Journal of the Taiwan Institute of Chemical Engineers 114 (2020) 1–11

Contents lists available at ScienceDirect

Journal of the Taiwan Institute of Chemical Engineers

journal homepage: www.elsevier.com/locate/jtice

Dynamic optimization of fluid catalytic cracking unit using a nonconvex sensitivity-based generalized Benders decomposition

Jia-Jiang Lin, Xiong-Lin Luo*, Feng Xu

Department of Automation, China University of Petroleum Beijing, 102249, China

ARTICLE INFO

Article History:
Received 11 May 2020
Revised 24 August 2020
Accepted 8 September 2020
Available online 2 October 2020

Keywords:
Continuous processes with batch operations
Hybrid dynamic optimization
Nonconvex sensitivity-based generalized Benders decomposition
NCO-tracking scheme
Fluid catalytic cracking unit

ABSTRACT

Fluid catalytic cracking unit, whose batch operations are operated in a multirate mode, is a typical continuous process with batch operations. The integrated optimization of this problem can be formulated as a hybrid parametric dynamic optimization. To obtain a high-quality solution, adaptive direct methods are usually required to solve the problem iteratively. By exploiting the decomposable structure, a novel framework is proposed in this paper, which can obtain an equivalent or better precision solution with relatively coarse discretization. In detail, by designating the batch operations as complicating variables, an optimal solution and sensitivity information about batch operations are obtained by a nonconvex sensitivity-based generalized Benders decomposition algorithm. Then the optimal continuous operations are implemented as extra closed-loop controllers by tracking the necessary conditions of optimality, while the optimal batch operations are improved by a line search method. The practical potential of the framework is demonstrated with several operation modes.

1. Introduction

In the field of process systems engineering (PSE), there are increasing demands for better modeling and optimization strategies. Particularly, the integrated optimization of decision making from different levels is a critical technique to further increasing the economic profit [1,2]. Decisions in chemical process can be classified into continuous and batch operations. Physically, continuous operations are the operations that can be adjusted in real-time, while batch operations are only implemented at certain time instants. In general, batch operations just mean that they are one-offs. Furthermore, they can be classified mathematically. If batch operations represent binary or integer numbers, such as planning, scheduling and designing of quantized decisions, then the integrated optimization will formulate a mixed-integer dynamic optimization (MIDO) [3–6]; if the batch operations represent real numbers, such as the adjustment of flow rate and the addition of catalysts, then the integrated optimization will formulate a hybrid dynamic optimization in terms of parameters and continuous variables[7], i.e. hybrid parametric dynamic optimization. In this work, the batch operations represent real numbers, and they are intermittent operations of continuous processes.

As there is an obvious decomposable structure in integrated optimization problem, generalized Benders decomposition (GBD) [8,9] are usually used to expedite the solving of these problems. In detail, the original problem is decomposed into a sequence of subproblems, namely primal and master problems. The solution of these subproblems leads to a sequence of nonincreasing upper bounds (UBD) and a sequence of nondecreasing lower bounds (LBD) which converge to an optimum of the problem. As GBD has many limitations, several efforts have been made to further generalize it. An extra variable and equality constraint are introduced to generate linear master problems [10]. A surrogate model can be constructed for specific nonconvex problem to conduct rigors GBD [11,12]. To apply GBD on a class of general nonconvex problems and sparing the construction of surrogate models, nonconvex sensitivity-based generalized Benders decomposition (NSGBD) algorithm tackles nonconvexity by directly manipulating the consistent linear Benders cuts and checking the optimality conditions [13].

FCCU is one of the most important processes in a refinery to convert low-value heavy oil feedstock into more valuable lighter products, e.g. naphtha, diesel and liquefied petroleum gas. Therefore, the FCCU is highly expected to operate near the optimal operating condition with respect to economics [14–17]. Recently, Huang et al. [18] applied the modifier adaptation for real-time optimization of the FCCU and Guan et al. [19] proposed a control reconfiguration method based on the self-optimizing control (SOC) methodology for the economic operation of FCCU. Usually, continuous operations are operated by regulatory PID controllers to maintain optimal operation points calculated by steady-state optimization, while batch operations are operated in an excessive mode. However, as the batch operations slightly changes the system's dynamic in every cycle, the independent optimization of continuous and batch operations loses economic benefit in every transition period and the integrated

* Corresponding author.
E-mail address: luoxl@cup.edu.cn (X.-L. Luo).

https://doi.org/10.1016/j.jtice.2020.09.017
1876-1070/

optimization of continuous and batch operations will largely improve the overall economic performance [7]. In detail, the batch operations of FCCU include the addition of CO promoter and the adjustment of combustion air flow rate, while CO promoter is a relatively expensive resource and air blower consumes electricity. Moreover, they can be operated in a multirate mode, i.e. implemented under different cycles. To investigate the economic performance of FCCU under different operation modes, several cases have been considered in this paper, such as CO promoter added every 8 h and combustion air flow rate adjusted every 2 h.

The integrated optimization of continuous and batch operations forms a hybrid parametric dynamic optimization problem. Intuitively, the parametrized solution can be obtained by Pontryagin maximum principle [20] or Belleman optimality principle [21] for fixed batch operations, then it leads to a standard steady-state finite-dimensional optimization problem. In fact, as a special kind of hybrid system, its optimality conditions can also be obtained, namely hybrid parametric minimum principle [22]. As most practical problems are too complex to allow for an analytical solution from optimality conditions, the direct methods are favored to solve optimal control problems, which parametrizes the continuous variables by discretization and reformulates the original problem as a nonlinear programming (NLP). As the batch operations can be parametrized by themselves, the direct methods can be directly used to solve hybrid parametric dynamic optimization problem. Moreover, according to the difference in the level of discretization, there are three types of direct methods, i.e., the sequential method [23,24], the simultaneous method [25,26] and multiple-shooting method [27,28].

By direct methods, the continuous and batch control variables are treated equally without discrimination in the reformulated NLP, which has three drawbacks: 1) it fails to distinguish the difference between the continuous and batch operations; 2) it usually obtains a suboptimal solution of original problem; 3) the solution is of the open-loop form. As the implementations of batch operations are one-offs, while the continuous control variables could be adjusted to meet any real disturbance or mismatch, the optimal solution of batch operation is far more valuable. Nevertheless, it is not reflected in the simultaneous parameterization of continuous and batch operations. To obtain a satisfactory solution efficiently, especially when the problem formulation contains large-scale models, e.g. those stemming from industrial applications, the application of adaptive methods [29,30] is usually inevitable, which generate a fully adaptive, problem-dependent parameterization by repetitive solution of increasingly refined finite-dimensional optimization problems. To implement the optimal solution in a close-loop form, the optimal continuous operations can be implemented as extra feedback controllers by tracking the necessary conditions of optimality (NCO-tracking scheme) [31, 32], which maintains near-optimal performance under uncertainty, while the optimal batch operations are directly used [7].

Recently, a novel decomposition algorithm that combines NSGBD with control vector parameterization (CVP) has been developed to solve the hybrid parametric dynamic optimization [33]. It designates batch operations as complicating variables and can be seen as a combination of dynamic optimization on the continuous control variables and sensitivity analysis on the batch control variables. In this paper, a novel implementation framework based on NSGBD is proposed, which implements optimal continuous control variables as close-loop form while exploits the sensitivity information of NSGBD. In detail, optimal continuous control variables are implemented by NCO-tracking and a line search on batch control variables is conducted to improve the solution quality of batch control variables. This framework overcomes the drawbacks stated above and obtains a high-quality solution of batch operations with a relatively coarse discretization against direct methods.

