

Optimizing Fed-Batch Processes with Dynamic Control Flux Balance Analysis

Mathias Gotsmy^{*,**} Dafni Giannari^{***}
Radhakrishnan Mahadevan^{***} Jürgen Zanghellini^{**}

^{*} Austrian Centre of Industrial Biotechnology, Austria.

^{**} Department of Analytical Chemistry, University of Vienna, Austria

^{***} Department of Chemical Engineering and Applied Chemistry,
University of Toronto, Canada

Abstract: Fed-batch processes are prevalent in biotechnological industries, but design of experiments often results in sub-optimal conditions due to incomplete solution space characterization. We employ a single-level dynamic control (DC) algorithm for dynamic flux balance analysis (dFBA), enhancing efficiency by reducing Karush-Kuhn-Tucker (KKT) condition constraints and adapting the algorithm for predicting optimal process length. In a growth-decoupled plasmid DNA production case study, we predict the optimal feeding profile and switching time between growth and production phase. Comparing our algorithm to its predecessor shows a speed-up of at least a factor of four. When the process length is part of the objective function the speed-up becomes considerably larger.

Copyright © 2024 The Authors. This is an open access article under the CC BY-NC-ND license (<https://creativecommons.org/licenses/by-nc-nd/4.0/>)

Keywords: bioprocess control, bi-level optimization, dynamic optimization, moving finite elements

1. INTRODUCTION

Contrary, to the often cumbersome and expensive design of experiment studies, computer simulations of dFBA-based process models are a cheap and comparatively easy way to systematically sample the solution space for optimal fed-batch process designs [Gotsmy et al., 2023]. They can provide suggestions for setting control variables, for example, the feed rate [Espinell-Ríos et al., 2024] or the switch times of two-stage processes [Raj et al., 2020, Bauer and Klamt, 2024].

Recently, more sophisticated process optimization algorithms have been developed [Nakama and Jäschke, 2022, de Oliveira et al., 2023]. Most importantly, they employ KKT condition constraints and collocation techniques to translate the dFBA process models into a non-linear programming (NLP) problem which is solved with interior-point methods. Unfortunately, these algorithms come with a new set of challenges, especially their sensitivity to hyperparameters and improvable convergence.

Therefore, in this study, we adapt the dynamic control flux balance analysis (dcFBA) algorithm from de Oliveira et al. [2023]. Specifically, we reduce the number of KKT condition constraints and reformulate the finite element length constraints. These adaptations improve the general convergence speed as well as performance for (fed-batch) process length optimization.

2. THEORY

Here, we highlight the basic principles of our dcFBA implementation. For clarity, Table 1 lists the dimensions of vectors and matrices used throughout the section.

2.1 Standard dynamic flux balance analysis

Fed-batch process simulations involve estimating state variables, such as biomass (X), substrate (e.g. glucose, G), product (P), and volume (V) from an initial state at a time t_0 throughout the process length (T). Differential equations for these state variables typically read

$$\dot{X}(t) = \mu(t)X(t), \quad X(t_0) = X^0, \quad (1a)$$

$$\dot{P}(t) = \pi(t)X(t), \quad P(t_0) = 0, \quad (1b)$$

$$\dot{G}(t) = \phi(t)g_\phi - \gamma(t)X(t), \quad G(t_0) = G^0, \quad (1c)$$

$$\dot{V}(t) = \phi(t), \quad V(t_0) = V^0. \quad (1d)$$

Here, \dot{X} , \dot{P} , and \dot{G} represent the rates of change for biomass, product, and substrate, respectively. The parameters μ , π , and γ denote the specific growth rate, production rate, and substrate uptake rate, while ϕ represents the feed rate, and g_ϕ is the substrate concentration in the feed medium. For simplicity, we will assume $P(t^0) = t^0 = 0$. Additionally, for any time-dependent variable x at time t^j , we adopt the shorthand notation $x(t^j) = x^j$.

In biotechnological production, engineers are typically interested in the optimization of key process performance metrics such as the final product titer

$$\max \mathcal{T} := P(T) / V(T), \quad (2a)$$

the average productivity

$$\max \mathcal{P} := P(T) / T, \quad (2b)$$

and the average product yield

$$\max \mathcal{Y} := P(T) / \Delta G, \quad (2c)$$

or combinations thereof [Zhuang et al., 2013], subject to the response of the cellular production host. Here, ΔG denotes the total substrate consumed.

Table 1. **Dimensions of vectors/matrices.** $n_R = n_{\text{ABCD}}$, number of reactions; n_M , number of metabolites; n_{FE} , number of finite elements.

