Models and Solution Techniques for Production Planning Problems with Increasing Byproducts

Srikrishna Sridhar Jeffrey Linderoth James Luedtke

January 6, 2014

Abstract

We consider a production planning problem where the production process creates a mixture of desirable products and undesirable byproducts. In this production process, at any point in time the fraction of the mixture that is an undesirable byproduct increases monotonically as a function of the cumulative mixture production up to that time. The mathematical formulation of this continuous-time problem is nonconvex.

We present a discrete-time mixed-integer nonlinear programming (MINLP) formulation that exploits the increasing nature of the byproduct ratio function. We demonstrate that this new formulation is more accurate than a previously proposed MINLP formulation. We describe three different mixed-integer linear programming (MILP) approximation and relaxation models of this nonconvex MINLP, and derive modifications that strengthen the linear programming relaxations of these models. We also introduce nonlinear programming formulations to choose piecewise-linear approximations and relaxations of multiple functions that share the same domain and use the same set of break points in the domain.

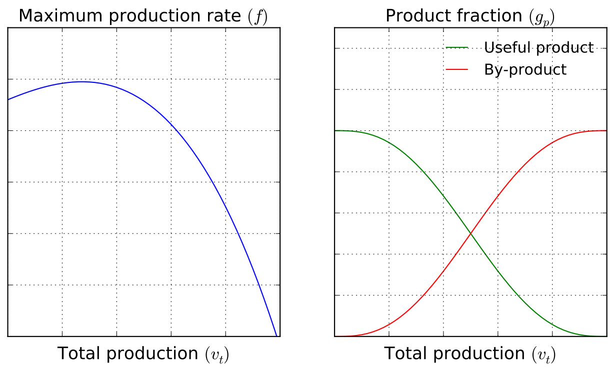
We conclude with computational experiments that demonstrate that the proposed formulation is more accurate than the previous formulation, and that the strengthened MILP approximation and relaxation models can be used to obtain provably near-optimal solutions for large instances of this nonconvex MINLP. Experiments also illustrate the quality of the piecewise-linear approximations produced by our nonlinear programming formulations.

1 Introduction

In this paper, we study a production planning problem in which the production process creates a mixture of products $\mathcal{P} = \mathcal{P}^+ \cap \mathcal{P}^-$, where the products $p \in \mathcal{P}^+$ are useful and the products $p \in \mathcal{P}^-$ are undesirable byproducts. As more of the mixture is produced, the fraction of the mixture that is a useful product decreases monotonically. Conversely, the fraction of each byproduct increases monotonically as a function of cumulative mixture production.

Production planning problems with these characteristics arise in engineering applications like the extraction of natural resources such as oil and gas [12, 14, 22, 24, 25] from fields where a pressure differential determines the rate at which the resources are extracted. This pressure difference drops as more of the resources are extracted which reduces the maximum rate at which more resources can be extracted. In these applications, the purity of the resources obtained reduces as more of the resources are extracted. The problem structure studied in this work also naturally appears while modeling the performance of hydro turbines [5, 18], designing chemical processes [17] and scheduling compressors in petroleum reservoirs [7].

Figure 1: Example production functions.



We describe and analyze the model for a single production process exhibiting these characteristics. We are motivated by applications that contain many processes having these characteristics, which are linked in some way, such as having common processing facilities or distribution networks. In Section 5 we describe an example application in which the facilities are part of a network in which demands are to be met. For such applications, the model we develop would be used for each production facility in the problem. Optimizing production for problems that contain multiple processes of this structure is complicated because the amount of each product produced is a *nonconvex* function of the cumulative mixture production up to that time. In addition, there are fixed costs associated with starting production, requiring the use of {0,1}-decision variables. We find that this nonconvex mixed-integer nonlinear programming problem is hard to solve directly using state-of-the-art software packages such as BARON [23] or Couenne [4], because the general purpose techniques for relaxing the nonconvex constraints used in these solvers appear to produce weak lower bounds. Our approach is to develop accurate and computationally useful formulations for an individual process model, which can then be used in a formulation that plans multiple such processes.

We now describe the production model for a single production process. Consider a production horizon $[0, L]$ with $x(t)$ representing the production rate of a mixture of products $\mathcal{P}$ at time $t \in [0, L]$. The cumulative production of the mixture up to time t is denoted by $v(t)$ and can be expressed as

$$v(t) = \int_0^t x(s)\, \mathrm{d}s, \qquad t \in [0, L]. \tag{1}$$

As input to the problem, we are given a production function $f(\cdot) : \mathbb{R}_+ \to \mathbb{R}_+$ that defines the maximum production rate of the mixture as a function of cumulative production up to time t. Hence, $x(t)$ must satisfy:

$$x(t) \leq f(v(t)), \qquad t \in [0, L]. \tag{2}$$

For each product $p \in \mathcal{P}$, we are given a function $g_p : \mathbb{R}_+ \to [0, 1]$, that specifies the fraction of the mixture that is product $p \in \mathcal{P}$ as a function of the cumulative mixture production. The rate of production of product p at time t, denoted by $y_p(t)$, is given by:

$$y_p(t) = \; x(t) \; g_p(v(t)), \qquad p \in \mathcal{P}, t \in [0, L]. \tag{3}$$

We make the following assumptions on the nonlinear functions in this model.

Assumption 1. *The production function $f : \mathbb{R}_+ \to \mathbb{R}_+$ is differentiable and concave, the ratio functions $g_p : \mathbb{R}_+ \to [0, 1]$ are non-increasing for $p \in \mathcal{P}^+$ (useful products) and non-decreasing for $p \in \mathcal{P}^-$ (byproducts), and the functions $g_p, p \in \mathcal{P}$ are differentiable and satisfy $\sum_{p \in \mathcal{P}} g_p(v) = 1$ for all $v \geq 0$.*

To facilitate modeling of a fixed cost for beginning production at time $t \in [0, L]$, the model contains a binary step function $z : [0, L] \to \{0, 1\}$ to indicate if production can occur at time $t \in [0, L]$. Let M^v be an upper bound on the maximum cumulative production. Then the following constraint ensures that production occurs only after the production facility has been opened:

$$v(t) \leq \mathrm{M}^v z(t), \qquad t \in [0, L]. \tag{4}$$

Constraints (1) and (4) ensure that once production commences, it may continue until the end of the time horizon. In other words, for any $t_1, t_2 \in [0, L]$, if $t_1 > t_2$ then $z(t_1) \geq z(t_2)$.

Figure 1 gives an example of how the maximum production rate $f(\cdot)$, useful product fraction $g_p(\cdot), p \in \mathcal{P}^+$, and byproduct fraction $g_p(\cdot), p \in \mathcal{P}^-$ evolve as a function of the cumulative mixture production. Our model can be extended to the case where $f(\cdot)$ is not necessarily differentiable and concave and the functions $g_p(\cdot)$ $\forall p \in \mathcal{P}$ are not necessarily differentiable and monotone. The primary modification that would be required is in the method for constructing the piecewise-linear approximations that are used in our formulation.

A time-discretization of this production planning problem is required to obtain a model that is suitable for implementation and numerical evaluation. Tarhan et al. [22] introduce a natural discrete-time mixed-integer nonlinear programming (MINLP) formulation (reviewed in Section 2.1) which discretizes the production horizon $[0, L]$ into T time periods and transforms constraints (1)-(4) into their discrete-time counterparts.

The first contribution of our work is to introduce an alternative discrete-time MINLP formulation that is more accurate than the one which appeared in [22]. This alternative formulation has also been independently derived by Gupta and Grossmann [12]. The key difference between the alternative formulation and that proposed in [22] lies in the discretization of constraints (3). In [22], the authors discretize the product production rate ($y_{p,t}$) during a time period t by multiplying the discretized production rate at the start of a time period (x_t) with the product production ratio at the end of the previous time period ($g_p(v_{t-1})$). We show in Section 2 that this discretization yields an MINLP formulation that may have significant errors in the approximation of the actual product production rate $y_p(t)$.

In this work, we propose an alternative discrete-time MINLP formulation that *exactly* calculates the amount of product produced during each time period. This formulation is based on the cumulative product production $\int_0^t y_p(s)\,\mathrm{d}s$. In Section 2.2, we demonstrate how this formulation results in a model that is a more accurate representation of the underlying continuous-time formulation. This improvement in accuracy is realized irrespective of the nature of the production functions $f(\cdot)$ and $g_p(\cdot)$. However, if the production functions satisfy the concavity and monotonicity assumptions, our proposed formulation exploits the property that the integral of a nondecreasing (nonincreasing) function is convex (resp. concave) thereby transforming the terms involving products of non-convex functions ($x_t g_p(v_{t-1})$) into *univariate convex/concave* functions representing the cumulative product production. As a result, as demonstrated by numerical experiments in Section 5, this formulation yields a model that is not only more accurate but is also easier to approximately solve.

As the second contribution of this work, we derive three different mixed-integer *linear* programming (MILP) models that approximate or relax the convex/concave functions. The first model, piecewise-linear approximation (PLA), approximates each of the nonlinear functions involved using piecewise-linear forms modeled with variables that are constrained to be special ordered sets of type 2 (SOS2) [3]. The PLA model is an approximation of the production planning problem but does not provide a bound on the optimal value of the MINLP model. In order to provide guarantees on solution quality, we also propose two MILP *relaxations* of the MINLP model based on piecewise-linear under- and over- approximations of the nonlinear functions. The secant relaxation (SEC) uses multiple tangents and a single secant to envelope the convex/concave cumulative production functions. An advantage of this formulation is that it does not require SOS2 variables. We also extend the SEC relaxation to the k-secant relaxation (k-SEC), which uses multiple tangents and multiple secants to relax the convex/concave cumulative production function. . The addition of multiple secants requires the use of SOS2 variables, similar to the PLA model. In Section 5, we compare the MILP formulations in terms of computational difficulty to solve, quality of solution produced, and in the case of the relaxations, the quality of the bound produced.

The third contribution of this work is to demonstrate two techniques for improving the LP relaxations of the proposed MILP formulations. These techniques exploit two special properties of this problem. First, a model of each of the nonlinear functions is required only when a corresponding binary indicator variable is set to one. This allows a formulation with a better LP relaxation bound to be obtained by slightly modifying the constraints of the SOS2 formulation. Second, we exploit the fact that the cumulative total production, which is the argument to the nonlinear functions being approximated, is increasing over time. This enables the derivation of valid inequalities that relate the variables of the SOS2 approximations of these functions in consecutive time periods to each other.

One special property of this production planning problem is that all the nonlinear production functions $f(\cdot)$ and $g_p(\cdot), p \in \mathcal{P}$ share the same domain $[0, \mathrm{M}^v]$ because they are functions of the same argument $v(t)$. One way to model piecewise-linear approximations and relaxations of these nonlinear functions is to introduce a set of break points for each function approximation/relaxation. The advantage of this method is that the choice of break points approximating each function $f(\cdot)$ and $g_p(\cdot), p \in \mathcal{P}$ is flexible, thereby allowing each function to use break points that provide the best approximation for that function. However, this approach requires introducing a separate set of branching entities (SOS2 variables) for each function, which can significantly increase the difficulty in solving the resulting model. An alternate model introduces a single set of SOS2 variables that are shared between all the production functions $f(\cdot)$ and $g_p(\cdot), p \in \mathcal{P}$. Such a formulation is possible only if the functions share the same domain and use the same set of break points in that domain.

While sharing the SOS2 variables is computationally desirable, the requirement that the same set of break points be used for all functions may reduce the accuracy of each of the individual function approximations/relaxations. In addition, most existing algorithms [6, 20, 23] for finding piecewise-linear approximations/relaxations are designed only for a single univariate convex/concave function. In Section 4, we propose nonlinear programming (NLP) formulations for determining the best possible piecewise-linear approximations and relaxations of multiple univariate convex/concave functions, with the requirement that the piecewise-linear functions share the same set of break points. We demonstrate that our method can reap the computational benefits of sharing break points while still maintaining the accuracy of each function approximation. The recent work of [11] has also considered the problem of finding piecewise-linear approximations of multiple functions sharing the same domain and using the same set of break points. The most important difference in our work is that in addition to finding a piecewise linear approximation, we also consider the problem of finding a good piecewise-linear *relaxation*.

After we had completed a preliminary draft of this work we were informed that Gupta and Grossmann [12] independently discovered the alternative formulation that we present in Section 2.2, and similarly found that this formulation yields computational benefits and improved formulation accuracy. Gupta and Grossmann also derived a MILP formulation of a piecewise-linear approximation of the alternative formulation. While the key idea of this formulation is the same, there are a number of differences which we now highlight. First, our production model is slightly different in that we consider a production process in which we are given functions for the *fraction* of the total mixture that corresponds to each product, as opposed to the ratio of one component of the mixture to another. This formulation may be advantageous numerically because the fraction is bounded between zero and one. More significantly, there are several differences between our approaches for obtaining piecewise-linear approximations. In addition to obtaining a piecewise-linear approximation as in [12], we also obtain piecewise-linear relaxations, which can be used to obtain bounds on the best possible MINLP solution value. We also show how the MILP formulations of the piecewise-linear models can be strengthened, and when necessary we use special ordered sets of type II (SOS2) to model the piecewise-linear functions, as opposed to explicitly introducing binary decision variables as in [12]. The SOS2 formulation is locally ideal, which is a desirable property for a mixed-integer programming formulation, whereas the formulation used in [12] does not have this property [26]. Finally, we study the problem of finding piecewise-linear approximations and relaxations when multiple functions sharing the same argument are to be approximated, with the restriction that they be approximated using the same set of break points, which is an important component of a solution process that uses this formulation.

The remainder of this paper is organized as follows. In Section 2, we present the two discrete-time MINLP formulations. In Section 3, we provide the three MILP approximations/relaxations of the proposed MINLP formulation, and demonstrate how these formulations can be strengthened. In Section 4, we introduce NLP formulations that identify piecewise-linear functions that most accurately approximate and relax multiple nonlinear functions that share the same domain. In Section 5 we present a computational study that compares the accuracy and performance of all formulations introduced in Section 2 and 3. We close with concluding remarks in Section 6.

2 Discrete-Time Formulations

This section describes two discrete-time MINLP formulations of the production planning problem that considers a decision horizon of finite decision-making periods $\mathcal{T} = \{1, 2, \ldots, T\}$ each of length Δ_t $(t \in \mathcal{T})$, such that $L = \sum_{t=1}^{T} \Delta_t$. For each $t \in \mathcal{T}$, let x_t represent the total mixture production quantity during period t, v_t represent the cumulative mixture production up to and including time period t, and $y_{p,t}$ represent the amount of product $p \in \mathcal{P}$ produced in period t. Also, let the binary variable $z_t \in \{0, 1\}$ indicate if production can occur during period $t \in \mathcal{T}$.

The discrete-time MINLP formulations we present in this section all include the constraints

$$v_t = v_{t-1} + x_t, \qquad t \in \mathcal{T} \tag{5a}$$
$$z_t \geq z_{t-1}, \qquad t \in \mathcal{T} \setminus \{1\} \tag{5b}$$
$$v_t \leq \mathrm{M}^v \, z_t, \qquad t \in \mathcal{T} \tag{5c}$$
$$x_t \leq \Delta_t \, f(v_{t-1}), \qquad t \in \mathcal{T}. \tag{5d}$$

where $v_0 \stackrel{\text{def}}{=} 0$ and $P \stackrel{\text{def}}{=} |\mathcal{P}|$. For ease of notation, we define

$$Y \stackrel{\text{def}}{=} \big\{ z \in \{0,1\}^T, x \in \mathbb{R}_+^T, v \in \mathbb{R}_+^{T+1}, y \in \mathbb{R}_+^{P \times T} : (5) \big\}.$$

Constraint (5a) is a direct discrete-time analog to (1) and (5b) simply states that once a facility is opened it must stay open. The constraint (5c) is a discrete-time analog of (4). Constraint (5d) is an approximation of the continuous-time constraints (2), where we assume that the maximum flow rate during period $t \in \mathcal{T}$ is determined by the cumulative production at the beginning of the period v_{t-1} and the interval length Δ_t. Alternative discretizations of (2) are possible, such as one that limits the production rate based on the cumulative production at the end of the time period, or one that uses an average of $f(v_{t-1})$ and $f(v_t)$. However, these alternative discretizations do not affect this work, so we consistently use equation (5d) in all of our formulations.