The rest of this paper is organized as follows: In Section 2, a brief introduction of a FCCU with high efficiency regenerator is given, and its batch properties are discussed, i.e. the addition of CO promoter and the adjustment of combustion air flow rate. In Section 3.1, the mathematical description of hybrid parametric dynamic optimization is given, which can be solved and implemented by a tailored algorithm given in Section 3.2 and an implementation framework given in Section 3.3. In Section 4, the mathematical formulation of hybrid parametric dynamic optimization for FCCU is given, and four cases are solved and implemented by the framework proposed in Section 3.3.

2. Batch properties of FCCU

The model adopted in this work is derived from the model of an industrial FCCU with a high-efficiency regenerator as proposed by the authors [7,34–36]. A schematic diagram of the FCCU with high-efficiency regenerator is given in Fig. 1. Here, a short description of model and operations is given.

The preheated raw crude oil reaches riser close to the bottom, and contacts with atomizing steam to atomizing the feed for efficient contacting of the feed and regenerated catalyst, which reactions are taken as a five-lump model. Then the vapors and the catalysts separate rapidly and efficiently in stripper, while product vapors exit the upper cyclones and flow to the main fractionator tower. The purpose of the main fractionator is to desuperheat and recover liquid products from the reactor vapors. The spent catalyst flows into regenerator for recovery, which mainly consists of two parts: a combustor where the gases and the solids are fast-fluidized and a dense bed where the gases and the solids are only bubbling-fluidized. The regenerator has three main functions: 1) it restores catalyst activity by burning off the coke covered in the catalyst; 2) it supplies heat for cracking reactions; 3) it delivers fluidized catalyst to the feed nozzles. Moreover, flue gas and catalyst are separated in the freeboard and flue gas exits the cyclones to a plenum chamber in the top of the regenerator.

The continuous operations of FCCU are normally controlled by PID controllers. In specific, the regulatory PID control system of FCCU is comprised of five controllers, which are the regenerated catalyst slide valve controlling the riser temperature, the spent catalyst slide valve controlling the catalyst inventory in the stripper, the recirculation slide valve controlling the catalyst inventory in the dense bed, the steam injected to the wet gas compressor turbine controlling the reactor pressure and the flue gas slide valve controlling the pressure difference between the reactor and the regenerator. Hence, the continuous operations are the adjusting of valve openings, or the setting of set-points of these controllers. Apart from these operations, FCCU discussed in this paper has two extra batch operations, namely the adjusting of combustion air flow rate and the addition of CO promoter.

The air blower provides sufficient air velocity and pressure to maintain the catalyst bed in a fluidized state while provides oxygen for the combustion in the regenerator. Combustion air flow rate is commonly adjusted through variable inlet guide vanes system [37], which consists of a series of flat plates that can be turned to induce a controlled inlet pre-whirl. The adjustment of combustion air flow

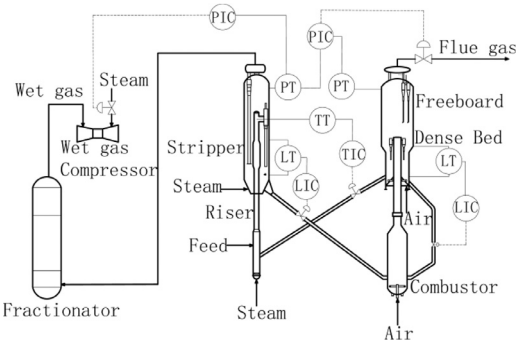

Fig. 1. Schematic diagram of FCCU with high-efficiency regenerator.

rate is realized by tuning the setting angles of vanes. To reduce the mechanical loss and increase the working life of variable inlet guide vanes system, the control of combustion air flow rate might better be intermittent. Hence, the adjustment of combustion air flow rate is a batch operation of the continuous process.

Most FCCU uses CO promoter to assist the combustion of CO to CO_2 in the regenerator and to guarantee the safety of the operation by avoiding afterburning in the freeboard. The amount and frequency of CO promoter additions varies from one FCCU to another. In some units, CO promoter is added to the regenerator two to three times a day. In other FCCU, CO promoter is added only if the temperature rise of freeboard exceeds the operation constraints. In this paper, CO promoter is added manually and periodically in FCCU, which also belongs to a batch operation of the continuous process.

Another special character about multi batch operations is they can be operated in different cycles, i.e. multirate mode. One case studied in this paper is that the addition of CO promoter is implemented every 8 h, while the adjustment of combustion air is taken every 2 h.

3. Hybrid parametric dynamic optimization

3.1. Mathematical formulation of hybrid parametric dynamic optimization

The mechanism model of industrial processes and the constraint equations are the basis of discussing hybrid parametric dynamic optimization. For universality, a process model can be represented by differential/algebraic equations (DAEs):

$$\dot{\boldsymbol{x}}_d = \boldsymbol{f}_d\left(t, \boldsymbol{x}_d(t), \boldsymbol{x}_a(t), \boldsymbol{u}(t), \overline{\boldsymbol{u}}\right) \tag{1a}$$

$$\boldsymbol{0} = \boldsymbol{f}_a\left(t, \boldsymbol{x}_d(t), \boldsymbol{x}_a(t), \boldsymbol{u}(t), \overline{\boldsymbol{u}}\right) \tag{1b}$$

where $\boldsymbol{x}_d(t) \in R^{n_d}/\boldsymbol{x}_a(t) \in R^{n_a}$ represents the vector of differential/algebraic state variables; $\boldsymbol{u}(t)/\overline{\boldsymbol{u}}$ represents the vector of continuous/batch control variables. In this paper, the initial conditions $\boldsymbol{x}_d(t_0)$ are given as the final values of the last period. Obviously, $\overline{\boldsymbol{u}}$ acts as a parameter in this process model.

The objective of a continuous process with batch operations is to maximize the product yield and minimize the cost of feed and operation, which is represented as

$$\min J(\boldsymbol{u}(t), \overline{\boldsymbol{u}}) = \int_{t_0}^{t_f} (-r(t, \boldsymbol{x}_d(t), \boldsymbol{x}_a(t), \boldsymbol{u}, \overline{\boldsymbol{u}}) + c_1(\boldsymbol{t}, \boldsymbol{x}_d(t), \boldsymbol{x}_a(t), \boldsymbol{u}, \overline{\boldsymbol{u}}))dt + c_2(\overline{\boldsymbol{u}}) \tag{2}$$

where cost function J is composed by three parts, $r\left(t, \boldsymbol{x}_d(t), \boldsymbol{x}_a(t), \boldsymbol{u}, \overline{\boldsymbol{u}}\right)$ and $c_1\left(\boldsymbol{t}, \boldsymbol{x}_d(t), \boldsymbol{x}_a(t), \boldsymbol{u}, \overline{\boldsymbol{u}}\right)$ are the product yield and the cost during continuous operation, while $c_2(\overline{\boldsymbol{u}})$ is the cost for batch operations at time instant t_0. An extra differential state variable $\tilde{x}(t)$ can be introduced to simplify the form of J. In detail, let

$$\tilde{x} = -r + c_1 \tag{3}$$

with initial condition

$$\tilde{x}(t_0) = 0 \tag{4}$$

Without loss of generality, the differential state variable could still be represented as $\boldsymbol{x}_d(t)$. Then the cost function J can be simplified as:

$$\min_{\boldsymbol{u}(t), \overline{\boldsymbol{u}}} J\left(\boldsymbol{x}(t_f), \overline{\boldsymbol{u}}\right) = \tilde{x}(t_f) + c_2(\overline{\boldsymbol{u}}) \tag{5}$$

where $\boldsymbol{x}(t) = (\boldsymbol{x}_d(t)^T, \boldsymbol{x}_a(t)^T)^T$.