Variable	Shape	Variable	Shape
\mathbf{S}	(n_M, n_R)	$\alpha_{\text{lb}}, \mathbf{s}_{\text{lb}}$	$(1, n_{\text{AB}})$
\mathbf{v}	$(n_R, 1)$	$\alpha_{\text{ub}}, \mathbf{s}_{\text{ub}}$	$(1, n_{\text{AC}})$
\mathbf{d}	$(1, n_R)$	$\tau, \hat{\tau}$	$(n_{\text{FE}}, 1)$
λ	$(1, n_M)$		

Fed-batch production processes typically have several control variables (\mathbf{u}) that can be set by the process engineer. Control variables, for example, comprise the feed rate [Dietzsch et al., 2011], the initial values of state variables [Gotsmy et al., 2023], or oxygen sparging and stirrer speed [Erian et al., 2018]. Thus, the DC problem asks: what values of \mathbf{u} give an optimal process?

In dFBA, the host's response to \mathbf{u} is often approximated through parsimonious flux balance analysis (pFBA):

$$\min_{\mathbf{v}^j} \mathbf{d}\mathbf{v}^j + \varepsilon/2 \mathbf{v}^j \cdot \mathbf{v}^j \quad (3a)$$

$$\text{s.t. } \mathbf{S}\mathbf{v}^j = \mathbf{0} \quad (3b)$$

$$\mathbf{v}_{\text{lb}}(\mathbf{u}^j) \leq \mathbf{v}^j \leq \mathbf{v}_{\text{ub}}(\mathbf{u}^j). \quad (3c)$$

Here, \mathbf{S} denotes the net stoichiometric matrix of the cell, \mathbf{v}^j represents the cellular flux distribution, and $\mathbf{v}_{\text{lb}}(\mathbf{u}^j)$ and $\mathbf{v}_{\text{ub}}(\mathbf{u}^j)$ are the lower and upper bounds of the flux, respectively. The scaling parameter ε is typically assigned a small value [Ploch et al., 2020, Nakama and Jäschke, 2022, de Oliveira et al., 2023].

The host's influence on the medium is evaluated in n_{FE} consecutive intervals τ^j , and then aggregated across the entire process duration T :

$$X(T) = X^0 + \sum_{j=1}^{n_{\text{FE}}} \Delta X^j, \quad (4a)$$

$$P(T) = \sum_{j=1}^{n_{\text{FE}}} \pi^j \Delta X^j / \mu^j, \quad (4b)$$

$$G(T) = G^0 + g_\phi \sum_{j=1}^{n_{\text{FE}}} \phi^j \tau^j - \sum_{j=1}^{n_{\text{FE}}} \gamma^j \Delta X^j / \mu^j, \quad (4c)$$

with

$$\Delta X^j := [\exp(\mu^j \tau^j) - 1] x^{j-1}, \quad (4d)$$

$$\tau^j := t^j - t^{j-1}, \quad \sum_{i=1}^{n_{\text{FE}}} \tau^j = T. \quad (4e)$$

The changes in the amounts (ΔX^j and the summands in (4b) and (4c)) result from constant cellular growth with μ^j during the interval τ^j . μ^j , π^j , and γ^j are a subset of the elements in the cellular flux vector \mathbf{v}^j .

Process (in)equality constraints depending on any system's variable may be added to the optimization problem,

$$\mathbf{a}(\cdot) = \mathbf{0}, \quad \mathbf{b}(\cdot) \leq \mathbf{0}. \quad (5)$$

They typically comprise physical limitations (e.g., non-negativity of concentrations), technical limitations (e.g., lower and upper bound of the reactor volume or aeration rates), or biological limitations (e.g., maximal achievable biomass concentrations).

The system of equations (3), (4), and (5), combined with one of the objectives in (2), presents a bilevel optimization problem, which is generally challenging to solve [de Oliveira et al., 2023] even for state-of-the-art algorithms like DFBAlab [Gomez et al., 2014].

2.2 Dynamic control FBA

dcFBA addresses this issue by transforming the bi-level optimization problem into a single non-linear optimization problem,

$$\max_{\mathbf{u}} F(\cdot) \quad (6a)$$

subject to

$$\mathbf{S}\mathbf{v}^j = \mathbf{0} \quad (6b)$$

$$\mathbf{v}_{\text{lb}}(\mathbf{u}^j) \leq \mathbf{v}^j \leq \mathbf{v}_{\text{ub}}(\mathbf{u}^j) \quad (6c)$$

$$\nabla_{\mathbf{v}^j} \mathcal{L}^j(\mathbf{v}^j, \lambda^j, \alpha_{\text{lb}}^j, \alpha_{\text{ub}}^j) = \mathbf{0} \quad (6d)$$

$$\mathbf{s}_{\text{lb}}^j(\mathbf{v}^j, \mathbf{v}_{\text{lb}}^j, \alpha_{\text{lb}}^j) = \mathbf{s}_{\text{ub}}^j(\mathbf{v}^j, \mathbf{v}_{\text{ub}}^j, \alpha_{\text{ub}}^j) = \mathbf{0} \quad (6e)$$