2.1 **Formulation** MINLP1

The first discrete-time formulation is MINLP1, defined as the set of $(z, x, v, y) \in Y$ that satisfy:

$$y_{p,t} = x_t g_p(v_{t-1}), \qquad p \in \mathcal{P}, t \in \mathcal{T}. \tag{6}$$

Constraint (6) is a natural discretization of (3), which approximates the amount of each product $p \in \mathcal{P}$ produced in period t by multiplying the total mixture production in the period, x_t, with the corresponding product p fraction, $g_p(v_{t-1})$ determined based on the cumulative production up to the beginning of period t. Such a discrete-time approximation has been used, for example, in [22].

There are two drawbacks of formulation MINLP1. To describe them, we first reformulate (6) as a combination of two constraints involving intermediate decision variables $u \in \mathbb{R}_+^{P \times T}$:

$$y_{p,t} = x_t u_{p,t}, \qquad p \in \mathcal{P}, t \in \mathcal{T} \tag{7a}$$

$$u_{p,t} = g_p(v_{t-1}), \qquad p \in \mathcal{P}, t \in \mathcal{T}. \tag{7b}$$

The first drawback is that MINLP1 contains two types of nonconvexities: the bilinear terms, written explicitly in (7a), and the nonconvex functions $g_p(\cdot)$ for each product $p \in \mathcal{P}$ in (7b). This combination of nonconvexities results in a formulation that is computationally difficult to solve, even approximately, as we illustrate in Section 5. The second drawback of MINLP1 is the inaccuracy of the discrete-time approximation used in (6). We next discuss an alternative MINLP formulation that eliminates this inaccuracy.

2.2 **Formulation** MINLP2

The second formulation is obtained by using an alternative approach to calculate $y_{p,t}$, the amount of each product $p \in \mathcal{P}$ that is produced in period $t \in \mathcal{T}$. First, we *exactly* calculate the cumulative amount of product $p \in \mathcal{P}$ produced up to and including time period $t \in \mathcal{T}$ using the continuous-time formulation. We define $w_p(t)$ as the cumulative production for each product $p \in \mathcal{P}$ up to time $t \in [0, L]$. Specifically, for any $p \in \mathcal{P}$ and $t \in [0, L]$,

$$w_p(t) \stackrel{\text{def}}{=} \int_0^t y_p(s)\,\mathrm{d}s = \int_0^t x(s) g_p(v(s))\,\mathrm{d}s \tag{8}$$

$$= \int_0^{v(t)} g_p(\theta)\,\mathrm{d}\theta \qquad \left(\text{since } \frac{\mathrm{d}v(t)}{\mathrm{d}t} = x(t)\right) \tag{9}$$

$$\stackrel{\text{def}}{=} h_p(v(t)). \tag{10}$$

The change in the variables of the integral from equation (8) to (9) yields $h_p(\cdot)$, the cumulative production function which *exactly* evaluates the total amount of product $p \in \mathcal{P}$ produced up to time $t \in [0, L]$. Now, we evaluate $y_p(t)$, the product production rate at a time $t \in [0, L]$ using the relationship

$$y_p(t) = \frac{\mathrm{d}w_p(t)}{\mathrm{d}t} \qquad p \in \mathcal{P}, t \in [0, L]. \tag{11}$$

A time-discretization of the constraints (10) and (11) yields an alternate formulation that uses variables $w_{p,t}$: MINLP2 is the set of $(z, x, v, y, w) \in Y \times \mathbb{R}_+^{P \times T}$ that satisfy

$$w_{p,t} = h_p(v_t), \qquad p \in \mathcal{P}, t \in \mathcal{T} \tag{12a}$$

$$y_{p,t} = w_{p,t} - w_{p,t-1}, \qquad p \in \mathcal{P}, t \in \mathcal{T} \tag{12b}$$

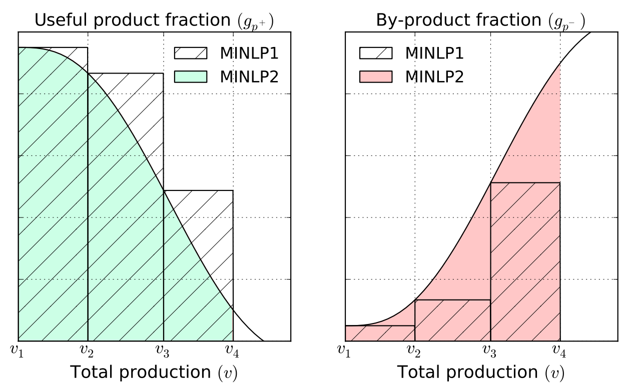
where $w_{p,0} \stackrel{\text{def}}{=} 0$ for $p \in \mathcal{P}$. Because the formulations MINLP2 and MINLP1 share the constraints (5) included in Y, they model the bounds on the mixture production variables, $x_t \leq \Delta_t f(v_{t-1})$, in the same way. However, in contrast to equation (12), formulation MINLP1 evaluates the cumulative product production $w_{p,t}$ as

$$w_{p,t} = \sum_{s \leq t} y_{p,s} = \sum_{s \leq t} x_s g_p(v_{s-1}), \quad p \in \mathcal{P}, t \in \mathcal{T}. \tag{13}$$

Equation (13) indicates that MINLP1 assumes that the product production fraction remains fixed at $g_p(v_{t-1})$ through the whole time period t. . Figure 2 illustrates the difference in the calculation of the product production $y_{p,t}$ using formulations MINLP1 and MINLP2 for a problem with four time periods. The total production up to and including time period t is denoted by v_t for each t. The hatched region represents the cumulative production $w_{p,t}$ calculated using equation (13) of the formulation MINLP1. In (13), the product fraction is assumed fixed throughout a time period, and so the estimate of the amount produced in a period is given by the area of the rectangle having base length equal to the total mixture amount produced in the period ($x_t = v_t - v_{t-1}$) and the height corresponding to that fixed product fraction, $g_p(v_{t-1})$. In contrast, the shaded region illustrates the correct calculation of the cumulative product production $w_{p,t}$ using $h_p(\cdot)$ defined in equation (12a) of formulation MINLP2. Clearly, MINLP1 overestimates the amount of useful product and underestimates the amount of byproduct.

Formulation MINLP2 also has two desirable computational properties. First, the total mixture production x_t and individual product productions $y_{p,t}$ are evaluated using *univariate* functions $f(\cdot)$ and $h_p(\cdot), p \in \mathcal{P}$, thereby eliminating the need to relax the product terms in equation (6). This enables us to derive univariate piecewise-linear approximations and relaxations, described in section 3, which, as we demonstrate in our numerical experiments in section 5, can yield high quality solutions and relaxation bounds. Second, since $g_p(\cdot)$ is non-increasing for useful products $p \in \mathcal{P}^+$, and $g_p(\cdot)$ is non-decreasing for byproducts $p \in \mathcal{P}^-$, it follows that $h_p(\cdot)$ is concave for $p \in P^+$ and convex for $p \in P^-$. Hence, product production profiles $y_{p,t}$ can be evaluated as a difference of univariate convex functions for byproducts and as a difference of univariate concave functions for useful products.

Figure 2: Differences in the estimation of product production $y_{p,t}$ using MINLP1 and MINLP2.



3 Piecewise-Linear Approximations and Relaxations

MINLPs are challenging to solve since they may include both nonlinear (and possibly nonconvex) functions and integer variables. A common approach is to approximate the nonlinear functions with piecewise-linear functions, and then solve the corresponding approximation as a mixed-integer *linear* program (MILP) (e.g., [1, 3, 8, 9, 15, 16, 20, 27, 26]). In this section, we describe one MILP approximation and two MILP relaxations of the set MINLP2.

The formulations in this section are based on piecewise-linear functions that either approximate or provide bounds on $f(\cdot)$ and $h_p(\cdot)$, $p \in \mathcal{P}$, over the domain. A piecewise-linear function $\bar{f}$ having m line segments and domain $[0, \mathrm{M}^v]$ is defined using break points $B \in \mathbb{R}^{m+1}$ with $0 = B_0 < B_1 < B_2 < \cdots < B_m = \mathrm{M}^v$, and function values $F = (F_0, F_1, \ldots, F_m)$ at these points:

$$\bar{f}(v; B, F) \stackrel{\text{def}}{=} F_{k-1} + \left(\frac{F_k - F_{k-1}}{B_k - B_{k-1}}\right)(v - B_{k-1}), \quad B_{k-1} \le v \le B_k,\ k = 1, \ldots, m. \tag{14}$$

Similarly, the piecewise-linear functions $\bar{h}_p$ for $p \in \mathcal{P}$ are defined as follows:

$$\bar{h}_p(v; B, H_p) \stackrel{\text{def}}{=} H_{p,k-1} + \left(\frac{H_{p,k} - H_{p,k-1}}{B_k - B_{k-1}}\right)(v - B_{k-1}), \quad B_{k-1} \le v \le B_k,\ k = 1, \dots, m \tag{15}$$

where $H_p \in \mathbb{R}^{m+1}$, $p \in \mathcal{P}$. The choice of the break points B and piecewise-linear function values, F and $H_p, p \in \mathcal{P}$, depends on whether we are seeking an approximation as in Section 3.1, or a relaxation, as in Section 3.2.

3.1 Piecewise-linear approximation (PLA)

The first piecewise-linear approximation is obtained by choosing break-points $\hat{B}$, and function values $\hat{F}$ and $\hat{H}_p, p \in \mathcal{P}$, such that $\bar{f}(v; \hat{B}, \hat{F}) \approx f(v)$ and $\bar{h}_p(v; \hat{B}, \hat{H}_p) \approx h_p(v), p \in \mathcal{P}$ over $v \in [0, \mathrm{M}^v]$. Figure 3 demonstrates an approximation scheme with $\hat{F}_k = f(\hat{B}_k)$ and $\hat{H}_{p,k} = h_p(B_k)$, i.e., each function approximation is constituted from linear approximations taken at points that lie on the curve. We assume that $\hat{H}_{p,0} = h_p(0)$ $(= 0)$, so that $\bar{h}_p(0; \hat{B}, \hat{H}_p) = h_p(0)$ for each $p \in \mathcal{P}$. In Section 4 we discuss a method for choosing the break points $\hat{B}$ and function approximation values $\hat{F}$ and $\hat{H}_p$.

The piecewise-linear approximation model is then obtained by replacing the functions $f(\cdot)$ and $h_p(\cdot)$ for $p \in \mathcal{P}$ with their piecewise-linear approximations $\bar{f}(\cdot\,; \hat{B}, \hat{F})$ and $\bar{h}_p(\cdot\,; \hat{B}, \hat{H}_p)$ for $p \in \mathcal{P}$ in (5d) and (12a), respectively, yielding:

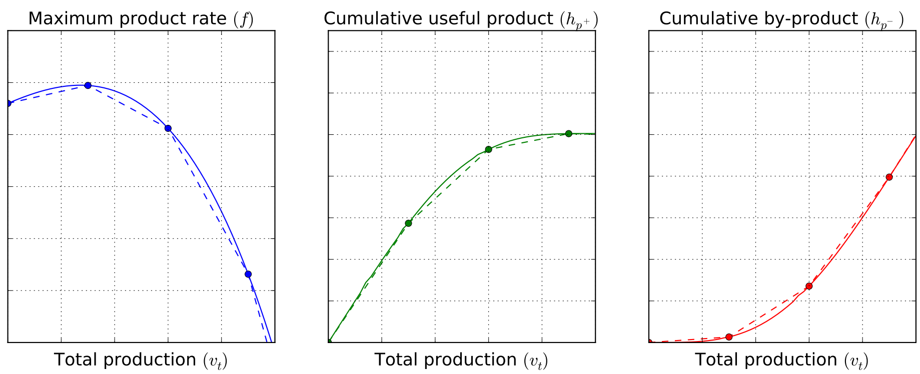
$$x_t \le \Delta_t\ \bar{f}(v_{t-1}; \hat{B}, \hat{F}), \qquad t \in \mathcal{T} \tag{16a}$$

$$w_{p,t} = \bar{h}_p(v_t; \hat{B}, \hat{H}_p), \qquad p \in \mathcal{P}, t \in \mathcal{T}. \tag{16b}$$

Note that, if the function approximations $\bar{f}$ and $\bar{h}$ are obtained by using the function values at the break points, then (16a) is a restriction of (5d), whereas (16b) is an approximation of (12a).

We next discuss how to model (16) using linear constraints and variables that are constrained to be a special ordered set of type 2 (SOS2) [3]. Define $\mathcal{M} = \{0, 1, \dots, m\}$. For each period $t \in \mathcal{T}$, we introduce a set of non-negative decision variables $\lambda_{t,k},\ k \in \mathcal{M}$ which are constrained to be SOS2: at most two elements in the set can be non-zero, and these two non-zero elements must be in adjacent positions (with respect to the ordering). Then, a piecewise-linear approximation of the formulation MINLP2 is obtained using additional variables $\lambda \in \mathbb{R}_+^{T \times (m+1)}$ and replacing (16) with the following constraints:

Figure 3: Piecewise-linear approximation model (PLA).



$$1 = \sum_{k \in \mathcal{M}} \lambda_{t,k}, \qquad t \in \mathcal{T} \tag{17a}$$

$$v_t = \sum_{k \in \mathcal{M}} \hat{B}_k \ \lambda_{t,k}, \qquad t \in \mathcal{T} \tag{17b}$$

$$x_t \leq \Delta_t \sum_{k \in \mathcal{M}} \hat{F}_k \ \lambda_{t-1,k}, \qquad t \in \mathcal{T} \tag{17c}$$

$$w_{p,t} = \sum_{k \in \mathcal{M}} \hat{H}_{p,k} \ \lambda_{t,k}, \qquad p \in \mathcal{P}, t \in \mathcal{T} \tag{17d}$$

$$\{\lambda_{t,k} \mid k \in \mathcal{M}\} \in \text{SOS2}, \qquad t \in \mathcal{T}. \tag{17e}$$

For each $t \in \mathcal{T}$, equations (17a) and (17b) ensure v_t is written as a convex combination of the break points $\{\hat{B}_k \mid k \in \mathcal{M}\}$, and (17c) and (17d) model $\bar{f}(v_t; \hat{B}, \hat{F})$ and $\bar{h}_p(v_t; \hat{B}, \hat{H}_p)$ for $p \in \mathcal{P}$ as a corresponding convex combination of the function values. The constraints (17e) then ensure that the convex combination is obtained only by using adjacent break points [2]. We adopt the convention that $\lambda_{0,0} = 1$ and $\lambda_{0,k} = 0$ for $k \neq 0$, so that (17c) for $t = 1$ enforces $x_1 \leq \Delta_1 f(0)$ which is consistent with the approximation (16a) because $v_0 = 0$.