The path constraints are represented as

$$\boldsymbol{x}^{lb} \le \boldsymbol{x}(t) \le \boldsymbol{x}^{ub} \tag{6a}$$

$$\boldsymbol{u}^{lb} \le \boldsymbol{u}(t) \le \boldsymbol{u}^{ub} \tag{6b}$$

where $\boldsymbol{x}^{lb} = ((\boldsymbol{x}_d^{lb})^T, (\boldsymbol{x}_a^{lb})^T)^T$, $\boldsymbol{x}^{ub} = ((\boldsymbol{x}_d^{ub})^T, (\boldsymbol{x}_a^{ub})^T)^T$, $\boldsymbol{x}_d^{lb}$ and $\boldsymbol{x}_d^{ub}$ are the lower and upper bounds of differential state variables; $\boldsymbol{x}_a^{lb}$ and $\boldsymbol{x}_a^{ub}$ are the lower and upper bounds of algebraic state variables; $\boldsymbol{u}^{lb}$ and $\boldsymbol{u}^{ub}$ are the lower and upper bounds of continuous control variables. Moreover, the constraints of $\overline{\boldsymbol{u}}$ are represented as

$$\overline{\boldsymbol{u}}^{lb} \le \overline{\boldsymbol{u}} \le \overline{\boldsymbol{u}}^{ub} \tag{7}$$

where $\overline{\boldsymbol{u}}^{lb}$ and $\overline{\boldsymbol{u}}^{ub}$ are the lower and upper bounds of batch control variables. The end-point inequality constraints are represented as

$$\boldsymbol{x}^{flb} \le \boldsymbol{x}(t_f) \le \boldsymbol{x}^{fub} \tag{8}$$

where $\boldsymbol{x}^{flb} = ((\boldsymbol{x}_d^{flb})^T, (\boldsymbol{x}_a^{flb})^T)^T$, $\boldsymbol{x}^{fub} = ((\boldsymbol{x}_d^{fub})^T, (\boldsymbol{x}_a^{fub})^T)^T$, $\boldsymbol{x}_d^{flb}$ and $\boldsymbol{x}_d^{fub}$ are the lower and upper bounds of differential state variables at end-point; $\boldsymbol{x}_a^{flb}$ and $\boldsymbol{x}_a^{fub}$ are the lower and upper bounds of algebraic state variables at end-point; The end-point equality constraints could be treated as special instances of end-point inequality constraints when the lower bounds equal upper bounds.

From the above, the optimization of a continuous process with batch operations is described as searching an optimal trajectory of continuous control variables $\boldsymbol{u}(t)$ and an optimal value of batch control variables $\overline{\boldsymbol{u}}$ to minimize Eq. (5), while satisfies Eqs. (1), (6), (7) and (8). Then the hybrid parametric dynamic optimization could be described by Problem (P1).

Problem (P1):

$$\min_{\boldsymbol{u}(t), \overline{\boldsymbol{u}}} J\left(\boldsymbol{x}(t_f), \overline{\boldsymbol{u}}\right) \tag{9a}$$

$$\text{s.t. } \boldsymbol{x}_d(t_0) = \boldsymbol{x}_d^0 \tag{9b}$$

$$\dot{\boldsymbol{x}}_d = \boldsymbol{f}_d\left(t, \boldsymbol{x}_d(t), \boldsymbol{x}_a(t), \boldsymbol{u}(t), \overline{\boldsymbol{u}}\right) \tag{9c}$$

$$\boldsymbol{0} = \boldsymbol{f}_a\left(t, \boldsymbol{x}_d(t), \boldsymbol{x}_a(t), \boldsymbol{u}(t), \overline{\boldsymbol{u}}\right) \tag{9d}$$

$$\boldsymbol{G}_{\overline{u}}(\overline{\boldsymbol{u}}) \le \boldsymbol{0} \tag{9e}$$

$$\boldsymbol{G}_u\left(\boldsymbol{u}(t)\right) \le \boldsymbol{0} \tag{9f}$$

$$\boldsymbol{G}_p\left(\boldsymbol{x}(t)\right) \le \boldsymbol{0} \tag{9g}$$

$$\boldsymbol{G}_e\left(\boldsymbol{x}(t_f)\right) \le \boldsymbol{0} \tag{9h}$$

where

$$\boldsymbol{G}_{\overline{u}}(\overline{\boldsymbol{u}}) = \begin{pmatrix} \overline{\boldsymbol{u}} - \overline{\boldsymbol{u}}^{ub} \\ \overline{\boldsymbol{u}}^{lb} - \overline{\boldsymbol{u}} \end{pmatrix} \tag{9i}$$

$$\boldsymbol{G}_u\left(\boldsymbol{u}(t)\right) = \begin{pmatrix} \boldsymbol{u}(t) - \boldsymbol{u}^{ub} \\ \boldsymbol{u}^{lb} - \boldsymbol{u}(t) \end{pmatrix} \tag{9j}$$

$$\boldsymbol{G}_p\left(\boldsymbol{x}(t)\right) = \begin{pmatrix} \boldsymbol{x}(t) - \boldsymbol{x}^{ub} \\ \boldsymbol{x}^{lb} - \boldsymbol{x}(t) \end{pmatrix} \tag{9k}$$

$$\boldsymbol{G}_e\left(\boldsymbol{x}(t_f)\right) = \begin{pmatrix} \boldsymbol{x}(t_f) - \boldsymbol{x}^{fub} \\ \boldsymbol{x}^{flb} - \boldsymbol{x}(t_f) \end{pmatrix} \tag{9l}$$

As most practical problems are too complex to allow for an analytical solution, the numerical algorithms are inevitable for solving optimal control problems. In practice, the direct methods are favored, and its key idea is transforming the original infinite-dimensional optimization problem into a finite-dimensional NLP by the parameterization of continuous variables. For sequential method, the continuous control variables are approximated in finite dimensional linear spaces, while other continuous variables are calculated by integration. Specifically, for continuous

control variable $\boldsymbol{u}(t) \in D$, the approximation is carried out by

$$\boldsymbol{u}(t) \approx \widehat{\boldsymbol{u}}^N(t) = \sum_{i=1}^{N} \boldsymbol{u}_i^N \phi_i^N(t) \tag{10}$$

where $\phi_i^N(t)$ is the basis of the linear space $D^N = \text{span}\{\phi_1^N(t), \phi_2^N(t), ..., \phi_N^N(t)\}$, and $\boldsymbol{u}_i^N$ is the projection of $\boldsymbol{u}(t)$ at $\phi_i^N(t)$. Usually, the piecewise Lagrange interpolation polynomials are used for $\phi_i^N(t)$, where $\text{m}\left(\text{supp}\phi_i^N(t) \cap \text{supp}\phi_j^N(t)\right) = 0$ (measure) for $\forall i \neq j$. Hence, it has $< \phi_i^N(t), \phi_j^N(t) > = 0$ for $\forall i \neq j$ for D_N. For piecewise constant controls, the optimization horizon $[t_0,t_f]$ is subdivided into $N \geq 1$ control stages, i.e. $t_0 < t_1 < t_2 < \ldots < t_N = t_f$, and the orthogonal basis function $\phi_i^N(t)$ can be represented as

$$\phi_i^N(t) = \begin{cases} 1, t_{i-1} \leq t \leq t_i \\ 0, \text{else} \end{cases} \tag{11}$$