$$\alpha_{\text{lb}}^j, \alpha_{\text{ub}}^j \geq \mathbf{0}, \quad j \in \{1, \dots, n_{\text{FE}}\} \quad (6f)$$

where the optimal solution $\bar{\mathbf{v}}^j = \max \mathbf{d}\mathbf{v}^j + \varepsilon/2 \mathbf{v}^j \cdot \mathbf{v}^j$ of the inner flux balance analysis (FBA) optimization in (3) satisfies the KKT conditions (6d)-(6f) [de Oliveira et al., 2023] with the Lagrangian

$$\mathcal{L}^j(\mathbf{v}^j, \lambda^j, \alpha_{\text{lb}}^j, \alpha_{\text{ub}}^j) = \mathbf{d}\mathbf{v}^j + \varepsilon/2 \mathbf{v}^j \cdot \mathbf{v}^j + \lambda^j \mathbf{S}\mathbf{v}^j + \alpha_{\text{lb}}^j (\mathbf{v}_{\text{lb}}^j - \mathbf{v}^j) + \alpha_{\text{ub}}^j (\mathbf{v}^j - \mathbf{v}_{\text{ub}}^j) \quad (7a)$$

and

$$\mathbf{s}_{\text{lb}}^j := \text{diag}(\alpha_{\text{lb}}^j)(\mathbf{v}_{\text{lb}}^j - \mathbf{v}^j) = \mathbf{0}, \quad (7b)$$

$$\mathbf{s}_{\text{ub}}^j := \text{diag}(\alpha_{\text{ub}}^j)(\mathbf{v}^j - \mathbf{v}_{\text{ub}}^j) = \mathbf{0}. \quad (7c)$$

For numerical reasons, only KKT stationary conditions (6d) and KKT nonnegativity conditions (6f) are implemented as constraints. The KKT complementary slackness conditions (6e) are incorporated into the optimization function (6a) as a penalty term (\mathcal{S}) [de Oliveira et al., 2023]:

$$\max_{\mathbf{u}} \psi_1 F(\cdot) + \psi_2 \mathcal{S} \quad \text{s.t. (6b)-(6f)} \quad (8a)$$

where

$$\mathcal{S} := \frac{1}{n_{\text{FE}} n_R} \sum_j \sum_i \left(\mathbf{s}_{\text{lb}}^j + \mathbf{s}_{\text{ub}}^j \right)_i. \quad (8b)$$

\mathcal{S} converges to 0 from negative values, therefore, \mathcal{S} is maximized in the objective function. To achieve a balanced contribution of both terms, F and \mathcal{S} , to the objective function, they are weighted by the hyperparameters ψ_1 and ψ_2 .

Including \mathcal{S} as a penalty term has shown to overcome linear independence constraint qualification (LICQ) violations similar to the definition of the pFBA (Equation 3) [Baumrucker et al., 2008]. In accordance with previous studies, we assume that the LICQ holds for dcFBA [de Oliveira et al., 2023], however, a detailed analysis will be the focus of future work.

The single-level dcFBA optimization reformulation significantly improves optimization efficiency and thus renders more complex process optimization problems solvable [de Oliveira et al., 2023].

Our study is based on the work done by de Oliveira et al. [2023] and more detailed description of their dcFBA can be

found there. Although their algorithm works well for short bioprocesses (i.e., $T \leq 10$ h), its convergence becomes problematic for longer bioprocess, typically fed-batches, (i.e., $T \geq 30$ h) if the temporal resolution (i.e. n_{FE}/T) is kept constant. Additionally, their implementation of finite element length bounds is impractical for process length optimization.

In the following sections, we focus on our adaptations to the algorithm. Henceforth, to increase the readability, we omit the index j from constraints when it is obvious.

2.3 Reducing KKT Condition Constraints

Previously, the KKT condition constraints of all fluxes of the metabolic model were included in the dcFBA algorithm [de Oliveira et al., 2023]. However, in metabolic modeling, for many fluxes no realistic constraints are known, usually indicated by lower bounds of -1000 and upper bounds of +1000 $\text{mmol g}^{-1} \text{h}^{-1}$. The reactions of a metabolic model can be sorted into four classes, **A** both bounds are known, **B** only lower bounds are known, **C** only upper bounds are known, and **D** no bounds are known.

Here we reduce the number of process optimization constraints in our dcFBA algorithm by only including KKT condition constraints for reaction bounds that are known.