To define a formulation that uses the model (17), first define the set W, which will be used in all formulations in this section, as

$$W \stackrel{\text{def}}{=} \big\{ z \in \{0,1\}^T, x \in \mathbb{R}_+^T, v \in \mathbb{R}_+^{T+1}, y \in \mathbb{R}_+^{P \times T}, w \in \mathbb{R}_+^{P \times T}, \lambda \in \mathbb{R}_+^{P \times (T+1)} :$$
$$v_t = v_{t-1} + x_t, \qquad t \in \mathcal{T}$$
$$z_t \geq z_{t-1}, \qquad t \in \mathcal{T} \backslash \{1\}$$
$$y_{p,t} = w_{p,t} - w_{p,t-1}, \qquad p \in \mathcal{P}, t \in \mathcal{T} \big\}.$$

The constraints in the definition of W are just restatements of (5a), (5b), and (12b). Then, we define the formulation PLA by:

$$\text{PLA} \stackrel{\text{def}}{=} \{(z,x,v,y,w,\lambda) \in W : (17),\ v_t \le \text{M}^v z_t,\ t \in \mathcal{T}\}.$$

The constraints $v_t \le \text{M}^v z_t,\ t \in \mathcal{T}$ are a restatement of (5c), which enforce that $v_t = 0$ when $z_t = 0$.

PLA approximates the formulation MINLP2 but is not guaranteed to be either a restriction or a relaxation of MINLP2. PLA includes integer restrictions and SOS2 constraints, both of which can be directly handled by commercial integer programming software.

We next discuss two techniques for improving the linear programming relaxation of the approximation PLA. The first is an application of a technique introduced in [21]. The idea is to exploit the fact that, when $z_t = 0$, the models of the functions $\bar{f}(v_t; \hat{B}, \hat{F})$ and $\bar{h}_p(v_t; \hat{B}, \hat{H}_p)$ for $p \in \mathcal{P}$ are not necessary, as in that case we immediately know that $x_t = v_t = 0$ and $y_{p,t} = w_{p,t} = 0$, $p \in \mathcal{P}$, and these conditions can all be obtained by replacing the constraint (17a) with

$$z_t = \sum_{k \in \mathcal{M}} \lambda_{t,k}, \qquad t \in \mathcal{T}. \tag{18}$$

In addition, with this substitution, the inequalities $v_t \le \text{M}^v z_t, t \in \mathcal{T}$ also become redundant because $z_t = 0$ already implies $v_t = 0$ via (18) and (17b). We define PLA-S as the formulation that includes (18) but excludes $v_t \le \text{M}^v z_t, t \in \mathcal{T}$ and (17a), specifically,

$$\text{PLA-S} = \{(z,x,v,y,w,\lambda) \in W : (17\text{b}) - (17\text{e}), (18)\}.$$

The following proposition proves the validity of the formulation PLA-S. Here, and elsewhere in the paper, for a set $S \subseteq \mathbb{R}^A \times \mathbb{R}^B$, we define $\text{Proj}_x(S) = \{x \in \mathbb{R}^A : (x,y) \in S \text{ for some } y \in \mathbb{R}^B\}$ as the projection of the set S into the space of x variables.

Proposition 2. $\text{Proj}_{(z,x,v,y,w)}(\text{PLA}) = \text{Proj}_{(z,x,v,y,w)}(\text{PLA-S})$.

Proof. Let $(z,x,v,y,w) \in \text{Proj}_{(z,x,v,y,w)}(\text{PLA})$ so $v_t \le \text{M}^v z_t,\ t \in \mathcal{T}$ and there exists λ' such that $(z,x,v,y,w,\lambda') \in W$ and satisfies (17). Let $\lambda_{t,k} = \lambda'_{t,k} z_t$ for all $t \in \mathcal{T}, k \in \mathcal{M}$, so that (18) and (17e) hold by construction. Let $t \in \mathcal{T}$. If $z_t = 1$, then $\lambda_{t,k} = \lambda'_{t,k}$ for $k \in \mathcal{M}$ and hence (17b) - (17d) trivially hold. Now suppose $z_t = 0$ so that $\lambda_{t,k} = 0$ for $k \in \mathcal{M}$. Because $v_t \le \text{M}^v z_t$, we have that $v_t = 0$, and hence $x_t = 0$ follows from $v_t = v_{t-1} + x_t$. Also, because $v_t = 0$ and $\hat{B}_k > 0$ for $k > 0$, (17b) implies that $\lambda'_{t,0} = 1$. Therefore, (17d) implies $w_{p,t} = 0$ for each $p \in \mathcal{P}$ since $\hat{H}_{p,0} = 0$. Thus, (17b) - (17d) all hold with zeros on both sides of the equations, and thus $(x,z,v,y,w) \in \text{Proj}_{(z,x,v,y,w)}(\text{PLA-S})$.

Now let $(z,x,v,y,w) \in \text{Proj}_{(z,x,v,y,w)}(\text{PLA-S})$ so there exists λ such that $(z,x,v,y,w,\lambda) \in W$ and satisfies (17b) - (17e) and (18). Let $t \in \mathcal{T}$. If $z_t = 1$, set $\lambda'_{t,k} = \lambda_{t,k}$ for all $k \in \mathcal{M}$. If $z_t = 0$, set $\lambda'_{t,0} = 1$ and $\lambda'_{t,k} = 0$ for $k > 0$. Then it is easy to check that $(z,x,v,y,w,\lambda') \in$ PLA, which yields the result. □

The following proposition shows that PLA-S has a stronger LP relaxation than PLA. For a set S defined by linear inequalities and integrality or SOS2 constraints, we define $\mathcal{R}(S)$ as the polyhedral relaxation of S, obtained by keeping the linear inequalities defining S but dropping the integrality and SOS2 constraints in S.

Proposition 3. $\mathrm{Proj}_{(z,x,v,y,w)} \mathcal{R}(\text{PLA-S}) \subseteq \mathrm{Proj}_{(z,x,v,y,w)} \mathcal{R}(\text{PLA})$ *and the inclusion can be strict.*

Proof. Let $(z,x,v,y,w) \in \mathrm{Proj}_{(z,x,v,y,w)} \mathcal{R}(\text{PLA-S})$ and so let λ be such that $(z,x,v,y,w,\lambda) \in \mathcal{R}(\text{PLA-S})$. By (17b) and (18), for each $t \in \mathcal{T}$, $v_t = \sum_{k\in\mathcal{M}} \hat{B}_k \; \lambda_{t,k} \leq \sum_{k\in\mathcal{M}} \lambda_{t,k} \mathrm{M}^v = \mathrm{M}^v z_t$. Next, for each $t \in \mathcal{T}$, let $\lambda'_{t,0} = \lambda_{t,0} + 1 - z_t$ and $\lambda'_{t,k} = \lambda_{t,k}$ for $k \in \mathcal{M} \setminus \{0\}$. Then $\sum_{k\in\mathcal{M}} \lambda'_{t,k} = \sum_{k\in\mathcal{M}} \lambda_{t,k} + 1 - z_t = 1$ by (18) and so (17a) holds. Also, $v_t = \sum_{k\in\mathcal{M}} \hat{B}_k \lambda_{t,k} = \sum_{k\in\mathcal{M}} \hat{B}_k \lambda'_{t,k} + (1-z_t)\hat{B}_0 = \sum_{k\in\mathcal{M}} \hat{B}_k \lambda'_{t,k}$ since $\hat{B}_0 = 0$, and so (17b) holds. (17c) and (17d) hold by an analogous argument. Thus, $(z,x,v,y,w,\lambda') \in \mathcal{R}(\text{PLA})$ and the first claim follows. Next, consider an instance with $T = 1$, $\mathcal{P} = \emptyset$ (so there are no y or w variables and the index t can be dropped) and with break points $(\hat{B}_0, \hat{B}_1, \hat{B}_2) = (0, 1/2, 1)$ and function values $(\hat{F}_0, \hat{F}_1, \hat{F}_2) = (0, 1/2, 0)$. The point $(z,x,v) = (1/2, 1/2, 1/2)$ is in $\mathrm{Proj}_{(z,x,v)} \mathcal{R}(\text{PLA})$ since it is easy to check that $(z,x,v,\lambda) \in \mathcal{R}(\text{PLA})$ with $\lambda = (0,1,0)$. However, $(1/2, /1/2, /1/2) \notin \mathrm{Proj}_{(z,x,v)} \mathcal{R}(\text{PLA-S})$ as there is no $\lambda' \in \mathbb{R}^3_+$ that satisfies (18), (17b) and (17c). Indeed, (18) and (17b) take the form $\lambda'_0 + \lambda'_1 + \lambda'_2 = 1/2$ and $(1/2)\lambda'_1 + \lambda'_2 = 1/2$, which because $\lambda' \geq 0$ implies $\lambda'_0 = \lambda'_1 = 0$ and $\lambda'_2 = 1/2$. But then (17c) is violated because $x = 1/2 > 0 = \sum_{k=0}^{2} \lambda'_k \hat{F}_k$. □

Results in [21] also demonstrate that MILP formulations of similar structure which use (18) are *locally ideal* [19] in the sense that the extreme points of the LP relaxation of such formulations, in the absence of other constraints, satisfy the integrality/SOS2 property.

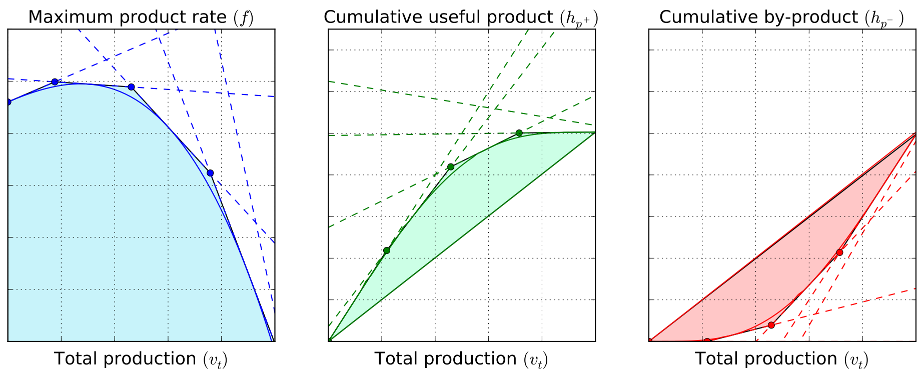
The next technique for improving the formulation PLA exploits the fact that we are building piecewise-linear models of multiple functions, and that the arguments of these functions, v_t for $t \in \mathcal{T}$, are related by the inequalities $v_t \geq v_{t-1}$ for $t \in \mathcal{T}$.

Proposition 4. *Let* $(z,x,v,y,w,\lambda) \in$ PLA-S. *Then,* λ *satisfies the inequalities:*

$$\sum_{k=k'}^{m} \lambda_{t,k} \geq \sum_{k=k'}^{m} \lambda_{t-1,k}, \quad k' \in \mathcal{M}, t \in \mathcal{T} \setminus \{1\}. \tag{19}$$

Proof. Let $t \in \mathcal{T} \setminus \{1\}$ and $k' \in \mathcal{M}$. If $z_{t-1} = 0$, then $\sum_{k=0}^{m} \lambda_{t-1,k} = 0$ and so (19) holds trivially. Therefore, assume $z_{t-1} = 1$, and so also $z_t = 1$ because $z_t \geq z_{t-1}$. Then, let $k_1^* = \max\{k \in \mathcal{M} \mid \lambda_{t-1,k} > 0\}$ and $k_2^* = \max\{k \in \mathcal{M} \mid \lambda_{t,k} > 0\}$. Then, because $v_t \geq v_{t-1}$, (17e) implies $k_1^* \leq k_2^*$. Observe that, if $k' > k_1^*$, then (19) is trivial because the right-hand side is zero, so assume $k' \leq k_1^*$. If $k_1^* = 0$, then $k' = 0$ and (19) holds because $\sum_{k=0}^{m} \lambda_{t,k} = z_t = 1 = z_{t-1} = \sum_{k=0}^{m} \lambda_{t-1,k}$. Thus, assume $k_1^* > 0$. Then (17e) implies

Figure 4: Piecewise-linear relaxation model with a single secant (SEC).



$\lambda_{t-1,k} = 0$ for all $k \notin \{k_1^*, k_1^* - 1\}$ and $\lambda_{t,k} = 0$ for all $k \notin \{k_2^*, k_2^* - 1\}$. Next, if $k' < k_2^*$ then

$$\sum_{k=k'}^{m} \lambda_{t,k} = z_t = 1 \geq \sum_{k=k'}^{m} \lambda_{t-1,k}$$

so that (19) holds. The remaining case is $k' \geq k_2^*$ and $k' \leq k_1^*$, which reduces to $k' = k_1^* = k_2^*$ because $k_1^* \leq k_2^*$. Then (17b) implies that $\hat{B}_{k'-1}\lambda_{t-1,k'-1} + \hat{B}_{k'}\lambda_{t-1,k'} = v_{t-1}$ and $\hat{B}_{k'-1}\lambda_{t,k'-1} + \hat{B}_{k'}\lambda_{t,k'} = v_t$. Then, using $v_t \geq v_{t-1}$, $\lambda_{t-1,k'-1} + \lambda_{t-1,k'} = 1$, and $\lambda_{t,k'-1} + \lambda_{t,k'} = 1$, it follows that $\lambda_{t,k'} \geq \lambda_{t-1,k'}$ which implies (19). $\square$

We define PLA-SS as the formulation obtained by adding the valid inequalities (19) to PLA-S, specifically:

$$\text{PLA-SS} = \{(z, x, v, y, w, \lambda) \in W : (17\text{b}) - (17\text{e}), (18), (19)\}.$$

A special property of this problem is that for each $t \in \mathcal{T}$, the nonlinear production functions $f(v_t)$ and $h_p(v_t), p \in \mathcal{P}$ are functions of the same argument v_t. We have exploited this property by introducing a *single* set of SOS2-constrained variables $\{\lambda_{t,k} \mid k \in \mathcal{M}\}$ to simultaneously approximate all these functions of v_t. In order to do this, we require that the set of break points in the interval $[0, \mathrm{M}^v]$ used to build the piecewise-linear approximations of $f(v_t)$ and $h_p(v_t), p \in \mathcal{P}$ must be the same for each of these functions. The potential drawback of this is that this added restriction may lead to a less accurate approximation of these functions for the same number of break points, compared to an approach that allows the break points to be selected separately for each function. However, the latter approach would require introducing a different set of SOS2-constrained variables for each function and each t, yielding a formulation with many more variables, and more significantly, many more SOS2 constraints.

3.2 Relaxations

In this section we present methods for using piecewise-linear functions to obtain MILP formulations that are a relaxation of MINLP2. Thus, the optimal value of a problem in which one of these MILP formulations is used in place of MINLP2 yields a bound on the best possible solution to MINLP2.

Secant Relaxation (SEC**)**

To obtain the first relaxation, we assume we choose break points $\bar{B}$ and functions values $\bar{F}$ and $\bar{H}_p, p \in \mathcal{P}$ such that

$$f(v) \leq \bar{f}(v; \bar{B}, \bar{F}), \qquad v \in [0, \mathrm{M}^v], \tag{20a}$$

$$h_p(v) \leq \bar{h}_p(v; \bar{B}, \bar{H}_p), \quad v \in [0, \mathrm{M}^v],\ p \in \mathcal{P}^+ \tag{20b}$$

$$h_p(v) \geq \bar{h}_p(v; \bar{B}, \bar{H}_p), \quad v \in [0, \mathrm{M}^v],\ p \in \mathcal{P}^- \tag{20c}$$

and such that the piecewise-linear functions $\bar{f}(v; \bar{B}, \bar{F})$ and $\bar{h}_p(v; \bar{B}, \bar{H}_p), p \in \mathcal{P}^+$ are concave, the piecewise-linear functions $\bar{h}_p(v; \bar{B}, \bar{H}_p), p \in \mathcal{P}^-$ are convex, and $\bar{h}_p(0; \bar{B}, \bar{H}_p) = 0$ for $p \in \mathcal{P}$. Figure 4 illustrates one such choice of $\bar{B}, \bar{F}$, and $\bar{H}_p, p \in \mathcal{P}$.