By sequential method, the continuous control variable $\boldsymbol{u}(t)$ is parameterized by $\widehat{\boldsymbol{u}}_N = ((\boldsymbol{u}_1^N)^T, ..., (\boldsymbol{u}_N^N)^T)^T$ and the dynamic optimization is transformed into an NLP. Since path constraints of original problem will be transformed into interior constraints by sequential method solver which only hold in separated points, additional end-point constraints must be employed to prevent the violation of path constraints on the entire time. For example, the variable $y(t)$ has an upper bound y^{ub} and lower bound y^{lb}, after introducing two new variables $y^l(t)$ and $y^u(t)$ with two differential equations $\dot{y}^l = \max\left(0, y^{lb} - y(t)\right)$ and $\dot{y}^u = \max\left(0, y(t) - y^{ub}\right)$, and two initial conditions $y^l(t_0) = 0$ and $y^u(t_0) = 0$, the additional end-point constraints can be written as: $y^l(t_f) = 0$ and $y^u(t_f) = 0$. Moreover, if piecewise constant controls are used for $\boldsymbol{u}(t)$, the path constraints for $\boldsymbol{u}(t)$ could be enforced on $\widehat{\boldsymbol{u}}_N$. At this point, the reformulated NLP is described by Problem (P2).

Problem (P2):

$$\min_{\widehat{\boldsymbol{u}}_N, \overline{\boldsymbol{u}}} J\left(\boldsymbol{x}(t_f), \overline{\boldsymbol{u}}\right) \tag{12a}$$

$$\text{s.t. } M\boldsymbol{X}(t_0) = \boldsymbol{X}_0 \tag{12b}$$

$$M\dot{\boldsymbol{X}}(t) = \boldsymbol{F}\left(t, \boldsymbol{X}(t), \widehat{\boldsymbol{u}}_N, \overline{\boldsymbol{u}}\right) \tag{12c}$$

$$\boldsymbol{G}\left(\overline{\boldsymbol{u}}, \widehat{\boldsymbol{u}}_N, \boldsymbol{X}(t_f)\right) \leq \boldsymbol{0} \tag{12d}$$

where

$$M = \text{diag}(\underbrace{1, ..., 1}_{3n_d+2n_a}, \underbrace{0, ..., 0}_{n_a}) \tag{12e}$$

$$\boldsymbol{X}(t) = \begin{pmatrix} \boldsymbol{x}_d(t) \\ \boldsymbol{x}^b(t) \\ \boldsymbol{x}_a(t) \end{pmatrix} \tag{12f}$$

$$\boldsymbol{X}_0 = \begin{pmatrix} \boldsymbol{x}_d^0 \\ \boldsymbol{0} \\ \boldsymbol{0} \end{pmatrix} \tag{12g}$$

$$\boldsymbol{F}\left(t, \boldsymbol{X}(t), \widehat{\boldsymbol{u}}_N, \overline{\boldsymbol{u}}\right) = \begin{pmatrix} \boldsymbol{f}_d\left(t, \boldsymbol{x}_d(t), \boldsymbol{x}_a(t), \widehat{\boldsymbol{u}}_N, \overline{\boldsymbol{u}}^j\right) \\ \boldsymbol{f}_b\left(t, \boldsymbol{x}(t)\right) \\ \boldsymbol{f}_a\left(t, \boldsymbol{x}_d(t), \boldsymbol{x}_a(t), \widehat{\boldsymbol{u}}_N, \overline{\boldsymbol{u}}^j\right) \end{pmatrix} \tag{12h}$$

$$\boldsymbol{f}_b\left(t, \boldsymbol{x}(t)\right) = \begin{pmatrix} \max\left(\boldsymbol{0}, \boldsymbol{x}^{lb} - \boldsymbol{x}(t)\right) \\ \max\left(\boldsymbol{0}, \boldsymbol{x}(t) - \boldsymbol{x}^{ub}\right) \end{pmatrix} \tag{12i}$$

$$\boldsymbol{G}\left(\overline{\boldsymbol{u}}, \widehat{\boldsymbol{u}}_N, \boldsymbol{X}(t_f)\right) = \begin{pmatrix} \boldsymbol{G}_{\overline{u}}(\overline{\boldsymbol{u}}) \\ \boldsymbol{G}'_u(\widehat{\boldsymbol{u}}_N) \\ \boldsymbol{G}'_e\left(\boldsymbol{x}^b(t_f)\right) \\ \boldsymbol{G}_e\left(\boldsymbol{x}(t_f)\right) \end{pmatrix} \tag{12j}$$

$$\boldsymbol{G}'_u(\widehat{\boldsymbol{u}}_N) = \begin{pmatrix} \boldsymbol{G}_u(\boldsymbol{u}_1^N) \\ \vdots \\ \boldsymbol{G}_u(\boldsymbol{u}_N^N) \end{pmatrix} \tag{12k}$$

$$\boldsymbol{G}'_e\left(\boldsymbol{x}^b(t_f)\right) = \boldsymbol{x}^b(t_f) \tag{12l}$$

3.2. Nonconvex sensitivity-based generalized Benders decomposition

Since the hybrid parametric dynamic optimization has clearly decomposable structure, GBD-based approach could be applied. Recently a novel decomposition algorithm that combines NSGBD with CVP has been developed to solve hybrid parametric dynamic optimization [33]. It discretizes continuous control variables, designates batch control variables as complicating variables and applies three techniques: (1) an extra variable and equality constraint is introduced to generate consistent linear Benders cuts, which provides linear programming (LP) master problems and sensitivity information of batch operations, no matter how the complicating variables appear in the model; (2) for infeasible points infeasible minimum problems are constructed to generate supporting hyperplanes to cut off the infeasible region of complicating variables and new feasible points; (3) with the check of optimality conditions linear Benders cuts are directly manipulated to tackle nonconvexity.

By designating the batch operations $\overline{\boldsymbol{u}}$ as complicating variables and introducing a new batch control variable $\underline{\boldsymbol{u}}$ and an extra equality constraint

$$\boldsymbol{h}(\overline{\boldsymbol{u}}, \underline{\boldsymbol{u}}) = \underline{\boldsymbol{u}} - \overline{\boldsymbol{u}} = \boldsymbol{0} \tag{13}$$

the NSGBD algorithm for Problem (P2) can be constructed with the corresponding problems listed below:

Problem (P3):

$$(\widehat{\boldsymbol{u}}_{N,a}^j | J_a^j) = \arg\min_{\widehat{\boldsymbol{u}}_N} J\left(\boldsymbol{x}(t_f), \overline{\boldsymbol{u}}_a^j\right) \tag{14a}$$

$$\text{s.t. } M\boldsymbol{X}(t_0) = \boldsymbol{X}_0 \tag{14b}$$

$$M\dot{\boldsymbol{X}}(t) = \boldsymbol{F}\left(t, \boldsymbol{X}(t), \widehat{\boldsymbol{u}}_N, \overline{\boldsymbol{u}}^j\right) \tag{14c}$$

$$\boldsymbol{G}\left(\overline{\boldsymbol{u}}_a^j, \widehat{\boldsymbol{u}}_N, \boldsymbol{X}(t_f)\right) \leq \boldsymbol{0} \tag{14d}$$

where J_a^j is the optimal value of objective function.