Taking the partial derivative gives

$$\frac{\partial \mathcal{L}}{\partial \mathbf{v}^{\text{A}}} = \mathbf{d}^{\text{A}} + \varepsilon(\mathbf{v}^{\text{A}})^{\text{T}} + \lambda \mathbf{S}^{\text{A}} - \alpha_{\text{lb}}^{\text{A}} + \alpha_{\text{ub}}^{\text{A}} = \mathbf{0} \quad (9\text{a})$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{v}^{\text{B}}} = \mathbf{d}^{\text{B}} + \varepsilon(\mathbf{v}^{\text{B}})^{\text{T}} + \lambda \mathbf{S}^{\text{B}} - \alpha_{\text{lb}}^{\text{B}} = \mathbf{0} \quad (9\text{b})$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{v}^{\text{C}}} = \mathbf{d}^{\text{C}} + \varepsilon(\mathbf{v}^{\text{C}})^{\text{T}} + \lambda \mathbf{S}^{\text{C}} + \alpha_{\text{ub}}^{\text{C}} = \mathbf{0} \quad (9\text{c})$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{v}^{\text{D}}} = \mathbf{d}^{\text{D}} + \varepsilon(\mathbf{v}^{\text{D}})^{\text{T}} + \lambda \mathbf{S}^{\text{D}} = \mathbf{0} \quad (9\text{d})$$

This method reduces the number of α -variables from $2n_{\text{R}}$ to $2n_{\text{A}} + n_{\text{B}} + n_{\text{C}}$.

2.4 Rescaling Moving Finite Elements

In the previous approach, the lengths τ of finite element (FE) were not fixed but constrained by absolute lower and upper bounds. Yet, their sum added up to the total process length (T),

$$\tau_{\text{lb}} \leq \tau^j \leq \tau_{\text{ub}}, \quad T = \sum_j^{n_{\text{FE}}} \tau^j. \quad (10)$$

Setting upper and lower bounds on FE lengths serves two purposes: (1) enabling the optimization of total process length, and (2) managing sudden and strong rate changes during optimization [de Oliveira et al., 2023]. For biotechnological process optimization, both functionalities are important. Yet, relaxing the bounds on τ^j can lead to poor convergence, numerical issues, and even local infeasibility of the solver (see Section 4). Therefore, we limit the relative deviation of each FE from the average length T/n_{FE} ,

$$\frac{\hat{\tau}_{\text{lb}}^j T}{n_{\text{FE}}} \leq \tau^j \leq \frac{\hat{\tau}_{\text{ub}}^j T}{n_{\text{FE}}}, \quad T = \sum_j^{n_{\text{FE}}} \tau^j, \quad (11)$$

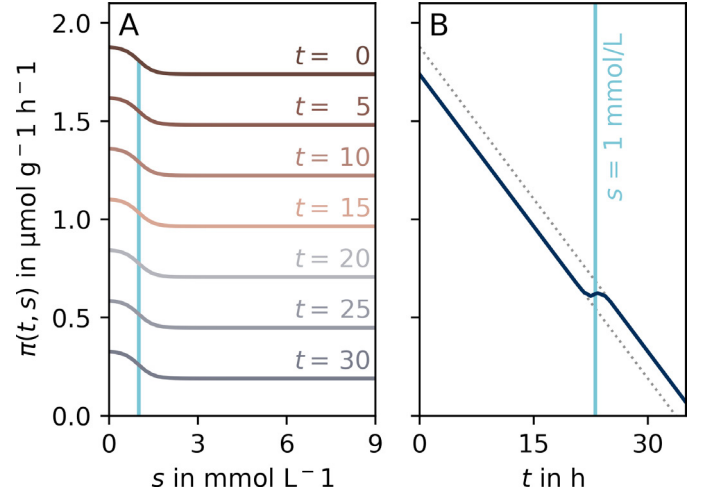


Fig. 1. **Production envelope of pDNA production** [Gotsmy et al., 2023]. $\pi(t, s)$ is linearly dependent on fed-batch time (t) and increases by a step function during sulfate (S) starvation. **A** $\pi(t, s)$ at different constant process times t . **B** $\pi(t, s)$ of an example process where $s = S/V = 1 \text{ mmol L}^{-1}$ is reached after $t = 23.1$ h. The dotted lines indicate $\pi(t)$ before and during S limitation.

rather than the absolute length as in (10). Therefore, in our implementation, lower and upper bounds for T and $\hat{\tau}$ can be set separately for reasons (1) and (2), respectively. Our adaptation increases the number of unknown variables for optimization by one (i.e., the value for T).

2.5 Adapted Dynamic control FBA

Finally, the derived constraints are integrated into our adapted dcFBA optimization problem,

$$\max_{\mathbf{u}, \tau, T} \psi_1 F(\cdot) + \psi_2 S \quad (12\text{a})$$

subject to FBA constraints

$$\mathbf{Sv} = \mathbf{0} \quad (12\text{b})$$

$$\mathbf{v}_{\text{lb}} \leq \mathbf{v} \leq \mathbf{v}_{\text{ub}} \quad (12\text{c})$$

subject to KKT condition constraints

$$\nabla_{\mathbf{v}} \mathcal{L} = \mathbf{0} \quad (12\text{d})$$

$$\alpha_{\text{lb}}, \alpha_{\text{ub}} \geq \mathbf{0} \quad (12\text{e})$$

subject to moving FE constraints

$$\hat{\tau}_{\text{lb}} T/n_{\text{FE}} \leq \tau \leq \hat{\tau}_{\text{ub}} T/n_{\text{FE}} \quad (12\text{f})$$

$$\mathbf{1}^{\text{T}} \tau = T \quad (12\text{g})$$

$$T_{\text{lb}} \leq T \leq T_{\text{ub}} \quad (12\text{h})$$

and subject to custom process (in)equality constraints

$$\mathbf{a}(\cdot) = \mathbf{0}, \quad \mathbf{b}(\cdot) \leq \mathbf{0}. \quad (12\text{i})$$

