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btp704.

Vital-Lopez, Francisco G., Vesna Memišević, and Bhaskar Dutta (2012). "Tutorial on biological networks". In: *WIREs Data Mining Knowl Discov* 2.4, pp. 298–325. ISSN: 1942-4787. DOI: 10.1002/widm.1061.

Volk, Michael J. et al. (2023). "Metabolic Engineering: Methodologies and Applications". In: *Chem. Rev* 123.9, pp. 5521–5570. ISSN: 0009-26651520-6890. DOI: 10.1021/acs.chemrev.2c00403.

Yadav, Sweta and Rajendra Kumar Saxena (2013). "Conversion of glycerol into biobutanol by Clostridium acetobutylicum : turning bacteria into biofuel factories". In: *Current opinion in biotechnology* 24, S43–S43. ISSN: 0958-1669. DOI: 10.1016/j.copbio.2013.05.094.

Zare, M. Hosein et al. (2019). "A note on linearized reformulations for a class of bilevel linear integer problems.(Advances in Theoretical and Applied Combinatorial Optimization)(Report)". In: *Annals of Operations Research* 272.1-2, p. 99. ISSN: 0254-5330. DOI: 10.1007s10479-017-2694-x.

Zhang, Hao, Qian Zhang, and Wenhao Chen (2019). "Bi-level programming model of truck congestion pricing at container terminals.(Technical report)(Author abstract)". In: 10.1, p. 385. ISSN: 1868-5137. DOI: 10.1007/s12652-017-0641-y.

Zhang, Xiaoning and Bert van Wee (2012). "Enhancing transportation network capacity by congestion pricing with simultaneous toll location and toll level optimization". In: *Engineering Optimization* 44.4, pp. 477–488. ISSN: 0305-215X. DOI: 10.1080/0305215X.2011.584534.

Zomorrodi, Ali R. et al. (2012). "Mathematical optimization applications in metabolic networks". In: *Metabolic Engineering* 14.6, pp. 672–686. ISSN: 1096-7176. DOI: 10.1016/j.ymben.2012.09.005.