Problem (P4):

$$(\widehat{\boldsymbol{u}}_{N,b}^j, \overline{\boldsymbol{u}}_b^j | \boldsymbol{\mu}_a^j) = \arg\min_{\widehat{\boldsymbol{u}}_N, \overline{\boldsymbol{u}}} J\left(\boldsymbol{x}(t_f), \overline{\boldsymbol{u}}\right) \tag{15a}$$

$$\text{s.t. } M\boldsymbol{X}(t_0) = \boldsymbol{X}_0 \tag{15b}$$

$$M\dot{\boldsymbol{X}}(t) = \boldsymbol{F}\left(t, \boldsymbol{X}(t), \widehat{\boldsymbol{u}}_N, \overline{\boldsymbol{u}}\right) \tag{15c}$$

$$\boldsymbol{G}\left(\overline{\boldsymbol{u}}, \widehat{\boldsymbol{u}}_N, \boldsymbol{X}(t_f)\right) \leq \boldsymbol{0} \tag{15d}$$

$$\boldsymbol{h}\left(\overline{\boldsymbol{u}}, \underline{\boldsymbol{u}}_a^j\right) = \boldsymbol{0} \tag{15e}$$

where $\boldsymbol{\mu}_a^j$ is the multiplier of the equality constraint (15e).

Problem (P5):

$$(\widehat{\boldsymbol{u}}^{j'}_{N,c}, \overline{\boldsymbol{u}}^{j'}_{c}) = \arg\min_{\widehat{\boldsymbol{u}}_N, \overline{\boldsymbol{u}}} \| \overline{\boldsymbol{u}} - \overline{\boldsymbol{u}}^{j}_{t} \|^2_A \tag{16a}$$

$$\text{s.t. } M\boldsymbol{X}(t_0) = \boldsymbol{X}_0 \tag{16b}$$

$$M\dot{\boldsymbol{X}}(t) = \boldsymbol{F}\left(t, \boldsymbol{X}(t), \widehat{\boldsymbol{u}}_N, \overline{\boldsymbol{u}}\right) \tag{16c}$$

$$\boldsymbol{G}(\overline{\boldsymbol{u}}, \widehat{\boldsymbol{u}}_N, \boldsymbol{X}(t_f))\leq \boldsymbol{0} \tag{16d}$$

where

$$\| \overline{\boldsymbol{u}} - \overline{\boldsymbol{u}}^{j} \|^2_A = (\overline{\boldsymbol{u}} - \overline{\boldsymbol{u}}^{j})^T A (\overline{\boldsymbol{u}} - \overline{\boldsymbol{u}}^{j}) \tag{16e}$$

A is a symmetric positive definite matrix.

Problem (P6):

$$(\alpha^{j'}_{d}, \widehat{\boldsymbol{u}}^{j'}_{N,d}, \overline{\boldsymbol{u}}^{j'}_{d} | \boldsymbol{\mu}^{j'}_{b}) = \arg\min_{\alpha, \widehat{\boldsymbol{u}}_N, \overline{\boldsymbol{u}}} \alpha \tag{17a}$$

$$\text{s.t. } M\boldsymbol{X}(t_0) = \boldsymbol{X}_0 \tag{17b}$$

$$M\dot{\boldsymbol{X}}(t) = \boldsymbol{F}\left(t, \boldsymbol{X}(t), \widehat{\boldsymbol{u}}_N, \overline{\boldsymbol{u}}\right) \tag{17c}$$

$$\boldsymbol{G}(\overline{\boldsymbol{u}}, \widehat{\boldsymbol{u}}_N, \boldsymbol{X}(t_f))-\alpha\boldsymbol{e}\leq \boldsymbol{0} \tag{17d}$$

$$\boldsymbol{h}(\overline{\boldsymbol{u}}, \overline{\boldsymbol{u}}^{j'}_{c}) = \boldsymbol{0} \tag{17e}$$

where $\boldsymbol{\mu}^{j'}_{b}$ is the multiplier of equality constraint (17e) and $\boldsymbol{e}$ is vector whose elements are all equal to one with proper dimension.

The constraints on $\overline{\boldsymbol{u}}$ (Eq. (7)) can be represented as:

$$\boldsymbol{\mu}^{i}_{c} \cdot \left(\underline{\boldsymbol{u}} - \overline{\boldsymbol{u}}^{ub}\right)\leq 0, i \in K_u \tag{18a}$$

$$\boldsymbol{\mu}^{i'}_{d} \cdot \left(\underline{\boldsymbol{u}} - \overline{\boldsymbol{u}}^{lb}\right)\leq 0, i' \in K_l \tag{18b}$$

where $\boldsymbol{\mu}^{i}_{c} = (\underbrace{0, \ldots, 0}_{i-1}, 1, \underbrace{0 \ldots, 0}_{l-i})$ and $\boldsymbol{\mu}^{i'}_{d} = (\underbrace{0, \ldots, 0}_{i'-1}, -1, \underbrace{0 \ldots, 0}_{l-i'})$.

Problem (P7):

$$\left(\eta^{k}_{b}, \underline{\boldsymbol{u}}^{k}_{b} | \boldsymbol{v}^{k}_{a}, \boldsymbol{v}^{k}_{b}, \boldsymbol{v}^{k}_{c}, \boldsymbol{v}^{k}_{d}\right) = \arg\min_{\eta, \underline{\boldsymbol{u}}} \eta \tag{19a}$$

$$\eta \geq J^{j}_{a} + \boldsymbol{\mu}^{j}_{a} \cdot \left(\underline{\boldsymbol{u}} - \overline{\boldsymbol{u}}^{j}_{a}\right), j \in K_{feas} \tag{19b}$$

$$\boldsymbol{\mu}^{j'}_{b} \cdot \left(\underline{\boldsymbol{u}} - \overline{\boldsymbol{u}}^{j'}_{c}\right)\leq 0, j' \in K_{infeas} \tag{19c}$$

$$\boldsymbol{\mu}^{i}_{c} \cdot \left(\underline{\boldsymbol{u}} - \overline{\boldsymbol{u}}^{ub}\right)\leq 0, i \in K_u \tag{19d}$$

$$\boldsymbol{\mu}^{i'}_{d} \cdot \left(\underline{\boldsymbol{u}} - \overline{\boldsymbol{u}}^{lb}\right)\leq 0, i' \in K_l \tag{19e}$$

where $\boldsymbol{v}^{k}_{a}$, $\boldsymbol{v}^{k}_{b}$, $\boldsymbol{v}^{k}_{c}$ and $\boldsymbol{v}^{k}_{d}$ are the Lagrangian multipliers for K_{feas}, K_{infeas}, K_u and K_l,