The KKT complementary slackness condition penalty term incorporated into the optimization function (12a) now reads:

$$S := \frac{1}{n_{\text{FE}}} \sum_j^{n_{\text{FE}}} \left[\frac{1}{n_{\text{AB}}} \sum_i^{n_{\text{AB}}} (s_{\text{lb}}^j)_i + \frac{1}{n_{\text{AC}}} \sum_i^{n_{\text{AC}}} (s_{\text{ub}}^j)_i \right] \quad (12\text{j})$$

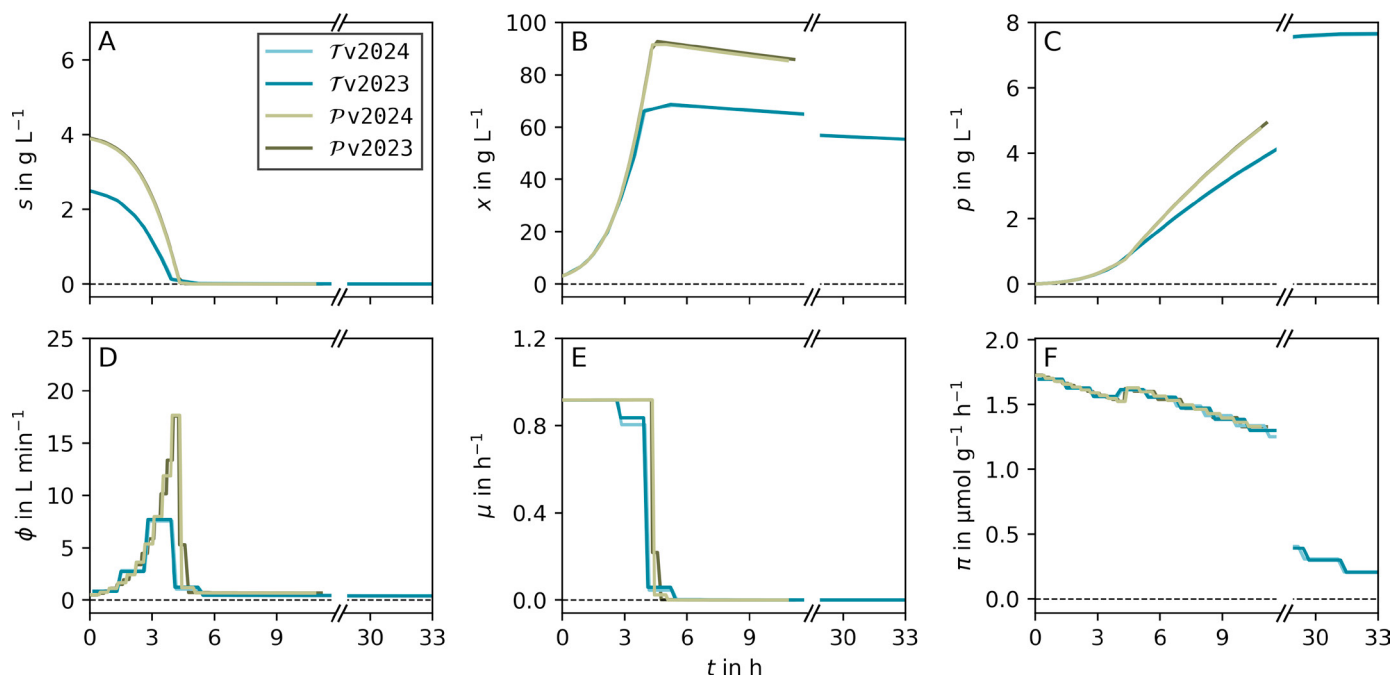


Fig. 2. **Comparison of the plasmid DNA (pDNA) production processes** optimizing the titer (\mathcal{T} , blue) and the productivity (\mathcal{P} , green) by v2023 (darker shade) and v2024 (lighter shade), respectively. The top panels show integrated process state variables (sulfate s , biomass x , product p), and the bottom panels show rates (feed rate ϕ , growth rate μ and product synthesis rate π). The initial concentration of sulfate (panel A) and the feed rate (panel D) comprise the control variables (\mathbf{u}) of the DC problem (Equation 12). For maximization of the titer both algorithms give the same result, however, there is a small difference for the maximization of productivity.

where n_{FE} , n_{AB} , and n_{AC} are used to normalize for the number of FEs and the number of lower and upper bound complementary slackness conditions, respectively.