We now obtain a relaxation of MINLP2 by replacing equations (5d) and (12a) with the following:

$$x_t \leq \Delta_t \bar{f}(v_{t-1}; \bar{B}, \bar{F}), \qquad t \in \mathcal{T} \tag{21a}$$

$$(\bar{H}_{p,m}/\mathrm{M}^v) v_t \leq w_{p,t} \leq \bar{h}_p(v_t; \bar{B}, \bar{H}_p), \qquad p \in \mathcal{P}^+, t \in \mathcal{T} \tag{21b}$$

$$\bar{h}_p(v_t; \bar{B}, \bar{H}_p) \leq w_{p,t} \leq \left(\bar{H}_{p,m}/\mathrm{M}^v\right) v_t, \qquad p \in \mathcal{P}^-, t \in \mathcal{T}. \tag{21c}$$

It is immediate from (20a) that (21a) is a relaxation of (5d). For $p \in \mathcal{P}$, the constraint (12a), $w_{p,t} = h_p(v_t)$, is relaxed by using $\bar{h}_p(v_t; \bar{B}, \bar{H}_p)$ as an upper bound on $h_p(v_t)$. Because $h_p(v)$ is concave for $p \in \mathcal{P}^+$, the *secant* inequality $h_p(v) \geq h_p(0) + ((h_p(\mathrm{M}^v) - h_p(0))/(\mathrm{M}^v - 0))v = (\bar{H}_{p,m}/\mathrm{M}^v)v$ is valid for $v \in [0, \mathrm{M}^v]$. Similarly, for byproducts $p \in \mathcal{P}^-$, the function $h_p(v_t)$ is convex so the secant provides an upper bound over $[0, \mathrm{M}^v]$, and the lower bound is obtained using the piecewise-linear function $\bar{h}_p(v; \bar{B}, \bar{H}_p)$.

The constraints (21) can be compactly formulated using the variables $\lambda \in \mathbb{R}_+^{T \times (m+1)}$ with the constraints:

$$1 = \sum_{k \in \mathcal{M}} \lambda_{t,k}, \qquad t \in \mathcal{T} \tag{22a}$$

$$v_t = \sum_{k \in \mathcal{M}} \bar{B}_k \ \lambda_{t,k}, \qquad t \in \mathcal{T} \tag{22b}$$

$$x_t \leq \Delta_t \sum_{k \in \mathcal{M}} \bar{F}_k \ \lambda_{t-1,k}, \qquad t \in \mathcal{T} \tag{22c}$$

$$w_{p,t} = \sum_{k \in \mathcal{M}} \bar{H}_{p,k} \ \lambda_{t,k}, \qquad p \in \mathcal{P}, t \in \mathcal{T}. \tag{22d}$$

The equations (22b) and (22d) enforce that for each $p \in \mathcal{P}, t \in \mathcal{T}$, $(v_t, w_{p,t})$ are written as a convex combination of the points $(\bar{B}_k, \bar{H}_{p,k}), k \in \mathcal{M}$, which is equivalent to (21b) for $p \in \mathcal{P}^+$ and to (21c) for $p \in \mathcal{P}^-$. Observe that, although we use λ variables and the convex combination constraints (22a) as in the formulation PLA, in SEC we *do not* need to enforce the SOS2 constraints as in (17e). This is because in this case, for a fixed binary vector z, the feasible region we are modeling is a polyhedral set as illustrated in Figure 4. Note that this polyhedral set could alternatively be formulated using linear inequalities to define the lower and upper limits of the regions. We choose to use this extreme point based formulation of these polyhedra because this requires fewer constraints and has a close connection to the next formulation we will present.

We let SEC be the MILP formulation defined as follows:

$$\text{SEC} = \{(z, x, v, y, w, \lambda) \in W : (22), \ v_t \leq \text{M}^v z_t, \ t \in \mathcal{T}\}.$$

Similar to formulation PLA, we share the same sets of break points $\{\bar{B}_k \mid k \in \mathcal{M}\}$ across the $P + 1$ piecewise-linear functions $\bar{f}(\cdot\,; \bar{B}, \bar{F})$ and $\bar{h}(\cdot\,; \bar{B}, \bar{H}_p), p \in \mathcal{P}$.

As with formulation PLA, we can improve the LP relaxation of the formulation SEC by replacing the constraint (22a) with (18), and also removing the constraints $v_t \leq \text{M}^v z_t, \ t \in \mathcal{T}$ which are then redundant, obtaining formulation SEC-S defined as

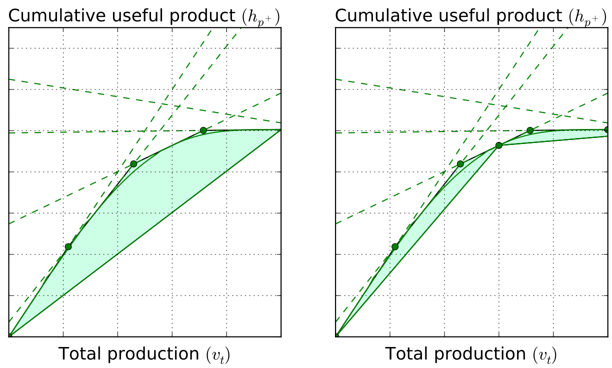
$$\text{SEC-S} = \{(z, x, v, y, w, \lambda) \in W : (18), (22\text{b}) - (22\text{d})\}.$$

Because SEC does not include SOS2 constraints, the inequalities (19) will not improve the LP relaxation, and so are not considered for SEC-S.

k-Secant Relaxation (k-SEC)

As Figure 4 illustrates, the formulation SEC relaxes the constraints $w_{p,t} = h_p(v_t)$ to allow points that lie in the shaded feasible region. The use of a single secant inequality to obtain a lower bound on $h_p(v)$ for $p \in \mathcal{P}^+$, or an upper bound on $h_p(v)$ for $p \in \mathcal{P}^-$ may allow solutions in the relaxation that are far from being feasible. We can obtain a tighter relaxation by using

Figure 5: Comparing SEC (left) and k-SEC formulation (right, $k = 2$) for constructing a relaxation of the constraint $w_p = h_p(v_t)$ for some $p \in \mathcal{P}^+$.



piecewise-linear functions for this purpose, as illustrated in Figure 5.

As in Section 3.2, we assume we choose break points $\bar{B}$ and functions values $\bar{F}$ and $\bar{H}_p, p \in \mathcal{P}$ that satisfy (20). In addition, we choose function values $\widetilde{H}_p, p \in \mathcal{P}$ that satisfy

$$h_p(v) \geq \bar{h}_p(v; \bar{B}, \widetilde{H}_p), \quad v \in [0, \mathrm{M}^v],\ p \in \mathcal{P}^+ \tag{23a}$$

$$h_p(v) \leq \bar{h}_p(v; \bar{B}, \widetilde{H}_p), \quad v \in [0, \mathrm{M}^v],\ p \in \mathcal{P}^-. \tag{23b}$$

We can then obtain a relaxation of the constraints (5d), $x_t \leq \Delta_t f(v_{t-1}),\ t \in \mathcal{T}$ and (12a), $w_{p,t} = h_p(v_t),\ t \in \mathcal{T}$, using variables $\lambda \in \mathbb{R}_+^{T\times(m+1)}$, the constraints (22a) - (22c), and

$$\sum_{k\in\mathcal{M}} \widetilde{H}_{p,k}\ \lambda_{t,k} \leq w_{p,t} \leq \sum_{k\in\mathcal{M}} \lambda_{t,k}\bar{H}_{p,k}, \qquad p \in \mathcal{P}^+, t \in \mathcal{T} \tag{24a}$$

$$\sum_{k\in\mathcal{M}} \bar{H}_{p,k}\ \lambda_{t,k} \leq w_{p,t} \leq \sum_{k\in\mathcal{M}} \lambda_{t,k}\widetilde{H}_{p,k}, \qquad p \in \mathcal{P}^-, t \in \mathcal{T} \tag{24b}$$

$$\{\lambda_{t,k} \mid k \in \mathcal{M}\} \in \mathrm{SOS2}, \qquad t \in \mathcal{T}. \tag{24c}$$

We let k-SEC be the resulting MILP formulation:

$$k\text{-SEC} = \big\{(z, x, v, y, w, \lambda) \in W : (22\text{a}) - (22\text{c}), (24),\ v_t \leq \mathrm{M}^v z_t,\ t \in \mathcal{T}\big\}.$$

Figure 5 illustrates the difference between formulations SEC and k-SEC.

Again, as with formulation PLA, we can improve the LP relaxation of the formulation k-SEC by replacing the constraint (17a) with (18), and also removing the constraints $v_t \leq \mathrm{M}^v z_t,\ t \in \mathcal{T}$ which are then redundant. In addition, because k-SEC includes SOS2 constraints, the inequalities (19) have the potential to further improve the LP relaxation. We define the formulation k-SEC-SS as:

$$k\text{-SEC-SS} = \big\{(z, x, v, y, w, \lambda) \in W : (18), (19), (22b) - (22c), (24)\big\}.$$

Table 1 summarizes the MILP formulations that have been introduced in this section.

Table 1: Summary of MILP formulations. All formulations contain variables $(z, x, v, y, w, \lambda) \in W$. The column SOS2 indicates whether or not the λ variables are SOS2-constrained.

Model	Constraints	SOS2?	Description
PLA	(5c),(17)	Yes	Approximation of MINLP2
PLA-S	(17b)-(17e),(18)	Yes	Stronger LP relaxation than PLA
PLA-SS	(17b)-(17e),(18),(19)	Yes	Stronger LP relaxation than PLA-S
SEC	(5c),(22)	No	Relaxation of MINLP2
SEC-S	(18),(22b)-(22c)	No	Stronger LP relaxation than SEC
k-SEC	(5c),(22a)-(22c),(24)	Yes	Relaxation of MINLP2
k-SEC-SS	(18),(19),(22b)-(22c),(24)	Yes	Stronger LP relaxation than k-SEC

4 Construction of piecewise-linear models

A key feature of our MILP formulations is that, for each period $t \in \mathcal{T}$, the functions $f(v_t)$ and $h_p(v_t)$ are approximated using piecewise-linear functions that share a common set of break points. In this section, we describe NLP formulations that identify the best possible approximation/relaxation of multiple univariate convex/concave functions with piecewise-linear forms that share the same set of break points. These formulations are nonconvex, and hence we can only expect to achieve local optimal solutions. However, our computational experience reported in Section 5.6 indicates that these formulations are easily solved by a commercial NLP solver, and the solutions are high quality. Our development of these NLP formulations is motivated by the need to simultaneously approximate a set of nonlinear functions. However, another benefit of these formulations is that they are flexible in the sense that it is easy to impose additional requirements on the piecewise-linear functions produced, such as requiring them to be convex or concave, monotone increasing or decreasing, etc.

4.1 Piecewise-linear approximation

We now describe a method for obtaining piecewise-linear approximations of nonlinear functions, with the goal to minimize the maximum error of the approximation. In this section, we extend previous work [6, 10, 13, 20] to propose an NLP formulation that identifies piecewise-linear approximations of multiple univariate convex/concave functions that share the same domain and break points.

Given a differentiable concave function $f(\cdot)$, differentiable concave functions $h_p(\cdot), p \in \mathcal{P}^+$, and differentiable convex functions $h_p(\cdot), p \in \mathcal{P}^-$, all defined on the domain $[0, \mathrm{M}^v]$, we seek a common set of break points $B \in \mathbb{R}_+^{m+1}$ with $0 = B_0 < B_1 < \cdots < B_m = \mathrm{M}^v$ and approximation function values $F \in \mathbb{R}^{m+1}$ and $H_p \in \mathbb{R}^{m+1}, p \in \mathcal{P}$, in order to minimize the weighted sum of maximum errors between the given functions and the constructed piecewise-linear approximations:

$$\min_{B,F,H_p,p\in\mathcal{P}} \; w^f \epsilon^f + \sum_{p\in\mathcal{P}} w_p^h \epsilon_p^h, \tag{25a}$$

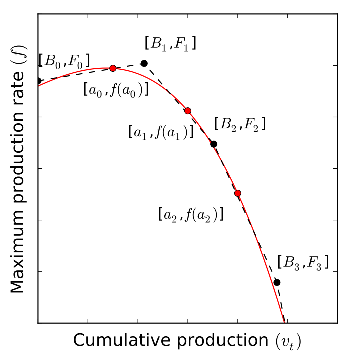
$$\text{subject to } \epsilon^f \geq |f(v) - \bar{f}(v; B, F)|, \qquad v \in [0, \mathrm{M}^v] \tag{25b}$$

$$\epsilon_p^h \geq |h_p(v) - \bar{h}(v; B, H_p)|, \qquad v \in [0, \mathrm{M}^v], \; p \in \mathcal{P} \tag{25c}$$

$$0 = B_0 \leq B_1 \leq \cdots \leq B_m = \mathrm{M}^v \tag{25d}$$

where $w^f > 0$ and $w_p^h > 0, p \in \mathcal{P}$ are fixed weights. We use weights $w^f = 1/\max\{|f(v)| : v \in [0, M_v]\}$ and $w_p^h = 1/\max\{|h_p(v)| : v \in [0, M_v]\}, p \in \mathcal{P}$ to prevent any single function from dominating the objective function.

Figure 6: Piecewise-linear approximation of a concave function $f(\cdot)$ with $m = 4$ line segments.