Define

$$\Xi_a = \{j \in Z : (\boldsymbol{v}^{k}_{a})_j \neq 0\} \tag{20a}$$

$$\Xi_b = \{j' \in Z : (\boldsymbol{v}^{k}_{b})_{j'} \neq 0\} \tag{20b}$$

$$\Xi_c = \{i \in Z : (\boldsymbol{v}^{k}_{c})_i \neq 0\} \tag{20c}$$

$$\Xi_d = \{i' \in Z : (\boldsymbol{v}^{k}_{d})_{i'} \neq 0\} \tag{20d}$$

Problem (P8):

$$(\underline{\boldsymbol{u}}^{k}_{c}) = \arg\min_{\underline{\boldsymbol{u}}} \boldsymbol{\mu}^{k}_{a} \cdot \underline{\boldsymbol{u}} \tag{21a}$$

$$\boldsymbol{\mu}^{j'}_{b} \cdot (\underline{\boldsymbol{u}} - \overline{\boldsymbol{u}}^{j'}_{c})\leq 0, j' \in K_{infeas} \tag{21b}$$

$$\boldsymbol{\mu}^{i}_{c} \cdot (\underline{\boldsymbol{u}} - \overline{\boldsymbol{u}}_{u})\leq 0, i \in K_u \tag{21c}$$

$$\boldsymbol{\mu}^{i'}_{d} \cdot (\underline{\boldsymbol{u}} - \overline{\boldsymbol{u}}_{l})\leq 0, i' \in K_l \tag{21d}$$

The basic equations about convergence:

$$\text{LBD} > \text{UBD} \tag{22a}$$

$$\text{UBD} - \text{LBD} < \varepsilon_1 \tag{22b}$$

The equations for the checking of optimality conditions:

$$\| \boldsymbol{\mu}^{k}_{a}/J^{k}_{a} \|_B \leq \varepsilon_2 \tag{23a}$$

$$\| \underline{\boldsymbol{u}}^{k}_{b} - \underline{\boldsymbol{u}}^{k}_{c} \|_B \leq \varepsilon_3 \tag{23b}$$

Algorithm 1: **NSGBD with CVP for Problem (P1)**

Step 1. Determine a set of proper inner points in optimization horizon $[t_0, t_f]$, namely $t_0 < t_1 < t_2 < \ldots < t_N = t_f$, and parameterize the continuous control variable $\boldsymbol{u}(t)$ as $\widehat{\boldsymbol{u}}_N$. Find a point $\overline{\boldsymbol{u}}^{1}_{t}$ according to Eq. (7); let $j = 1$, $j' = 1$, $K_{feas} = \emptyset$, $K_{infeas} = \emptyset$, $LBD = -\infty$; set the hyper parameters γ, ε_1, ε_2 and ε_3.

Step 2. Solve the primal Problem (P3) for $\overline{\boldsymbol{u}}^{j}_{a} = \overline{\boldsymbol{u}}^{j}_{t}$. One of the following cases must occur:

(1). Primal problem (P3) is feasible with $\widehat{\boldsymbol{u}}^{j}_{N,a}$ and J^{j}_{a}; solve the primal Problem (P4) for $\underline{\boldsymbol{u}}^{j}_{a} = \overline{\boldsymbol{u}}^{j}_{a}$ at $\overline{\boldsymbol{u}}^{j}_{a}$ and $\widehat{\boldsymbol{u}}^{j}_{N,a}$ with $\boldsymbol{\mu}^{j}_{a}$. UBD $= J^{j}_{a}$; One of the following cases must occur:

①. LBD > UBD. $\boldsymbol{\mu}^{l}_{a} = \gamma\boldsymbol{\mu}^{l}_{a}, l \in \Xi_a$.

②. UBD − LBD $< \varepsilon_1$. One of the following cases must occur:

(a). $|\Xi_b| + |\Xi_c| + |\Xi_d| = 0$ ($\overline{\boldsymbol{u}}^{j}_{a}$ is an inner point). If Eq. (23a) is satisfied, then the algorithm terminated with optimal solution $(\widehat{\boldsymbol{u}}^{j}_{N,a}, \overline{\boldsymbol{u}}^{j}_{a})$, else $\boldsymbol{\mu}^{l}_{a} = \gamma\boldsymbol{\mu}^{l}_{a}, l \in \Xi_a$.

(b). $|\Xi_b| + |\Xi_c| + |\Xi_d| \neq 0$ ($\overline{\boldsymbol{u}}^{j}_{a}$ is a boundary point). Solve Problem (P8) at $\overline{\boldsymbol{u}}^{j}_{a}$ with $\underline{\boldsymbol{u}}^{j}_{c}$. If Eq. (23b) is satisfied ($\underline{\boldsymbol{u}}^{j}_{b} = \overline{\boldsymbol{u}}^{j}_{a}$), then the algorithm terminated with optimal solution $(\widehat{\boldsymbol{u}}^{j}_{N,a}, \overline{\boldsymbol{u}}^{j}_{a})$.

③. Otherwise.

$$K_{feas} = \{K_{feas}, j\};\ j = j + 1.$$

(2). Primal Problem (P3) is infeasible. Solve the Problem (P5) for $\overline{\boldsymbol{u}}^{j}_{t}$ with $(\widehat{\boldsymbol{u}}^{j'}_{N,c}, \overline{\boldsymbol{u}}^{j'}_{c})$. Solve the Problem (P6) for $\overline{\boldsymbol{u}}^{j'}_{c}$ at $(\widehat{\boldsymbol{u}}^{j'}_{N,c}, \overline{\boldsymbol{u}}^{j'}_{c})$ and $\alpha = 0$ with $\boldsymbol{\mu}^{j'}_{b}$. $K_{infeas} = \{K_{infeas}, j'\}$. $\overline{\boldsymbol{u}}^{j}_{t} = \overline{\boldsymbol{u}}^{j'}_{c}$, $j' = j' + 1$. Return to Step 2.

Step 3. Solve master Problem (P7) with η^{j}_{b}, $\underline{\boldsymbol{u}}^{j}_{b}$, $\boldsymbol{v}^{j}_{a}$, $\boldsymbol{v}^{j}_{b}$, $\boldsymbol{v}^{j}_{c}$ and $\boldsymbol{v}^{j}_{d}$; LBD $= \eta^{j}_{b}$, $\overline{\boldsymbol{u}}^{j}_{t} = \underline{\boldsymbol{u}}^{j}_{b}$. Return to Step 2.

Remark 1. The algorithm flowchart of NSGBD [13] is given at Fig. 2. Problems (P3) and (P4) are the primal problems for fixed $\overline{\boldsymbol{u}}^{j}_{a}$ to generate a Benders cut. The primal Problem (P3) is solved with fixed complicating variables and the primal Problem (P4) is solved at the solution found in primal Problem (P3). Problems (P5) and (P6) are the infeasible minimum problems for infeasible $\overline{\boldsymbol{u}}^{j}_{t}$ to generate a feasible point $\overline{\boldsymbol{u}}^{j'}_{c}$ and a supporting hyperplane of feasible region of $\overline{\boldsymbol{u}}$ described by $\boldsymbol{\mu}^{j'}_{b}$, which is used to approximate the feasible region of $\overline{\boldsymbol{u}}$. Problem (P5) finds a feasible point that is closest to the infeasible point. Problem (P6) is solved at the solution found in primal Problem (P5) and $\alpha = 0$. Problem (P7) is the LP master problem for the generation of new points. Ξ_a is the index of active Benders cuts. At nonconvex points, the multipliers of active Benders cuts indexed by Ξ_a will be multiplied by γ to provide valid LBDs. Ξ_b, Ξ_c and Ξ_d are the index of