3. METHODS

3.1 Plasmid DNA Production

In a prior study, we found that limiting sulfate in the reactor medium uncouples plasmid DNA (pDNA) production from biomass growth [Gotsmy et al., 2023]. Using dFBA, we designed and validated an optimal pDNA production process. Simulations were performed with the *E. coli* model *iML1515* [Monk et al., 2017] where pDNA synthesis was added expanding the model's size to $n_{\text{R}} = 2714$, $n_{\text{M}} = 1878$.

To reduce the size of the genome-scale metabolic model (GSMM), we reran (classical) dFBA simulations for control and sulfate-limited processes from our prior study. At each evaluation time point, a lexicographic FBA was calculated followed by a pFBA. Reactions with no flux in any evaluation step were removed, resulting in a condensed GSMM with a size of $n_{\text{R}} = 470$ and $n_{\text{M}} = 454$.

Analysis of the experimental data [Gotsmy et al., 2023] showed that the pDNA synthesis rate (π) linearly decreased with process time and increased step-wise upon sulfate (S) limitation (Figure 1). For numerical reasons and differentiability, the step function was implemented as a $\tanh(\cdot)$ with midpoint at $s = S/V = 1 \text{ mmol L}^{-1}$.

3.2 Implementation of the Algorithm

The dcFBA algorithm was implemented following de Oliveira et al. [2023] using the Julia package JuMP [Lubin et al., 2023] and the nonlinear IPOPT solver [Wächter and Biegler, 2006]. Unless stated otherwise, all simulations used $n_{\text{FE}} = 20$ FEs and the IPOPT solver options were set as follows: "tol" => $1e-4$, "acceptable_iter" => 15, and "acceptable_tol" => $1e-2$.

The complete code for this study is available at <https://github.com/Gotsmy/dcfba>.

4. RESULTS

In the following, we compare our adapted dcFBA implementation (referred to as v2024) with the previously published version [de Oliveira et al., 2023] (referred to as v2023). We conducted optimization for two pDNA production processes, aiming for maximal titer and maximal productivity using both algorithms.

Table 2 provides an overview of the algorithmic differences and optimization results for all four simulations. Columns n_{A} to n_{D} show the number of reactions per constraint class, as explained in the Methods section. In the v2023 algorithm, all reactions are treated as fully constrained, despite their bounds being set to biologically unrealistic $\pm 1000 \text{ mmol g}^{-1} \text{ h}^{-1}$, placing them in class A. The number of dcFBA Lagrange multipliers (i.e., λ , α_{lb} , α_{ub}) per simulation is calculated as

$$n_{\text{KKT}} = n_{\text{FE}}(2n_{\text{A}} + n_{\text{B}} + n_{\text{C}} + n_{\text{M}}). \quad (13)$$

Table 2. **Algorithmic Comparison** of a previously published dcFBA algorithm ([de Oliveira et al., 2023], v2023) and our new method (v2024). We tested the optimization of titer (\mathcal{T}) and productivity (\mathcal{P}) in our pDNA production case study.

Objective (F)	Algorithm	n_A	n_B	n_C	n_D	n_{KKT}	Runtime [min]	\mathcal{T} [g L^{-1}]	\mathcal{P} [g h^{-1}]
\mathcal{T}	v2024	1	290	0	179	14920	7.6	7.64	0.22
\mathcal{T}	v2023	470	0	0	0	27880	29.2	7.66	0.22
\mathcal{P}	v2024	1	290	0	179	14920	3.3	4.75	0.42
\mathcal{P}	v2023	470	0	0	0	27880	36.2	4.92	0.42

The corresponding column in Table 2 reveals that v2024 reduces these multipliers by 54 % compared to v2023 in this case study.

Optimizing the control problem involves optimizing each added multiplier during the simulation, contributing to computational costs. Notably, in the two optimization scenarios for productivity (\mathcal{P}) and titer (\mathcal{T}), both implementations converged to similar values. However, v2024 outperforms v2023, being 11 and 4 times faster in the respective scenarios (see Table 2).

For the maximization of titer (\mathcal{T}), both v2023 and v2024 not only converged to the same values but also identified the same optimal process (Figure 2, light and dark blue line). Moreover, both v2023 and v2024 successfully predicted a sulfate-limited process for optimal productivity (\mathcal{P}) that shows identical characteristics to our previously experimentally implemented process [Gotsmy et al., 2023] (Figure 2, light and dark green line).

To analyze the difference between v2024 and v2023, we investigated variations in the length distributions of FEs

between both versions. Figure 3 compares FE lengths (τ) for v2023 (panel A) and v2024 (panel B) when maximizing \mathcal{P} .