To formulate the semi-infinite program (25) as an NLP, we reformulate (25b) and (25c) using finitely many constraints using ideas from Geoffrion [10]. Consider first the concave function $f(\cdot)$, and consider an arbitrary interval $[B_{k-1}, B_k]$. Note that, on the this interval, the function $\bar{f}(\cdot; B, F)$ is affine:

$$\bar{f}(v; B, F) = F_{k-1} + \left(\frac{F_k - F_{k-1}}{B_k - B_{k-1}}\right)(v - B_{k-1}).$$

Because $f(\cdot)$ is concave, the function $\mathrm{err}_f^+(v) \stackrel{\text{def}}{=} f(v) - \bar{f}(v; B, F)$ is also concave on the interval $[B_{k-1}, B_k]$ for fixed B, F, and so the maximum of $\mathrm{err}_f^+(v)$ over this interval is given by $\mathrm{err}_f^+(a_k)$, where a_k is the point at which the derivative of $\mathrm{err}_f^+(v)$ equals zero. The point

a_k satisfies:

$$f'(a_k) = \frac{F_k - F_{k-1}}{B_k - B_{k-1}} \Leftrightarrow f'(a_k)(B_k - B_{k-1}) = F_k - F_{k-1} \tag{26}$$

since we can assume $B_k > B_{k-1}$. See Figure 6 for an example of the point a_k. Observe that, using (26),

$$\begin{aligned} \text{err}_f^+(a_k) &= f(a_k) - \left(F_{k-1} + \left(\frac{F_k - F_{k-1}}{B_k - B_{k-1}}\right)(a_k - B_{k-1})\right) \\ &= f(a_k) - F_{k-1} - f'(a_k)(a_k - B_{k-1}). \end{aligned}$$

Thus, $\epsilon^f \geq \text{err}_f^+(x)$ for all $x \in [B_{k-1}, B_k]$ if and only if there exists $a_k \in [B_{k-1}, B_k]$ that satisfies (26) and

$$\epsilon^f + F_{k-1} \geq f(a_k) - f'(a_k)(a_k - B_{k-1}).$$

Consider now $\text{err}_f^-(v) \stackrel{\text{def}}{=} \bar{f}(v; B, F) - f(v)$. Because $f(\cdot)$ is concave, it holds that

$$\max\{\text{err}_f^-(v) : v \in [B_{k-1}, B_k]\} = \max\{\text{err}_f^-(B_{k-1}), \text{err}_f^-(B_k)\}.$$

Thus, $\epsilon^f \geq \text{err}_f^-(x)$ for all $x \in [B_{k-1}, B_k]$ if and only if

$$\epsilon^f \geq \text{err}_f^-(B_i) = F_i - f(B_i), \quad i = k-1, k.$$

Putting this all together, we obtain that (25b) can be formulated using additional variables $a_k, k = 1, \ldots, m$ and the constraints

$$\begin{aligned} \epsilon^f + F_{k-1} &\geq f(a_k) - f'(a_k)(a_k - B_{k-1}), && k = 1, \ldots, m, && \text{(27a)} \\ \epsilon^f &\geq F_k - f(B_k), && k = 0, 1, \ldots, m, && \text{(27b)} \\ f'(a_k)(B_k - B_{k-1}) &= F_k - F_{k-1}, && k = 1, \ldots, m, && \text{(27c)} \\ B_{k-1} &\leq a_k \leq B_k, && k = 1, \ldots, m. && \text{(27d)} \end{aligned}$$

Identical arguments can be used to formulate constraints (25c), using additional variables $b_{k,p}, k = 1, \ldots, m, p \in \mathcal{P}$ and the constraints:

$$\begin{aligned} \epsilon_p^h + H_{k-1,p} &\geq h_p(b_{k,p}) - h_p'(b_{k,p})(b_{k,p} - H_{k-1,p}), && k = 1, \ldots, m, \quad p \in \mathcal{P}^+, && \text{(28a)} \\ \epsilon_p^h - H_{k-1,p} &\geq -h_p(b_{k,p}) + h_p'(b_{k,p})(b_{k,p} - H_{k-1,p}), && k = 1, \ldots, m, \quad p \in \mathcal{P}^-, && \text{(28b)} \\ \epsilon_p^h &\geq H_{k,p} - h_p(B_k), && k = 0, 1, \ldots, m, \; p \in \mathcal{P}^+, && \text{(28c)} \\ \epsilon_p^h &\geq h_p(B_k) - H_{k,p}, && k = 0, 1, \ldots, m, \; p \in \mathcal{P}^-, && \text{(28d)} \\ h_p'(b_{k,p})(B_k - B_{k-1}) &= H_{k,p} - H_{k-1,p}, && k = 1, \ldots, m, \quad p \in \mathcal{P}, && \text{(28e)} \\ B_{k-1} &\leq b_{k,p} \leq B_k, && k = 1, \ldots, m, \quad p \in \mathcal{P}, && \text{(28f)} \end{aligned}$$

where the constraints (28a) and (28c) are analogous to (27a) and (27b) since the functions h_p, $p \in \mathcal{P}^+$ are concave, whereas (28b) and (28d) follow because the functions h_p, $p \in \mathcal{P}^-$ are convex.

We also require that the piecewise-linear functions $\bar{h}_p(v; B, H_p)$ be exact at $v = 0$, and hence we enforce

$$H_{0,p} = 0, \quad p \in \mathcal{P}. \tag{29}$$

We therefore obtain the NLP formulation:

$$\min_{B,F,H,a,b} w^f \epsilon_f + \sum_{p \in \mathcal{P}} w_p^h, \tag{30a}$$

$$\text{subject to } (27), (28), (29), B_0 = 0, B_m = \mathrm{M}^v. \tag{30b}$$

where the inequalities $B_0 \leq B_1 \leq \cdots \leq B_m$ are omitted because they are implied by (27d). We use a locally optimal solution obtained by solving this NLP to define the break points $\hat{B}$ and function values $\hat{F}$ and $\hat{H}_p, p \in \mathcal{P}$, used in Section 3.1.

4.2 Piecewise-linear relaxation

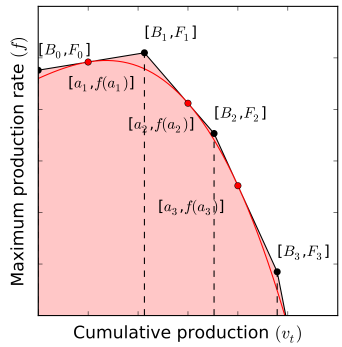
We now describe an NLP formulation that can be used to generate piecewise-linear functions that provide lower and upper bounds for a set of convex and concave functions, respectively. The key new feature of our approach is that it simultaneously approximates a set of functions sharing the same domain using the same set of break points for each function. Our work builds on ideas of algorithms [6, 20, 23] that find tight piecewise-linear upper and lower approximations of a single univariate convex function.

We seek piecewise-linear upper bounds on the concave differentiable production functions $f(\cdot)$, $h_p(\cdot), p \in \mathcal{P}^+$ and lower bounds on the convex differentiable functions $h_p(\cdot), p \in \mathcal{P}^-$. All these functions share the same domain $[0, \mathrm{M}^v]$. Observe that $f(\cdot)$ and $h_p(\cdot)$ are non-negative over $[0, \mathrm{M}^v]$. The goal of the NLP formulation we propose is to minimize a weighted sum of the areas between the curves of the functions being approximated and their corresponding piecewise-linear approximations.

The primary variables in the formulation are the break points $B \in \mathbb{R}^{m+1}$, and function values $F \in \mathbb{R}^{m+1}$ and $H_p \in \mathbb{R}^{m+1}, p \in \mathcal{P}$. The domain $[0, \mathrm{M}^v]$ is divided into m intervals, $[B_{k-1}, B_k]$, $k = 1, \ldots, m$, and the constructed piecewise-linear functions are linear on each interval, defined according to (14) and (15). On each interval, for each function being relaxed we require there be a point in the interval such that the piecewise-linear function is tangent to the function being approximated at that point. For the function $f(\cdot)$, we introduce variables a_k, $k = 1, \ldots, m$, to identify these points, and for the functions $h_p(\cdot), p \in \mathcal{P}$, we introduce variables $b_{k,p}$, $k = 1, \ldots, m$ to identify these points. Figure 7 highlights the notation used in this section and depicts a possible four-piece relaxation of a single function $f(\cdot)$.

For each interval $k = 1, \ldots, m$, the piecewise-linear approximation of $f(\cdot)$ in that interval

Figure 7: Piecewise-linear relaxation of a concave function $f(\cdot)$ with $m = 4$ segments.



is defined by the affine function $\ell_k(v) = f(a_k) + f'(a_k)(v - a_k)$ obtained as the first-order approximation of $f(\cdot)$ at the point a_k. As $f(\cdot)$ is concave on $[0, \mathrm{M}^v]$, it follows that $\ell_k(v) \geq f(v)$ on $[0, \mathrm{M}^v]$. The break points, B_k, and function values F_k, $k \in \mathcal{M}$, are then determined uniquely by finding the intersection of consecutive affine functions $\ell_k(v)$ and $\ell_{k+1}(v)$, as illustrated in Figure 7. Specifically, for $1 \leq k \leq m$, since F_k represents the value of the piecewise linear function at the point B_k, we must have $F_k = \ell_k(B_k)$, and for $0 \leq k < m$, we must also have $F_k = \ell_{k+1}(B_k)$, i.e.,

$$F_k = f(a_k) + f'(a_k)(B_k - a_k), \qquad k = 1, \ldots, m, \tag{31a}$$

$$F_k = f(a_{k+1}) + f'(a_{k+1})(B_k - a_{k+1}), \qquad k = 0, \ldots, m-1. \tag{31b}$$

We write similar constraints for the functions $h_p(\cdot)$:

$$H_{p,k} = h_p(b_{k,p}) + h_p'(B_k)(B_k - b_{k,p}), \qquad p \in \mathcal{P}, k = 1, \ldots, m, \tag{31c}$$

$$H_{p,k} = h_p(b_{k+1,p}) + h_p'(B_{k+1})(B_k - b_{k+1,p}), \qquad p \in \mathcal{P}, k = 0, \ldots, m-1. \tag{31d}$$

We measure the quality of the relaxation using the area between the two curves defined by the piecewise-linear approximation function and the corresponding function being relaxed. We combine the measures for the different functions into a single objective function using weights w^f for $f(\cdot)$ and w_p^h for each function $h_p(\cdot), p \in \mathcal{P}$. We use weights of $w^f = 1/\int_0^{M_v} f(s)\,\mathrm{d}s$ and $w_p^h = 1/\int_0^{M_v} h_p(s)\,\mathrm{d}s$ for our experiments. We therefore obtain the following NLP formulation:

$$\min_{B,F,H,a,b} w^f \Big[\frac{1}{2}\sum_{k\in\mathcal{M}}(F_k+F_{k-1})(B_k-B_{k-1}) - \int_0^{\mathrm{M}^v} f(s)\,\mathrm{d}s\Big] \tag{32a}$$
$$+ \sum_{p\in\mathcal{P}^+} w_p^h \Big[\frac{1}{2}\sum_{k\in\mathcal{M}}(H_{p,k}+H_{p,k-1})(B_k-B_{k-1}) - \int_0^{\mathrm{M}^v} h_p(s)\,\mathrm{d}s\Big]$$
$$+ \sum_{p\in\mathcal{P}^-} w_p^h \Big[\int_0^{\mathrm{M}^v} h_p(s)\,\mathrm{d}s - \frac{1}{2}\sum_{k\in\mathcal{M}}(H_{p,k}+H_{p,k-1})(B_k-B_{k-1})\Big]$$

subject to (31), (29)

$$B_{k-1} \le a_k \le B_k, \quad k=1,\dots,m \tag{32b}$$
$$B_{k-1} \le b_{k,p} \le B_k, \quad k=1,\dots,m, p\in\mathcal{P} \tag{32c}$$
$$B_0 = 0,\ B_m = \mathrm{M}^v, \tag{32d}$$

where we again include the constraints (29) which enforce that $\bar{h}_p(0; B, H_p) = 0$, $p \in \mathcal{P}$. The first term in the objective calculates the difference between the area under the piecewise-linear approximation of $f(\cdot)$, calculated by summing the area under each piece, and the area under the curve defined by $f(\cdot)$, calculated with the integral $\int_0^{\mathrm{M}^v} f(s)\,\mathrm{d}s$. Similar calculations are done for the byproduct functions $h_p(\cdot)$, $p \in \mathcal{P}$. Since the definite integrals $\int_0^{\mathrm{M}^v} f(s)\,\mathrm{d}s$ and $\int_0^{\mathrm{M}^v} h_p(s)\,\mathrm{d}s$ are constants, they can be eliminated from the objective in (32). We use a locally optimal solution obtained from solving this NLP to define the break points $\bar{B}$ and functions values $\bar{F}$ and $\bar{H}_p, p \in \mathcal{P}$ used in the SEC and k-SEC models in Section 3.2.

For the k-SEC model, we also require functions values $\widetilde{H}_p, p \in \mathcal{P}$ that satisfy (23). Having obtained the break points $\bar{B}$ from the NLP (32), we simply set

$$\widetilde{H}_{p,k} = h(B_{p,k}), \quad k = 0, 1, \dots, m, p \in \mathcal{P}.$$

Then, for $p \in \mathcal{P}^+$, $\bar{h}(v; \bar{B}, \widetilde{H}_p) \le h_p(v)$ for $v \in [0, \mathrm{M}^v]$ by concavity of $h_p(\cdot)$, and similarly for $p \in \mathcal{P}^-$, $\bar{h}(v; \bar{B}, \widetilde{H}_p) \ge h_p(v)$ for $v \in [0, \mathrm{M}^v]$ by convexity of $h_p(\cdot)$, and so (23) is satisfied.

5 Computational Results

We conducted numerical experiments to test the effectiveness of formulation MINLP2, the piecewise-linear approximation and relaxation MILP problems, and our NLP formulations for obtaining the piecewise-linear approximations and relaxations. The goals of these experiments are to: (1) compare the accuracy of formulations MINLP1 and MINLP2, (2) compare the computational effort of solving MINLP1 using a state of the art global optimization solver and approximations of MINLP2 using the three MILP formulations PLA, SEC, k-SEC, and (3) measure the quality of the piecewise-linear function approximations and relaxations produced by formulations (30) and (32), respectively.

We used Gurobi 4.5.1 to solve all MILPs, Conopt 3.14 to solve all nonlinear programs to local optimality, except formulations (30) and (32) for which we used Knitro 7.0.0 to exploit the multistart feature. We used BARON 9.3.1 to solve all MINLPs to global optimality. We used the default options for all solvers except that the LP method of Gurobi was changed to the barrier algorithm for better performance and the NLP solver of BARON was changed to Conopt 3.14 to avoid some observed numerical issues. We ran all experiments with a maximum time limit of one hour, relative optimality gap of 0.1% and feasibility tolerance of 10^{-5}. We ran all experiments with a single thread on a 2.30GHz Intel E5-2470 Xeon processor with 128GB RAM.

5.1 Sample Application

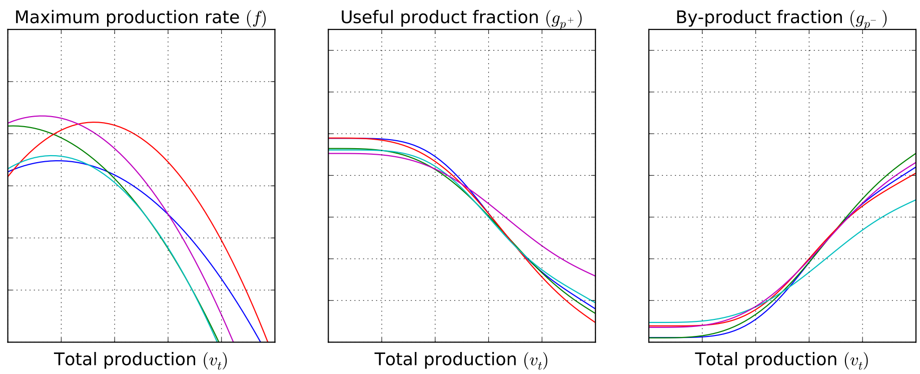
We conducted our numerical experiments on a multi-period production and distribution planning problem. In this problem, the production facilities ($\mathcal{I}$) produce a mixture of products ($\mathcal{P}$), of which the useful products $\mathcal{P}^+$ are supplied to match the demand of customers ($\mathcal{J}$). The manufacturing process also produces byproducts ($\mathcal{P}^-$) which are not sent to customers, but do incur a processing cost. The development of each facility $i \in \mathcal{I}$ over the planning horizon ($\mathcal{T}$) involves deciding when to open the facility and how much of the product mixture to produce over time. We let d_{jpt} denote the known demand for product $p \in \mathcal{P}^+$, at time period $t \in \mathcal{T}$ for each customer $j \in \mathcal{J}$.

Opening a facility $i \in \mathcal{I}$ at time period $t \in \mathcal{T}$ incurs a fixed cost α_{it}. Each unit of product $p \in \mathcal{P}$ processed at facility $i \in \mathcal{I}$ in time period $t \in \mathcal{T}$ incurs a processing cost of β_{ipt}. The cost of shipping a unit of product $p \in \mathcal{P}^+$ from facility $i \in \mathcal{I}$ to customer $j \in \mathcal{J}$ during time period $t \in \mathcal{T}$ is γ_{ijpt}, and δ_{jpt} represents the per unit penalty cost of unsatisfied demand of product $p \in \mathcal{P}^+$ for customer $j \in \mathcal{J}$.