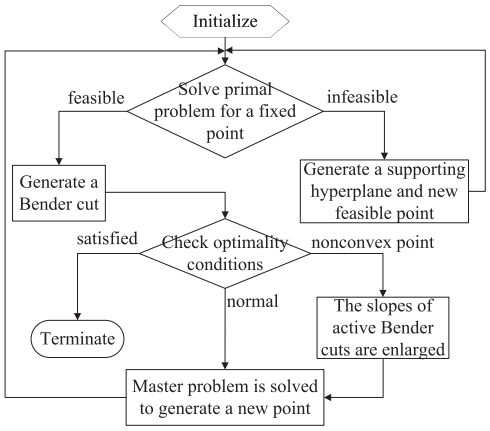

Fig. 2. The algorithm flowchart of NSGBD.

active constraints for K_{infeas}, K_u and K_l, which are used to distinguish between boundary and inner points. Eq. (22a) indicates that the master problem gives an invalid LBD, which indicates the nonconvexity of original problem. Eq. (23a) and (23b) are the local conditions for optimality at inner and boundary points. Problem (P8) is the corresponding problem for Eq. (23b), which aims to check the optimality conditions numerically. The key technique of NSGBD is the detection of nonconvex points and the operation of corresponding active Benders cuts, which is described by Step 2 of Algorithm 1.

Remark 2. There are two properties of NSGBD that will be exploited in the follows: 1) $\boldsymbol{\mu}_a^j$ is the gradient (sensitivity) of projected cost function at differentiable points [13] with respect to $\bar{\boldsymbol{u}}$; 2) parallel scheme [38] can be easily embedded into NSGBD, i.e. multiple primal problems can be solved independently and simultaneously.

3.3. Novel implementation framework of optimal solution

As primal problems are solved by a direct method, namely CVP, the solution of continuous operations still share the common limitations of direct methods, i.e. the solution of continuous operations is usually suboptimal, which renders the solution of batch operations suboptimal. Moreover, the solution of continuous operations is of the open-loop form. To overcome these limitations, a novel implementation framework is proposed as follows.

The open-loop optimal continuous operations could tell the composition of arcs of the analytical solution, which includes constraint-seeking and sensitivity-seeking arcs in the sense of necessary conditions of optimality. In detail, the constraint-seeking arcs are determined by the active path constraints, while sensitivity-seeking arcs are determined by the dynamic model and cost function, and regulating these conditions around zero leads to optimality. Hence, the NCO-tracking scheme can be used to implement the optimal continuous operations of dynamic optimization as extra closed-loop controllers under the assumption that the set of active constraints calculated by former dynamic optimization does not change with uncertainty, i.e. the structure of the optimal solution (the types and sequence of arcs) of the true system is known a priori.

Due to the discretization of continuous variables, the feasible region of continuous operations for reformulated NLP shrinks (a subspace of original feasible region.). Working through the dynamic system, it also shrinks the feasible region of batch operations, which makes the optimal batch operations obtained by NSGBD suboptimal. However, this solution includes extra sensitivity information, which makes it possible to conduct an extra line search (sensitivity analysis) to improve its quality. In detail, with the optimal continuous operations implemented by NCO-tracking scheme, a line search with respect to scalar κ is given as follows:

$$\bar{\boldsymbol{u}} = \bar{\boldsymbol{u}}_a^{j*} + \kappa \boldsymbol{\mu}_a^{j*} \tag{24}$$

where $\bar{\boldsymbol{u}}_a^{j*}$ is the optimal batch operations obtained by NSGBD algorithm, and $\boldsymbol{\mu}_a^{j*}$ is the corresponding multiplier of Problem (P4). Then several simulations about different κ are conducted to find improved optimal batch operations that maximizes the economic performance without violating the constraints under satisfactory precision. Note that if $\bar{\boldsymbol{u}}_a^{j*}$ is an inner point, where $\| \boldsymbol{\mu}_a^{j*} \|$ is small, then the sign of κ is uncertain; if $\bar{\boldsymbol{u}}_a^{j*}$ is a boundary point, where $\| \boldsymbol{\mu}_a^{j*} \|$ is large, then $\kappa < 0$. The magnitude of κ can be estimated by the neighbor points of $\bar{\boldsymbol{u}}_a^{j*}$ and the precision required in practice.

Combining the NSGBD algorithm with NCO-tracking scheme, this implementation framework, which is illustrated by Fig. 3, can also obtain near-optimal performance under uncertainty. Comparing to the implementation framework that integrates the adaptive CVP with NCO-tracking scheme [7], this novel scheme shares and exploits the advantages of NSGBD algorithm in two aspects: 1) when the dimension of batch operations is relatively large, the solving of primal problems of NSGBD will be much easier than the solving of NLP that parameterizes the continuous and batch operations simultaneously, while LP master problems can be easily solved. In extreme cases, large amount of batch operations and discretized continuous operations will give too many decision variables, which makes simultaneous optimization of continuous and batch operations intractable and renders the NSGBD become the only alternative; 2) the sensitivity information about batch operations is fully exploited by conducting a line search, which obtains a high-quality solution without using adaptive methods or a very fine discretization mesh.

Note that the scheme proposed at Fig. 3 is not an extension of the general methodology of NCO tracking. The NCO-tracking is used in Fig. 3 to improve the solution quality of continuous operations while the solution quality of batch operations is improved with the help of sensitivity information of NSGBD. In every simulation of Fig. 3, no uncertainty or disturbance is introduced. The four NCO parts include both the profile and pointwise objectives that must be met and each objective involves the enforcement of active constraints and the zeroing of reduced gradients [39], where the active constraints can be controlled at the active boundaries using appropriate operational degrees of freedom. In practice, the gradients can be calculated by the neighboring-extremal (NE) approach [40] (the gradients are computed using the difference between nominal (corresponding to $\bar{\boldsymbol{u}}_a^{j*}$) and measured outputs), finite-difference (perturbation) approach or regression approach [41]. Also, note that it is not always necessary to evaluate the sensitivities directly, because quantities that go to zero with them suffice for implementation. In fact, the batch operations can also be treated as pointwise decision variables in NCO-tracking framework, which can be adapted on a run-to-run basis under the real uncertain processes [42–44]. Then the scheme proposed at Fig. 3 is nothing but providing a good initial guess for batch operations which speeds up the convergence. In practice, the algorithm can be executed in each control interval to provide a new solution of batch operations and a new NCO structure of continuous operations for

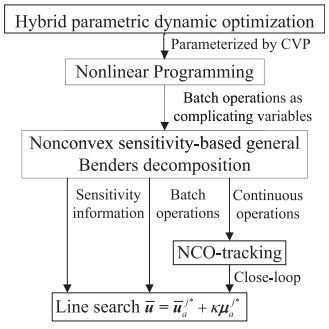

Fig. 3. A novel implementation scheme combining NSGBD with NCO-tracking.

next cycle, or the batch operations can be taken into a part of NCO enforcement and adapt with continuous operations batch-to-batch, while the algorithm is executed when large-scale change of working conditions is expected.