Both versions exhibit a trend to shorter FE lengths initially, with FE becoming longer after growth cessation ($t \geq 4.3$ h). However, in v2023, the average length of FE (1.1 h) after growth cessation is over 3.5 times larger than during the growth phase. This disparity leads to numerical and accuracy challenges, causing poor convergence of the IPOPT solver in v2023. In contrast, v2024, with its ability to independently set (tighter) bounds for T and τ^j , exhibits smaller difference in FE lengths (Figure 3B), effectively avoiding numerical difficulties.

5. DISCUSSION

Designing optimal bioprocess is a grand challenge in industrial applications, often addressed by using dFBA. Optimizing bioprocesses with dFBA introduces computational complexities due to the bi-level nature of the problem. To mitigate this challenge, one strategy involves utilizing the KKT conditions, allowing for a transformation into a single-level but still non-linear optimization problem. Despite this simplification, the optimization remains computationally costly, with the attainment of a feasible solution heavily reliant on the algorithm’s hyperparameters.

Our results underscore these two critical considerations: (i) The necessity to identify and eliminate non-limiting or redundant constraints for accelerated performance. (ii) The importance of tightly controlling the algorithm’s hyperparameters to prevent numerical artifacts.

In standard GSMs, reversible and irreversible reactions are commonly bounded within $[-1000, 1000]$ and $[0, 1000]$ $\text{mmol g}^{-1} \text{h}^{-1}$, respectively, approximating the diffusion limit (private communication). These bounds are mostly active when the solution is otherwise unbounded, serving merely as indicators. This presents numerical challenges by inflating the number of constraints and computational costs, impacting both dFBA and FBA. While the importance of eliminating redundant constraints is recognized [Estinmgasih et al., 2019] and has been partially addressed [Nakama and Jäschke, 2022], the impact of default flux bounds has been overlooked. Our study demonstrates the effectiveness of removing these bounds; however, in rare cases, it may result in unbounded solutions. Utilizing formal methods [Estinmgasih et al., 2019] to identify redundant constraints mitigates this problem. Moreover, we believe that other approaches for dFBA for process optimization (e.g., [Scott et al., 2018]) or even the calculation of a standard FBA can profit from our proposed constraint reduction strategy.

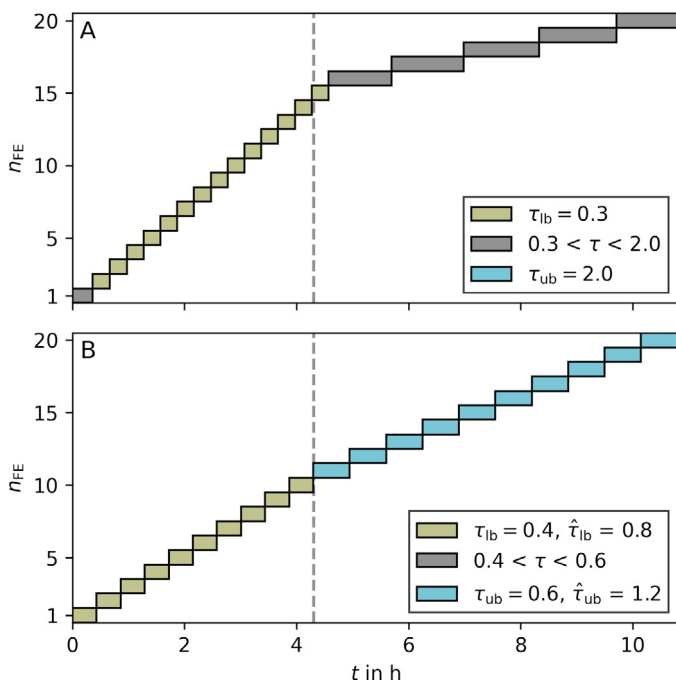


Fig. 3. **Comparison of FE lengths** of $\mathcal{P}_{\text{v2023}}$ (panel A) and $\mathcal{P}_{\text{v2024}}$ (panel B). FEs colored green are equal to the lower length bound, FEs colored in blue are equal to the upper length bound, and grey FEs are in between. The grey dashed line indicates the switch from sulfate excess to sulfate limitation at $t = 4.3$ h.

While the inclusion of moving FEs in the DC problem increases its non-linearity, past cases have shown promising results [Gao et al., 2023]. However, without algorithmic incentives to minimize the discretization errors of the FEs, numerical errors may accumulate. Particularly when FE length bounds are broad due to process length optimization (Figure 3A). To mitigate this risk, setting bounds on relative FE lengths proves effective. This ensures a narrower distribution of FE lengths without overly constraining the total process length. Consequently, our algorithm is well-suited for optimizing bioprocesses where total process time (T) is a part of the objective function.

Recently, an enhanced moving FE method which takes numeric errors into account has been developed [Gao et al., 2023], which will be part of future work.

6. CONCLUSION

Here we present an improved dynamic control flux balance analysis (dcFBA) algorithm. We showcase that strategic reformulation of the problem can decisively improve performance as well as convergence and accuracy compared to previous implementations. Overall, we hope that with careful initialization and selection of hyperparameters, this algorithm will offer a valuable tool for simulating and designing a wide range of bioprocesses.