Each facility $i \in \mathcal{I}$ operates according to the production process described in the introduction. Thus, in the discrete-time formulation of this problem, for each facility $i \in \mathcal{I}$, we have decision variables x_{it} and v_{it}, which represent the total mixture production during time period $t \in \mathcal{T}$ and cumulative mixture production up to and including time period $t \in \mathcal{T}$, respectively. The binary decision variables $z_{it}, t \in \mathcal{T}$ determine the time when each facility $i \in \mathcal{I}$ is opened (if at all), and the variables y_{ipt} represent the amount of each product $p \in \mathcal{P} = \mathcal{P}^+ \cup \mathcal{P}^-$ produced during time period $t \in \mathcal{T}$ at facility $i \in \mathcal{I}$.

The facilities are linked by the need to supply demand to the common set of customers $j \in \mathcal{J}$. Let q_{ijpt} define a decision variable that represents the amount of product $p \in \mathcal{P}^+$ transported from production facility $i \in \mathcal{I}$ to customer $j \in \mathcal{J}$ during time period $t \in \mathcal{T}$, and let u_{jpt} be a decision variable representing the unsatisfied demand of product $p \in \mathcal{P}^+$ for customer $j \in \mathcal{J}$

Figure 8: Sample production functions used for numerical experiments.



during time period $t \in \mathcal{T}$. Then, the multi-period production and distribution problem is:

$$
\begin{aligned}
\min \ & \sum_{t\in\mathcal{T}} \Big(\sum_{i\in\mathcal{I}} \alpha_{it} z_{it} + \sum_{i\in\mathcal{I}} \sum_{p\in\mathcal{P}} \beta_{ipt} y_{ipt} + \sum_{i\in\mathcal{I}} \sum_{j\in\mathcal{J}} \sum_{p\in\mathcal{P}} \gamma_{ijpt} q_{ijpt} + \sum_{j\in\mathcal{J}} \sum_{p\in\mathcal{P}} \delta_{jpt} u_{jpt} \Big) \\
\text{subject to} \ & \sum_{j\in\mathcal{J}} q_{ijpt} = y_{ipt} && i \in \mathcal{I}, p \in \mathcal{P}^+, t \in \mathcal{T} \\
& \sum_{i\in\mathcal{I}} q_{ijpt} = d_{jpt} - u_{jpt}, && j \in \mathcal{J}, p \in \mathcal{P}^+, t \in \mathcal{T} \qquad \text{(PP-}X\text{)} \\
& (z_i, x_i, v_i, y_i) \in X_i, && i \in \mathcal{I} \\
& u_{jpt} \geq 0, && j \in \mathcal{J}, p \in \mathcal{P}^+, t \in \mathcal{T}
\end{aligned}
$$

Here X_i denotes the set of constraints defining the production process for each facility $i \in \mathcal{I}$. One can model the production set in this production and distribution problem using either MINLP1 or MINLP2, which results in two different problems denoted by PP-MINLP1 and PP-MINLP2, respectively.

5.2 Test Instances

We report tests conducted on randomly generated instances of the sample application problem. All aspects of the dataset, including customer demands and unit costs, the production functions $f_i(\cdot)$ and the product production ratio function $g_{i,p}(\cdot)$ for each facility $i \in \mathcal{I}$ were generated randomly. Figure 8 illustrates some sample production functions that were used in our instances. The appendix provides additional details on how the instances were generated. We solved instances categorized by the number of binary variables. We grouped instances as small ($<$ 150 binary variables), and large ($>$200 binary variables). We measured formulation accuracy on small instances and computational impact on large instances.

Table 2: Solution statistics for formulation MINLP1 on 12 small instances.

$\|\mathcal{I}\|$	$\|\mathcal{T}\|$	$\|\mathcal{P}\|$	MINLP1 Gap (%)	Δy_{ipt} Max	Δy_{ipt} Avg	Gap to Best MINLP2 Obj (%)
5	5	2	17.5	3.75	0.85	29.9
5	5	2	13.8	2.64	0.60	27.4
5	5	2	15.7	3.02	0.71	24.8
5	10	2	24.5	2.43	0.44	19.0
5	10	2	20.6	2.22	0.54	19.3
5	10	2	27.1	2.37	0.45	17.3
10	10	2	35.1	3.29	0.59	19.4
10	10	2	32.6	2.98	0.55	15.6
10	10	2	35.1	2.27	0.55	17.7
10	15	2	34.8	2.62	0.44	13.5
10	15	2	33.6	2.53	0.45	12.6
10	15	2	35.5	2.40	0.38	11.3

5.3 Inaccuracy and Computational Difficulty of MINLP1

We first demonstrate the disadvantages of the formulation based on MINLP1, in terms of the difficulty in solving the problem, and the inaccuracy of the resulting solution. This experiment is performed on relatively small instances ($<$ 150 binary variables) of our sample application problem.

First, to analyze the computational difficulty of solving these small instances, we use BARON to attempt to solve PP-MINLP1, with a ten hour time limit. None of these instances are solved to optimality, and so we obtain the ending optimality gap, which is defined as $(UB - LB)/UB$, where UB and LB are the best upper and lower bounds, respectively, obtained by BARON in the time limit. Table 2 displays statistics about these values for twelve small test instances. Each row in this table corresponds to a different instance, and the first two columns describe the size of the instance in terms of the number of facilities $|\mathcal{I}|$ and the number of time periods $|\mathcal{T}|$. The column 'MINLP1 Gap' provides the ending optimality gaps of these instances. These large remaining gaps indicate that it is not possible to obtain provably near-optimal solutions of MINLP1 using the general-purpose global optimization solver like BARON.

We next study the quality of the solutions obtained using MINLP1. This is somewhat challenging because MINLP1 and MINLP2 represent two *different* approximations of the underlying problem. In particular, MINLP1 is based on the assumption that the fraction of products produced during period t is equal to $h_p(v_{t-1})$ throughout the period. Since $h_p(v)$ is increasing in v for byproducts and decreasing in v for desirable products, this assumption is optimistic, and leads to an over-estimation of the actual obtainable objective value. In contrast, MINLP2 is based on an exact calculation of the amount of each product p produced during each period, so the objective value in MINLP2 is a more accurate representation of the actual cost. Because of this difference, we cannot simply compare the objective value of a solution obtained form MINLP1 to a solution obtained using MINLP2.

To overcome this challenge, we first obtain a solution to PP-MINLP1, say $(z^1, x^1, v^1, y^1, q^1, u^1)$, and then *repair* the solution to obtain a feasible solution to the more accurate formulation PP-MINLP2. The solution $(z^1, x^1, v^1, y^1, q^1, u^1)$ is taken as the best feasible solution to PP-MINLP1 found by BARON within the time limit. To repair the solution, for each facility i, we fix the decision variables z_i^1, x_i^1, v_i^1, and then use the equations (12) to exactly calculate the amount of each product $p \in \mathcal{P}$ produced in each period $t \in \mathcal{T}$, given that x_{it}^1 units of the mixture are produced, and denote this amount by y_{ipt}^2. Then, with these values all fixed, we solve (PP-X), which then amounts to a transportation problem that determines how much of each product to ship from each facility to each customer in all time periods to minimize the transportation costs and penalty costs of unmet demand. This leads to a feasible solution to the more accurate formulation PP-MINLP2.

One measure of the inaccuracy of the formulation MINLP1 is the difference between the values y_{ipt}^1 obtained directly from solving PP-MINLP2, to the repaired values y_{ipt}^2, which we denote by $\Delta y_{ipt} = |y_{ipt}^1 - y_{ipt}^2|$. The columns under the heading Δy_{ipt} in Table 2 display the maximum and average of Δy_{ipt} over all facilities, products, and time periods. For comparison purposes, the average and maximum value of the y_{ipt} variables (over the solutions obtained in all instances) is 3.43 and 24.6, respectively. The average errors range between 0.38 and 0.85, and the maximum error ranges between 2.22 and 3.75. Thus, the estimates of the product quantities used in formulation MINLP1 are significantly different (at least 10% on average) from the actual.

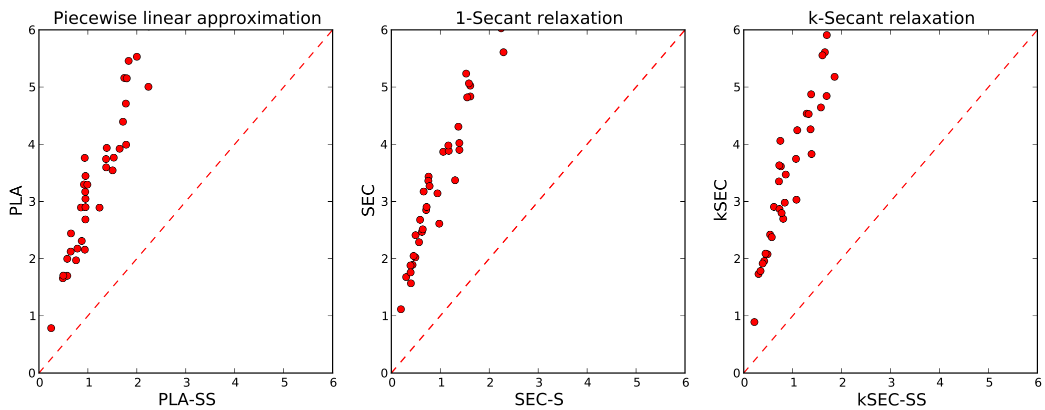
In the last column of Table 2, we give the percentage difference between the cost of the repaired solution to the cost of the best known feasible solution to PP-MINLP2, obtained using methods that directly use the formulation MINLP2, as described in the following sections. Here the percentage gap is calculated as $(A - B)/A$ where A is the cost of the repaired solution and B is the cost of the best known solution. Thus, we see that the best known solutions to PP-MINLP2 are between 11.3% and 29.9% less costly than the repaired solution obtained from solving PP-MINLP1.

Finally, we observe that the gap between using formulation MINLP2 and MINLP1 reduces as the number of time periods ($|\mathcal{T}|$) increases, which is expected as the approximation error of formulation MINLP2 decreases as the period length decreases. However, the difficulty in solving PP-MINLP1 also appears to increase with the number of time periods.

5.4 The effect of formulation strengthening on the MILP formulations

In our next experiment, we study the effect of using equations (18) and (19) to strengthen each of the three MILP formulations PLA-SS, SEC-S, and k-SEC-SS, compared to the formu-

Figure 9: Effect of formulation strengthening on terminal MILP optimality gaps (%) of the three MILP formulations.



lations PLA, SEC,and k-SEC, which do not use these. These tests are conducted on 36 large instances of PP-MINLP2. These instances have between 15-20 facilities, 15-25 time periods, and 2-6 products, resulting in instances with 200-500 binary variables for all formulations and 200-500 SOS2 sets of variables for the PLA and k-SEC formulations. The piecewise-linear functions used in these MILP approximation all have three line segments that were constructed using the formulations in Section 4. For each of these instances, we solved the MILP formulations with and without strengthening with a time limit of one hour. We then found the ending optimality gap of the MILP instance, calculated as $(UB - LB)/UB$, where UB and LB are the best upper and lower bounds, respectively, obtained for the MILP formulation within the time limit. Figure 9 displays scatter plots of the resulting optimality gaps for the MILP formulations with and without strengthening. When using PLA-SS, SEC-S, and k-SEC-SS, most instances have ending optimality gap less than 2%. In comparison, when using PLA, SEC, k-SEC most instances have optimality gaps in the range of 2-6%. Thus, it is clear that strengthening significantly helps solve these piecewise-linear MILP models.

We can further analyze the improvements obtained from using the strengthened formulations by comparing the quality of the linear programming (LP) relaxations. For an MILP instance, the LP relaxation gap is calculated as $(\hat{z} - z^{LP})/\hat{z}$, where z^{LP} is the LP relaxation objective value of the given instance, and $\hat{z}$ is the value of the best known feasible solution (found using *any* formulation for that instance). The LP relaxation gaps using the strengthened and unstrengthened versions of the three MILP formulations are summarized in Table 3. Each row in this table corresponds to an average over three different instances with the same size of $|\mathcal{I}|$, $|\mathcal{T}|$ and $|\mathcal{P}|$. The final row provides an average over all 36 instances. We observed more than a ten-fold improvement in the LP relaxation gaps across all three MILP formulations due to equations (18) and (19). In our remaining experiments, we use only the strengthened versions of the MILP formulations.

Table 3: Average linear programming relaxation gaps (%) for MILP formulations of MINLP2 with and without strengthening while solving 36 large instances of PP.

$\|\mathcal{I}\|$	$\|\mathcal{T}\|$	$\|\mathcal{P}\|$	PLA	PLA-SS	SEC	SEC-S	k-SEC	k-SEC-SS
15	15	2	20.97	1.38	21.07	1.47	21.10	1.47
15	15	4	21.17	1.59	21.23	1.62	21.37	1.67
15	15	6	22.73	1.64	23.10	1.74	23.13	1.91
20	15	2	21.33	1.28	21.30	1.32	21.30	1.33
20	15	4	22.30	1.46	22.47	1.46	22.57	1.53
20	15	6	22.50	1.51	22.80	1.59	22.70	1.78
20	20	2	21.93	1.68	22.07	1.75	22.17	1.77
20	20	4	21.57	2.11	21.87	2.20	22.20	2.36
20	20	6	23.07	2.23	23.57	2.43	23.80	2.50
20	25	2	21.43	2.39	21.70	2.53	21.73	2.53
20	25	4	21.47	2.37	21.70	2.47	22.03	2.60
20	25	6	22.53	2.56	23.07	2.81	23.37	3.06
Overall Average			21.92	1.85	22.16	1.95	22.29	2.04

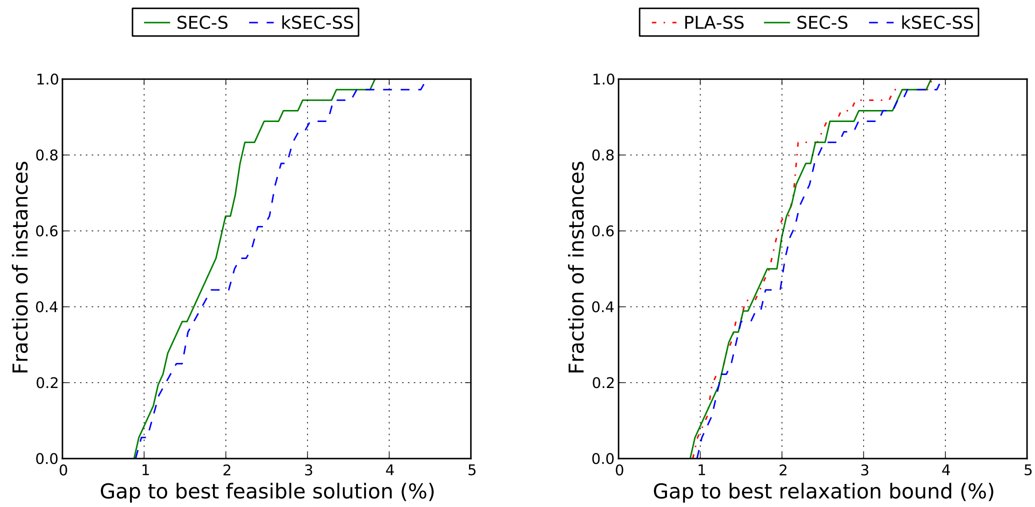
5.5 Comparing the MILP approximations and relaxations of MINLP2

We next compare the performance of the different piecewise-linear approximations and relaxations that we proposed in Section 3 for obtaining lower bounds and feasible solutions to PP-MINLP2. These tests are conducted on the same 36 large instances of PP-MINLP2 used in Section 5.4.