ACKNOWLEDGEMENTS

Authors acknowledge funding from the Canada Research Chairs Program to Radhakrishnan Mahadevan and the *Carbon-Cycle Economy Demonstration* (C-CED) project funded by the Austrian Climate and Energy Fund, carried out under the program “Vorzeigeregion Energie”, project number: 887638.

REFERENCES

- Bauer, J. and Klamt, S. (2024). Optmsp: A toolbox for designing optimal multi-stage (bio) processes. *Journal of Biotechnology*.
- Baumrucker, B., Renfro, J.G., and Biegler, L.T. (2008). Mpec problem formulations and solution strategies with chemical engineering applications. *Computers & Chemical Engineering*, 32(12), 2903–2913.
- de Oliveira, R.D., Le Roux, G.A., and Mahadevan, R. (2023). Nonlinear programming reformulation of dynamic flux balance analysis models. *Computers & Chemical Engineering*, 170, 108101.
- Dietzsch, C., Spadiut, O., and Herwig, C. (2011). A dynamic method based on the specific substrate uptake rate to set up a feeding strategy for *pichia pastoris*. *Microbial Cell Factories*, 10(1), 1–9.
- Erian, A.M., Gibisch, M., and Pflügl, S. (2018). Engineered *e. coli* enables efficient 2, 3-butanediol production from glucose and sugar beet molasses using defined minimal medium as economic basis. *Microbial Cell Factories*, 17, 1–17.
- Espinel-Ríos, S., Morabito, B., Pohlodek, J., Bettenbrock, K., Klamt, S., and Findeisen, R. (2024). Toward a modeling, optimization, and predictive control framework for fed-batch metabolic cybergenetics. *Biotechnology and Bioengineering*, 121(1), 366–379.
- Estinmgsih, Y., Farikhin, and Tjahjana, R.H. (2019). Some methods for identifying redundant constraints in linear programming. *Journal of Physics: Conference Series*, 1321(2), 022073. doi:10.1088/1742-6596/1321/2/022073. URL <https://dx.doi.org/10.1088/1742-6596/1321/2/022073>. Publisher: IOP Publishing.
- Gao, Y., Wei, Z., Shao, Z., Chen, W., Song, Z., and Biegler, L.T. (2023). Enhanced moving finite element method based on error geometric estimation for simultaneous trajectory optimization. *Automatica*, 147, 110711.
- Gomez, J.A., Höffner, K., and Barton, P.I. (2014). Df-balab: a fast and reliable matlab code for dynamic flux balance analysis. *BMC bioinformatics*, 15, 1–10.
- Gotsmy, M., Strobl, F., Weiß, F., Gruber, P., Kraus, B., Mairhofer, J., and Zanghellini, J. (2023). Sulfate limitation increases specific plasmid dna yield and productivity in *e. coli* fed-batch processes. *Microbial Cell Factories*, 22(1), 242.
- Lubin, M., Dowson, O., Dias Garcia, J., Huchette, J., Legat, B., and Vielma, J.P. (2023). JuMP 1.0: Recent improvements to a modeling language for mathematical optimization. *Mathematical Programming Computation*. doi:10.1007/s12532-023-00239-3.
- Monk, J.M., Lloyd, C.J., Brunk, E., Mih, N., Sastry, A., King, Z., Takeuchi, R., Nomura, W., Zhang, Z., Mori, H., et al. (2017). i ml1515, a knowledgebase that computes *escherichia coli* traits. *Nature biotechnology*, 35(10), 904–908.
- Nakama, C.S. and Jäschke, J. (2022). Analysis of control models based on dfba for fed-batch bioreactors solved by interior-point methods. *IFAC-PapersOnLine*, 55(7), 131–136.
- Ploch, T., von Lieres, E., Wiechert, W., Mitsos, A., and Hannemann-Tamás, R. (2020). Simulation of differential-algebraic equation systems with optimization criteria embedded in modelica. *Computers & Chemical Engineering*, 140, 106920.
- Raj, K., Venayak, N., and Mahadevan, R. (2020). Novel two-stage processes for optimal chemical production in microbes. *Metabolic Engineering*, 62, 186–197.
- Scott, F., Wilson, P., Conejeros, R., and Vassiliadis, V.S. (2018). Simulation and optimization of dynamic flux balance analysis models using an interior point method reformulation. *Computers & Chemical Engineering*, 119, 152–170.
- Wächter, A. and Biegler, L.T. (2006). On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical programming*, 106, 25–57.
- Zhuang, K., Yang, L., Cluett, W.R., and Mahadevan, R. (2013). Dynamic strain scanning optimization: an efficient strain design strategy for balanced yield, titer, and productivity. dyssco strategy for strain design. *BMC biotechnology*, 13, 1–15.