We used the MILP approximations and relaxations to obtain feasible solutions to PP-MINLP2 as follows. First, we solve each of the MILP problems PLA-SS, SEC-S, and k-SEC-SS with a one-hour time limit. Then, for each formulation, we take the best feasible MILP solution found, fix all the binary decision variables $\{z_{it}\ t \in \mathcal{T}\}$ for all facilities $i \in \mathcal{I}$, and then solve the resulting nonlinear program (NLP) using Conopt 3.14. This returns a feasible solution to PP-MINLP2, and hence provides an upper bound on the optimal objective value. The overall best upper bound for PP-MINLP2 is the minimum of the objective values of each of the feasible solutions found. The formulations SEC and k-SEC are relaxations of MINLP2. We solve these MILP relaxations with a time limit of one hour, and the lower bound on the MILP instance after that time limit is then a lower bound on the optimal objective value of problem PP-MINLP2. For each test instance, the larger of these two lower bounds is the best lower bound we obtain.

Figure 10 and Table 4 summarize the results of these experiments. We separately analyze the performance of these formulations in terms of quality of the lower bounds and the feasible solutions (upper bounds). Only SEC and k-SEC are guaranteed to provide lower bounds

Figure 10: Evaluating formulations PLA-SS, SEC-S, k-SEC-SS on large instances PP-MINLP2.



(a) Relaxation bound quality

(b) Feasible solution quality

for PP-MINLP2. To compare the quality of these lower bounds, for each instance and each formulation we compute the gap between the lower bound obtained from the formulation and the *best* feasible solution to PP-MINLP2 for that instance (found by any method using the procedure described in the previous paragraph). Figure 10(a) plots the cumulative distribution function of this gap over the 36 test instances for SEC and k-SEC, and the second and third columns of Table 4 provide summary statistics. Again, each row refers to an average over 3 instances of the same size. First, we observe that in both cases, the average bounds provided are reasonably small, between 1.81% and 2.11% on average, and less than 4% in all cases. Second, we observe that, except for some of the smallest instances, the bounds provided from formulation SEC are slightly better than those obtained from k-SEC; SEC yields average gaps of 1.81% compared to 2.11% from k-SEC. This result is somewhat counterintuitive, because the feasible region of the k-SEC formulation is a subset of the feasible region of the SEC formulation, so that if both of these formulations were solved to optimality, k-SEC must provide a bound at least as good as SEC. However, these MILP instances are not solved to optimality within the time limit, and so the bounds obtained are the best lower bounds on the respective MILP instances within the time limit. Because k-SEC contains SOS2 variables, these instances are more difficult to solve, and hence have a larger MILP optimality gap after the time limit, which translates to a worse bound on the original problem PP-MINLP2.

For each instance and each MILP formulation we also compute the gap between the feasible solution of PP-MINLP2 found by that formulation to the largest lower bound of PP-MINLP2 obtained for that instance. Figure 10(b) plots the cumulative distribution function of this gap over the 36 test instances each of the MILP formulations, and the last three columns of each

Table 4: Summary statistics for MILP approximations and relaxations of MINLP2 while solving 36 large instances of PP-MINLP2.

			Average gap to best feasible solution of PP-MINLP2 (%)		Average gap to best bound of PP-MINLP2 (%)		
$\lvert\mathcal{I}\rvert$	$\lvert\mathcal{T}\rvert$	$\lvert\mathcal{P}\rvert$	SEC-S	k-SEC-SS	PLA-SS	SEC-S	k-SEC-SS
15	15	2	1.02	1.01	1.03	1.00	1.09
15	15	4	1.01	1.32	1.06	1.10	1.14
15	15	6	1.81	2.58	1.85	1.83	1.98
15	20	2	1.16	1.14	1.16	1.16	1.21
15	20	4	1.34	1.59	1.36	1.38	1.44
15	20	6	2.02	2.78	2.03	2.06	2.23
20	20	2	1.47	1.42	1.45	1.45	1.58
20	20	4	1.91	2.33	1.91	1.98	2.07
20	20	6	2.58	2.88	2.62	2.60	2.94
25	20	2	1.98	2.00	2.00	2.04	2.15
25	20	4	2.17	2.54	2.17	2.38	2.41
25	20	6	3.27	3.76	3.27	3.55	3.61
Overall Average			1.81	2.11	1.83	1.88	1.99

row in Table 4 provide averages over three instances with the same size. We observe that all formulations provide good solutions for all of the test instances: each formulation provided a solution within 4% of the best known lower bound on all of the instances. However, we observed that formulation PLA-SS is marginally better than SEC-S and k-SEC-SS.

In summary, for these test instances, the SEC-S formulation appears to be preferable to the k-SEC-SS formulation in terms of both the quality of the lower bound obtained within a time limit, and the quality of the feasible solutions obtained. The SEC-S and PLA-SS formulation are complementary: PLA-SS can be used to consistently obtain high quality feasible solutions, whereas SEC-S can be used to provide a lower bound to evaluate the quality of this solution.

5.6 Evaluating approximations of nonlinear functions

In our final experiment, we study the quality of the piecewise-linear approximations and relaxations obtained using the formulations (30) and (32), respectively. Because these NLPs are nonconvex, we used a multi-start strategy with 100 randomly generated starting solutions to obtain different local optimal solutions. Each NLP is solved to local optimality using Knitro 7.0.0 with an iteration limit of 1000 and of the solutions generated we choose the solution with the best objective value. The starting points were randomly generated between the natural upper and lower bounds of all bounded variables. Starting points for unbounded variables were set to zero. We conducted numerical experiments on 765 sets of functions which were generated as described in Section 5.2. Each set of functions contained a single production rate function $f(\cdot)$ and between two and six cumulative product production functions $h_p(\cdot)$, leading to a total of 3645 functions. We calculate piecewise-linear approximations and relaxations consisting of three line segments. The average CPU time for solving all 100 NLP problems (30) and (32) for a given set of functions was in the order of a few seconds.

We first evaluate the quality of the piecewise-linear approximations produced by formulation (30). For a given piecewise-linear approximation $\hat{f}$ of a nonnegative function f, both with domain $[0, \mathrm{M}^v]$, we measure the accuracy of the approximation by calculating the *relative function evaluation error* (RFE), evaluated as $\max\{|\hat{f}(v) - f(v)| : v \in [0, \mathrm{M}^v]\}/\bar{f}$, where $\bar{f} = \max\{f(v) : v \in [0, \mathrm{M}^v]\}$. For each of the 765 sets of functions, we calculated three different sets of piecewise-linear approximations of these functions, and evaluated the RFE value of each approximation of each function in the set. First, we used uniformly spaced break points in the interval $[0, \mathrm{M}^v]$, and then optimized the function values $\hat{F}$ or $\hat{H}_p$ to minimize the error with these break points fixed. This strategy can be seen as a naive method for achieving a set of piecewise linear approximations that share the same set of break points. Next, for each set of functions we used the NLP formulation (30). Finally, for each *individual* function, we used an adaptation of formulation (30) in which only this single function is approximated. This method yields break points that are different for the functions in a set, and hence does not satisfy our goal of using the same set of break points. However, it provides an estimate of the best possible piecewise-linear approximation that could be obtained for each function, if the requirement to share break points were removed. Table 5 presents the average and maximum of the RFE over all these function approximations, taken over all sets and all functions in each set. We observe that using (30) yields approximations that are better than using uniform break points (0.96% average RFE compared to 1.21% average RFE). We also observe that the approximations using (30) are not much worse than the approximations using separate break points. Thus, we conclude that the loss in accuracy induced by the requirement to share break points across functions within a set is not that significant when using (30).

Table 5: Average accuracy of three-segment linear approximations of nonlinear functions on 765 sets of functions (3645 functions total).

		Relative function evaluation error (%)		
$\lvert\mathcal{P}\rvert$	# Sets	Separate break points	(30)	Uniform break points
2	315	0.78	1.01	1.16
4	225	0.73	0.94	1.21
6	225	0.72	0.91	1.28
Overall Average		0.75	0.96	1.21
Maximum		1.22	1.53	1.62

Table 6: Average accuracy of three-segment linear relaxations of nonlinear functions on 765 sets of functions (3645 functions total).

$\lvert\mathcal{P}\rvert$	# Sets	Relative area difference (%)	
		Separate break points	(32)
2	315	1.03	1.10
4	225	1.07	1.11
6	225	1.10	1.14
Overall Average		1.06	1.11
Maximum		1.41	1.43

We next evaluate the quality of the piecewise-linear relaxations produced by formulation (32). For a given piecewise-linear relaxation $\hat{f}$ of a function f, both with domain $[0, \mathrm{M}^v]$, we measure the accuracy of the relaxation by calculating the *relative area difference* (RAD), evaluated as $|A_f - A_{\hat{f}}|/A_f$, where $A_f = \int_0^{\mathrm{M}^v} f(v)dv$ and similarly for $A_{\hat{f}}$. For each set of functions, we obtain two sets of piecewise-linear relaxations. The first set is obtained using (32), and the second set is also obtained using (32), except that in this case we solve (32) separately for each function, allowing the break points to be different for different functions within a set. Table 6 provides the average and maximum RAD for these two methods over the 765 sets of functions. Once again, we observe that when using (32), the requirement that the break points be shared across functions within a set only slightly decreases the quality of the relaxation.

6 Concluding remarks

In this paper, a production planning problem was described with a special structure that the production process creates a mixture of desirable products and undesirable byproducts. A distinguishing feature of this nonconvex MINLP problem is that the fraction of undesirable byproducts increases monotonically as a function of the total mixture production up to that point in time. We present a continuous-time formulation and two discrete-time approximations (MINLP1 and MINLP2) of this problem. A MILP-based approximation (PLA) and two MILP-based relaxations (SEC, k-SEC) of this formulation were presented, and modifications to these formulations to improve the linear programming relaxations were derived.

Numerical experiments on small instances demonstrated our proposed formulation MINLP2 yielded solutions up to 30% less costly than those obtained by the natural formulation MINLP1. We found that the strengthening of the MILP formulations had a significant positive impact on the ability of a commercial MILP solver to obtain near-optimal solutions. We demonstrated that, in contrast to the formulation based on MINLP1, using the MILP formulations we were able to obtain good quality feasible solutions, along with lower bounds that verify that these solutions are near-optimal for this nonconvex MINLP problem. Finally, we found that by using our proposed NLP formulations for obtaining piecewise-linear relaxations and approximations for sets of functions, it is possible to enforce that these piecewise-linear functions share the same set of break points without significantly sacrificing the quality of the approximations.

7 Acknowledgements

We thank Stephen J. Wright for several suggestions that helped improve this work, and we thank Ignacio Grossmann for bringing the reference [12] to our attention. We are grateful to the anonymous referees for suggestions that helped improve the paper.

References

[1] Balakrishnan, A., Graves, S.C.: A composite algorithm for a concave-cost network flow problem. Networks **19**(2), 175–202 (1989)

[2] Beale, E.M.L., Forrest, J.: Global optimization using special ordered sets. Mathematical Programming **10**(1), 52–69 (1976)

[3] Beale, E.M.L., Tomlin, J.A.: Special facilities in a general mathematical programming system for non-convex problems using ordered sets of variables. Operations Research **69**, 447–454 (1970)

[4] Belotti, P., Lee, J., Liberti, L., Margot, F., Wächter, A.: Branching and bounds tighteningtechniques for non-convex MINLP. Optimization Methods & Software **24**(4-5), 597–634 (2009)

[5] Borghetti, A., D'Ambrosio, C., Lodi, A., Martello, S.: An MILP approach for short-term hydro scheduling and unit commitment with head-dependent reservoir. Power Systems, IEEE Transactions on **23**(3), 1115–1124 (2008)

[6] Burkard, R., Hamacher, H., Rote, G.: Sandwich approximation of univariate convex functions with an application to separable convex programming. Naval Research Logistics **38**, 911–924 (1991)

[7] Camponogara, E., de Castro, M.P., Plucenio, A., Pagano, D.J.: Compressor scheduling in oil fields. Optimization and Engineering **12**(1-2), 153–174 (2011)

[8] Dantzig, G.B.: On the significance of solving linear programming problems with some integer variables. Econometrica: Journal of the Econometric Society pp. 30–44 (1960)

[9] Geissler, B., Martin, A., Morsi, A., Schewe, L.: Using piecewise linear functions for solving MINLPs. In: J. Lee, S. Leyffer (eds.) Mixed Integer Nonlinear Programming, *The IMA Volumes in Mathematics and its Applications*, vol. 154, pp. 287–314. Springer New York (2012)

[10] Geoffrion, A.M.: Objective function approximations in mathematical programming. Mathematical Programming **13**(1), 23–37 (1977)

[11] Goldberg, N., Kim, Y., Leyffer, S., Veselka, T.D.: Adaptively refined dynamic program for linear spline regression. Tech. Rep. Preprint ANL/MCS-P3040-0912, Argonne National Laboratory (2012)

[12] Gupta, V., Grossmann, I.E.: An efficient multiperiod MINLP model for optimal planning of offshore oil and gas field infrastructure. Industrial & Engineering Chemistry Research **51**(19), 6823–6840 (2012)

[13] Imamoto, A., Tang, B.: Optimal piecewise linear approximation of convex functions. In: Proceedings of the world congress on engineering and computer science, pp. 1191–1194 (2008)

[14] Iyer, R., Grossmann, I.E.: Optimal planning and scheduling of offshore oil field infrastructure investment and operations. Industrial & Engineering Chemistry Research **37**(4), 1380–1397 (1998)

[15] Lee, J., Wilson, D.: Polyhedral methods for piecewise-linear functions I: the lambda method. Discrete Applied Mathematics **108**(3), 269–285 (2001)

[16] Markowitz, H.M., Manne, A.S.: On the solution of discrete programming problems. Econometrica: Journal of the Econometric Society pp. 84–110 (1957)

[17] Martín, M., Grossmann, I.E.: Energy optimization of hydrogen production from lignocellulosic biomass. Computers & Chemical Engineering **35**(9), 1798–1806 (2011)

[18] Orero, S., Irving, M.: A genetic algorithm modelling framework and solution technique for short term optimal hydrothermal scheduling. IEEE Transactions on Power Systems **13**(2), 501–518 (1998)

[19] Padberg, M.W., Rijal, M.P.: Location, scheduling, design and integer programming, vol. 19. Kluwer Academic Norwell (1996)

[20] Rote, G.: The Convergence Rate of the Sandwich Algorithm for Approximating Convex Functions. Computing (1992)

[21] Sridhar, S., Linderoth, J., Luedtke, J.: Locally ideal formulations for piecewise linear functions with indicator variables. Operations Research Letters **41**(6), 627 – 632 (2013)

[22] Tarhan, B., Grossmann, I.E., Goel, V.: Stochastic programming approach for the planning of offshore oil or gas field infrastructure under decision-dependent uncertainty. Industrial & Engineering Chemistry Research **48**(6), 3078–3097 (2009)

[23] Tawarmalani, M., Sahinidis, N.V.: Global optimization of mixed-integer nonlinear programs: A theoretical and computational study. Mathematical Programming **99**(3), 563–591 (2004)

[24] Van Den Heever, S.A., Grossmann, I.E.: An iterative aggregation/disaggregation approach for the solution of a mixed-integer nonlinear oilfield infrastructure planning model. Industrial & Engineering Chemistry Research **39**(6), 1955–1971 (2000)

[25] Van Den Heever, S.A., Grossmann, I.E., Vasantharajan, S., Edwards, K.: Integrating complex economic objectives with the design and planning of offshore oilfield infrastructures. Computers & Chemical Engineering **24**(2), 1049–1055 (2000)

[26] Vielma, J.P., Ahmed, S., Nemhauser, G.: Mixed-integer models for nonseparable piecewise-linear optimization: unifying framework and extensions. Operations Research **58**(2), 303–315 (2010)

[27] Vielma, J.P., Keha, A.B., Nemhauser, G.L.: Nonconvex, lower semicontinuous piecewise linear optimization. Discrete Optimization **5**(2), 467–488 (2008)

Appendix A

Production functions

We normalized the range of the total production function so that the cumulative production variable v has takes values in $[0, 1]$. We used bounded, concave production functions of the form $f(x) = r_0 + r_1 x + r_2 x^2$ with $r_2 < 0$. We randomly generated $f(\cdot)$ by choosing $r_0 \sim \mathrm{U}(15, 25)$, $r_2 \sim \mathrm{U}(-50, 0)$ and $r_0 + r_1 + r_2 \sim \mathrm{U}(-15, 0)$. This procedure was repeated to generate independently and identically distributed samples of production functions $f_i(\cdot)$ for each facility $i \in \mathcal{I}$. The family of functions generated by this procedure always ensures that $f(\cdot)$ is concave.

For each product $p \in \mathcal{P}$, we must generate either a monotonically increasing or decreasing function. Additionally, we must ensure that $\sum_{p \in \mathcal{P}} g_p(v) = 1, \forall v \in [0, 1]$. We simulated such functions by scaling and translating a randomly generated function from a family of functions. One such family of monotonically increasing functions is $\hat{g}(v) = \int_0^v u^k (1-u)^k du / \int_0^1 u^k (1-u)^k du, k = \{1, 2, 3\}$. We chose this family of functions because (a) it was rich enough to express productions functions of different kinds, (b) these functions are monotonically increasing, and (c) these functions satisfy $\hat{g}(0) = 0$ and $\hat{g}(1) = 1$. We then obtain functions $g_p(v), p \in \mathcal{P}$ by choosing positive real numbers a_p and b_p for $p \in \mathcal{P}$ and then setting

$$g_p(v) = a_p + (b_p - a_p)\hat{g}(v) \qquad \forall p \in \mathcal{P}.$$

Since the family of generated functions must satisfy $\sum_{p \in \mathcal{P}} g_p(v) = 1, \forall v \in [0, 1]$, we require

$$\sum_{p \in \mathcal{P}^+} a_p + \sum_{p \in \mathcal{P}^-} a_p = 1 \tag{33a}$$

$$\sum_{p \in \mathcal{P}^+} b_p + \sum_{p \in \mathcal{P}^-} b_p = 1 \tag{33b}$$

where $a_p = g_p(0)$ and $b_p = g_p(1)$. Additionally, we must also ensure that

$$a_p > b_p \qquad \forall p \in \mathcal{P}^+ \tag{33c}$$

$$a_p < b_p \qquad \forall p \in \mathcal{P}^- \tag{33d}$$

which makes the useful product ratio functions decreasing and the byproduct ratio function increasing. The following sampling procedure generates product ratio functions which satisfy all the required conditions.

- We sampled $\hat{a}_p \sim \mathrm{U}(0.8, 1), p \in \mathcal{P}^+$ and $\hat{a}_p \sim \mathrm{U}(0, 0.2), p \in \mathcal{P}^-$.
- We normalized the above samples so that $a_p = \frac{\hat{a}_p}{\sum_{p \in \mathcal{P}} \hat{a}_p}, \forall p \in \mathcal{P}$ which ensures that condition (33a) is satisfied.

- Next, we sampled $\hat{b}_p \sim \mathrm{U}(0, \frac{a_p}{2})$ $p \in \mathcal{P}^+$ and $\hat{b}_p \sim \mathrm{U}(2a_p, 1)$ $p \in \mathcal{P}^-$ which ensures that conditions (33c) and (33d) are satisfied.

- We normalized the above samples so that $b_p = \frac{\hat{b}_p}{\sum_{p\in\mathcal{P}} \hat{b}_p}, \forall p \in \mathcal{P}$ which ensures that condition (33b) is satisfied.

Customer Demands

We generated increasing demand profiles as follows. For each customer $j \in \mathcal{J}$ and product $p \in \mathcal{P}$, we randomly generated a demand time frame uniformly from start-time between $[1, \frac{T}{3}]$ and end-time between $[\frac{2T}{3}, T]$. We generated demands $\{d_{jpt}\} \sim \mathrm{U}(0.8, 1.2)$ for each time period within the demand time frame, sorted them in increasing order and scaled them by $\frac{\sum_{i\in\mathcal{I}}\sum_{p\in\mathcal{P}^+} h_{ip}(1)}{3|\mathcal{J}|}$. Here, $h_{i,p}(1)$ is the cumulative production of each useful product $p \in \mathcal{P}^+$ from a facility $i \in \mathcal{I}$. The scaling ensures that the total demand across all customers is roughly a third of the total possible production of useful products over all facilities.

Costs

We used the following procedure to generate the four different types of costs incurred at each facility. This procedure ensured that at the end of the planning horizon, the total facility opening cost is of the same order as the sum of operating, production and penalty costs. For each customer $j \in \mathcal{J}$, facility $i \in \mathcal{I}$ and product $p \in \mathcal{P}$ during the first time period, we generated operating costs $\beta_{i,p,1} \sim \mathrm{U}(0.8, 1.2)$, transportation costs $\gamma_{ijp1} \sim \mathrm{U}(0.8, 1.2)$, penalty costs $\delta_{j,p,1} \sim \mathrm{U}(16, 24)$ and fixed costs $\alpha_{i1} \sim \frac{\sum_{p\in\mathcal{P}^+} h_{ip}(1)}{3|\mathcal{J}|}\mathrm{U}(0.8, 1.2)$. We discounted costs at 5% for subsequent time periods.

Appendix B: Detailed computational results

We now provide additional details of the computational results described in Section 5 of the manuscript. Tables 7-10 provide detailed computational results for the experiments performed on the 36 large instances of PP-MINLP2. Tables 7 and 8 illustrate the effect of formulation strengthening on the linear programming relaxation gaps of the MILP formulations of MINLP2, and were used to generate the summary statistics displayed in Figure 9 and Table 3 of the manuscript, respectively. Table 9 shows feasible solution quality of the MILP approximation and relaxation schemes (PLA-SS, SEC-S and k-SEC-SS) by measuring the percentage gap to the best known solution bound of PP-MINLP2. Table 10 measures solution bound quality of the MILP relaxation schemes (SEC-S and k-SEC-SS) using the percentage gap to the best known solution feasible solutions of PP-MINLP2. Tables 9 and 10 were used to construct summary statistics for Table 4 in the manuscript.

Table 7: Terminal MILP optimality gaps for MILP formulations MINLP2 with and without strengthening while solving 36 large instances of PP-MINLP2.

Problem size			Terminal MILP Optimality Gap(%)					
$\lvert\mathcal{I}\rvert$	$\lvert\mathcal{T}\rvert$	$\lvert\mathcal{P}\rvert$	PLA-SS	PLA	SEC-S	SEC	k-SEC-SS	k-SEC
15	15	2	1.69	0.57	1.56	0.39	1.91	0.38
15	15	2	0.78	0.24	1.11	0.19	0.89	0.21
15	15	2	1.65	0.48	1.67	0.30	1.73	0.30
15	15	4	1.97	0.75	2.04	0.45	2.07	0.48
15	15	4	2.12	0.64	1.88	0.39	2.08	0.44
15	15	4	2.15	0.93	2.02	0.49	2.37	0.57
15	15	6	2.30	0.87	2.28	0.56	2.69	0.80
15	15	6	2.68	0.94	2.51	0.64	2.79	0.77
15	15	6	2.17	0.78	2.46	0.62	2.86	0.73
20	15	2	1.70	0.49	1.75	0.39	1.78	0.34
20	15	2	1.99	0.57	1.89	0.43	1.95	0.41
20	15	2	2.44	0.65	2.41	0.49	2.42	0.53
20	15	4	3.16	0.94	3.17	0.65	3.63	0.72
20	15	4	2.89	0.95	2.84	0.71	2.97	0.83
20	15	4	2.89	0.85	2.67	0.58	2.90	0.61
20	15	6	2.89	1.23	2.61	0.97	3.03	1.07
20	15	6	3.44	0.95	3.14	0.94	3.74	1.06
20	15	6	3.29	0.91	3.36	0.75	3.47	0.85
20	20	2	3.04	0.95	2.90	0.72	3.35	0.71
20	20	2	3.29	0.98	3.26	0.78	3.61	0.75
20	20	2	3.76	0.93	3.43	0.76	4.05	0.74
20	20	4	3.74	1.37	3.97	1.16	4.87	1.37
20	20	4	3.54	1.50	3.37	1.30	3.82	1.38
20	20	4	3.59	1.37	3.88	1.17	4.52	1.32
20	20	6	5.15	1.79	5.06	1.58	5.55	1.60
20	20	6	3.76	1.52	3.90	1.39	4.84	1.69
20	20	6	5.52	2.00	5.23	1.52	5.17	1.85
20	25	2	3.93	1.38	3.86	1.05	4.24	1.09
20	25	2	3.99	1.78	4.30	1.37	4.53	1.28
20	25	2	3.92	1.64	4.02	1.38	4.26	1.36
20	25	4	4.71	1.77	5.02	1.61	5.60	1.65
20	25	4	5.15	1.74	4.83	1.61	5.90	1.69
20	25	4	4.39	1.71	4.81	1.54	4.64	1.57
20	25	6	5.45	1.83	5.60	2.29	6.80	2.38
20	25	6	6.05	2.24	6.48	1.92	7.17	2.10
20	25	6	5.00	2.23	6.02	2.24	6.66	2.04

Table 8: Linear programming relaxation gaps for MILP formulations of MINLP2 with and without strengthening while solving 36 large instances of PP-MINLP2.

Problem size			LP Relaxation Gaps(%)					
$\|\mathcal{I}\|$	$\|\mathcal{T}\|$	$\|\mathcal{P}\|$	PLA-SS	PLA	SEC-S	SEC	k-SEC-SS	k-SEC
15	15	2	21.7	1.44	22.0	1.50	22.4	1.48
15	15	2	18.9	1.22	19.0	1.26	19.2	1.27
15	15	2	21.6	1.43	21.7	1.43	22.2	1.39
15	15	4	20.6	1.58	20.9	1.60	21.3	1.56
15	15	4	20.5	1.45	20.7	1.43	21.4	1.47
15	15	4	21.8	1.64	22.1	1.64	22.7	1.66
15	15	6	23.3	1.64	23.3	1.66	23.7	1.82
15	15	6	23.2	1.69	23.3	1.63	23.7	1.74
15	15	6	22.6	1.68	22.6	1.67	22.8	1.76
20	15	2	20.7	1.09	20.8	1.12	21.1	1.08
20	15	2	20.5	1.40	20.5	1.39	20.9	1.37
20	15	2	22.1	1.28	22.2	1.26	22.6	1.25
20	15	4	22.5	1.44	22.8	1.39	23.2	1.43
20	15	4	22.5	1.45	22.8	1.47	23.4	1.48
20	15	4	21.6	1.43	21.7	1.38	22.2	1.39
20	15	6	21.0	1.69	21.2	1.69	21.7	1.72
20	15	6	23.4	1.42	23.4	1.48	23.6	1.59
20	15	6	24.0	1.35	23.9	1.37	24.1	1.46
20	20	2	20.3	1.55	20.5	1.55	21.0	1.54
20	20	2	22.5	1.66	22.5	1.66	23.0	1.66
20	20	2	22.6	1.69	22.8	1.76	23.3	1.77
20	20	4	22.3	1.88	22.6	1.91	23.1	2.03
20	20	4	21.0	2.29	21.2	2.30	22.1	2.41
20	20	4	21.7	1.97	21.9	1.98	22.6	2.14
20	20	6	23.4	2.18	23.6	2.26	24.1	2.28
20	20	6	22.4	2.08	22.5	2.09	23.3	2.32
20	20	6	24.2	2.46	24.2	2.29	25.1	2.61
20	25	2	21.2	2.07	21.3	2.11	21.6	2.10
20	25	2	21.1	2.78	21.1	2.66	21.2	2.60
20	25	2	21.9	2.40	22.2	2.40	22.6	2.42
20	25	4	22.0	2.24	22.4	2.34	22.9	2.45
20	25	4	21.9	2.29	22.2	2.42	22.9	2.47
20	25	4	20.1	2.39	20.3	2.41	20.9	2.43
20	25	6	22.0	2.35	22.3	2.64	23.2	2.97
20	25	6	23.3	2.63	23.3	2.66	23.7	2.86
20	25	6	23.3	2.73	23.6	2.68	24.1	2.79

Table 9: Best found feasible solution statistics for MILP formulations of MINLP2 while solving 36 large instances of PP-MINLP2.

Problem size			Gap to best bound of PP-MINLP2 (%)		
$\|\mathcal{I}\|$	$\|\mathcal{T}\|$	$\|\mathcal{P}\|$	PLA-SS	SEC-S	k-SEC-SS
15	15	2	1.08	1.09	1.14
15	15	2	0.90	0.87	0.96
15	15	2	1.11	1.05	1.16
15	15	4	1.01	1.24	1.12
15	15	4	1.05	0.92	1.07
15	15	4	1.11	1.15	1.24
15	15	6	1.91	1.79	2.17
15	15	6	1.69	1.67	1.79
15	15	6	1.96	2.03	1.98
20	15	2	0.96	0.98	0.99
20	15	2	1.17	1.21	1.22
20	15	2	1.35	1.29	1.42
20	15	4	1.27	1.31	1.47
20	15	4	1.40	1.35	1.41
20	15	4	1.41	1.49	1.45
20	15	6	2.17	2.14	2.28
20	15	6	1.82	1.95	2.20
20	15	6	2.09	2.09	2.22
20	20	2	1.28	1.23	1.32
20	20	2	1.54	1.61	1.66
20	20	2	1.53	1.50	1.77
20	20	4	1.88	1.98	2.12
20	20	4	1.95	2.01	2.00
20	20	4	1.89	1.95	2.09
20	20	6	2.46	2.52	2.89
20	20	6	2.89	2.89	3.22
20	20	6	2.52	2.40	2.72
20	25	2	1.75	1.75	1.98
20	25	2	2.16	2.11	2.07
20	25	2	2.10	2.25	2.41
20	25	4	2.15	2.20	2.38
20	25	4	2.19	2.57	2.45
20	25	4	2.18	2.37	2.39
20	25	6	2.68	3.45	3.39
20	25	6	3.83	3.82	3.95
20	25	6	3.31	3.38	3.49

Table 10: Solution bound statistics for MILP relaxations of MINLP2 while solving 36 large instances of PP-MINLP2.

Problem size			Gap to best feasible solution of PP-MINLP2 (%)	
$\lvert\mathcal{I}\rvert$	$\lvert\mathcal{T}\rvert$	$\lvert\mathcal{P}\rvert$	SEC-S	k-SEC-SS
15	15	2	1.12	1.08
15	15	2	0.87	0.89
15	15	2	1.07	1.05
15	15	4	1.01	1.47
15	15	4	0.92	1.13
15	15	4	1.11	1.37
15	15	6	1.79	2.66
15	15	6	1.67	2.32
15	15	6	1.96	2.75
20	15	2	0.99	0.96
20	15	2	1.19	1.17
20	15	2	1.29	1.30
20	15	4	1.27	1.47
20	15	4	1.35	1.60
20	15	4	1.41	1.69
20	15	6	2.14	2.53
20	15	6	1.82	2.82
20	15	6	2.09	2.99
20	20	2	1.27	1.23
20	20	2	1.59	1.54
20	20	2	1.54	1.50
20	20	4	1.88	2.57
20	20	4	1.95	2.12
20	20	4	1.89	2.30
20	20	6	2.46	2.78
20	20	6	2.89	3.28
20	20	6	2.40	2.58
20	25	2	1.75	1.81
20	25	2	2.07	2.08
20	25	2	2.12	2.10
20	25	4	2.15	2.65
20	25	4	2.19	2.59
20	25	4	2.18	2.39
20	25	6	2.68	3.28
20	25	6	3.82	4.45
20	25	6	3.31	3.54