4. Hybrid parametric dynamic optimization of FCCU

4.1. Mathematical formulation of FCCU

As discussed in Lin et al. [7], riser temperature is directly related to the productivity of the valuable product, combustion air flow rate is a mainly measure of adjustable utility cost and the addition of CO promoter consumes extra resources, then continuous control variable $u(t)$ is taken as the set-points of riser temperature $T_{ra\_sp}(t)$, while batch control variables $\bar{\boldsymbol{u}}$ are the adjustment of combustion air flow rate V and the amount of added CO promoter M_{pro}. The batch operations are operated in a multirate mode, i.e. the CO promoter is added every $8\,h = 480$ min and the air blower is adjusted every $2\,h = 120$ min. For simplicity, the value of economic objective function is calculated by the difference between the current operation and the nominal operation, which is:

$$\min J\left(T_{ra\_sp}(t),\{V_i\}_{i=1}^{4},M_{pro}\right)=\int_0^{480}\left(-\omega_{1d}F_d\left(t,T_{ra\_sp}(t),\{V_i\}_{i=1}^{4},M_{pro}\right)-\omega_{1n}F_n\left(t,T_{ra\_sp}(t),\{V_i\}_{i=1}^{4},M_{pro}\right)\right)dt+\sum_{i=1}^{4}\int_{120(i-1)}^{120i}f(V_i)dt+\omega_3 M_{pro} \tag{25}$$

where ω_{1d}, ω_{1n} and ω_3 denote the price of diesel, naphtha and CO promoter; F_d and F_n are the yield of diesel and naphtha; f is the energy consumption of air blower with respect to $\{V_i\}_{i=1}^{4}$. The constraints imposed on the decision variables and on the other state variables for FCCU can be referred in Lin et al. [7]. Hence, the hybrid parametric dynamic optimization is to search an optimal trajectory of $T_{ra\_sp}(t)$ and optimal values of M_{pro} and $\{V_i\}_{i=1}^{4}$ to minimize objective function Eq. (25), while satisfies the system model and constraints. Note that the economic objective function is multiplied by a minus sign for the reformulation of a minimization form. To coincide with the conventional custom, the minus sign is discarded in the narration.

To compare the effect of optimizing V as a batch operation and provide an upper bound for above cases, a case considering V as a continuous operation is carried out, whose cost function is

$$\min J\left(T_{ra\_sp}(t),V(t),M_{pro}\right)=\int_0^{480}\left(-\omega_{1d}F_d\left(t,T_{ra\_sp}(t),V(t),M_{pro}\right)-\omega_{1n}F_n\left(t,T_{ra\_sp}(t),V(t),M_{pro}\right)+f\left(V(t)\right)\right)dt+\omega_3 M_{pro} \tag{26}$$

Considering the CO promoter may work on a preset way ($M_{pro} = 4$kg), two extra cases are carried out to compare these effects. All cases considered are given by Table 1. The detail about the well-posed of this hybrid parametric dynamic optimization can be referred in Lin et al. [7].

The NSGBD algorithm was implemented using gPROMS/gOPT to solve the dynamic optimization primal problems, and MATLAB for the LP master problems. Moreover, the piecewise constant controls are used for the approximation of continuous controls with 16 equidistant meshes. Note that there aren't batch operations in Case 3, which makes the NSGBD unnecessary and is solved by normal CVP.

Table 1
All cases considered.

Case	1	2	3	4
CO promoter	Batch	Batch	Preset	Preset
Combustion air	Continuous	Batch	Continuous	Batch

4.2. Case 1: Combustion air as a continuous operation whereas CO promoter as a batch operation

In this case, the batch operation is M_{pro} and the continuous operations are T_{ra_sp} and V, which is the case considered in Lin et al. [7], where the adaptive CVP is used and the result will be compared latter. Here, there are 32 decision variables in the reformulated NLPs of primal problems. Using the NSGBD algorithm, all the feasible primal solutions, Lagrange multipliers and master solutions are summarized in Table 2. Using a termination tolerance of ε_1=1, ε_2= 0.1 and ε_2= 0.1, NSGBD algorithm converged after 5 iterations.

The parallel scheme is used in the first iteration, where primal problems for $M_{pro}^{1.1}=2$ and $M_{pro}^{1.2}=4$ are solved simultaneously. The algorithm converged in 5 iterations when $J^5 - LBD^5 \le 1$ and $\|\mu^5/J^5\| \le 5\times10^{-3}$ (inner point), and all the M_{pro}^{j} are feasible points. Current optimal solution is $M_{pro}^{5}=3.1864$kg, $J^5 = 1655.2$¥. The optimal solutions of other continuous variables at iteration 5 are given in the solid lines of Fig. 4.

As shown in the solid lines of Fig. 4a and 4b, the optimal solution requires the higher riser temperature and combustion air flow rate with the higher activity of CO promoter, which is the same as Lin et al. [7]. As shown in Fig. 4c and 4d, the upper-bound constraints of O_2 molar fraction in flue gas and temperature rise in the freeboard are active throughout the entire time, which means the optimal solution is only composed by constraint-seeking arcs. Hence, according to NCO-tracking scheme, the optimal continuous control variables, namely riser temperature set-points and combustion air flow rate, can be determined by these two constraints. Hence, two extra regulatory controllers are needed to maintain two upper bounds constraints active, and it leads to an input-output pairings problem. As discussed in Lin et al. [7], riser temperature set-points should pair with temperature rise in the freeboard and combustion air flow rate pairs with O_2 molar fraction in flue gas.

Define $L = \{k \in \mathrm{N}: \mu^k < 0\}$, $R = \{k \in \mathrm{N}: \mu^k > 0\}$, then it should have $M_{pro}^k < M_{pro}^*$ for $k \in L$ and $M_{pro}^k > M_{pro}^*$ for $k \in R$. Hence, the rough estimation of optimal batch operations for the original problem is

$$M_{pro}^* \in (\max_{k\in L} M_{pro}^k, \min_{k\in R} M_{pro}^k) = (3.1864, 3.3440)\text{kg} \tag{27}$$

By the novel framework proposed in Section 3.3, a line search with the implementation of NCO-tracking can be conducted, which is shown in Table 3.

As shown at the first and third rows of Table 3, i.e. $M_{pro} = 3.1864$kg and $M_{pro} = 3.3440$kg, better performances than the solution of primal problems are obtained with the help of NCO-tracking scheme, which is caused by the safety margin left at every constant control section caused by discretization. By the last two rows of Table 3, optimal solution of batch operation M_{pro} should reside in the interval [3.344, 3.45]kg. Then optimal implementation of batch operation can be $M_{pro} = 3.4kg$ with $J = 2732.3$¥, and the corresponding simulation with NCO-tracking scheme is given by the dash lines of Fig. 4. As can be seen at the dash lines of Fig. 4c and 4d, the safety margin left at every constant control section has been largely eliminated by NCO-tracking. Comparing with the result obtained in Lin et al. [7] ($M_{pro} = 3.35$kg, $J = 2697.05$¥), which is solved by simultaneous adaptive CVP with the

Table 2
Progress of iterations of NSGBD algorithm.

Iteration	M_{pro}(kg)	μ	J(¥)	LBD(¥)	J-LBD(¥)
1.1	2.0000	−134.12	−1582.8	–	–
1.2	4.0000	45.905	−1636.8	–	–
2	2.8097	−35.649	−1647.9	−1691.4	43.570
3	3.3440	7.2533	−1655.0	−1666.9	11.945
4	3.0655	−14.828	−1654.3	−1657.0	2.6695
5	3.1864	−4.6607	−1655.2	−1656.1	0.87264

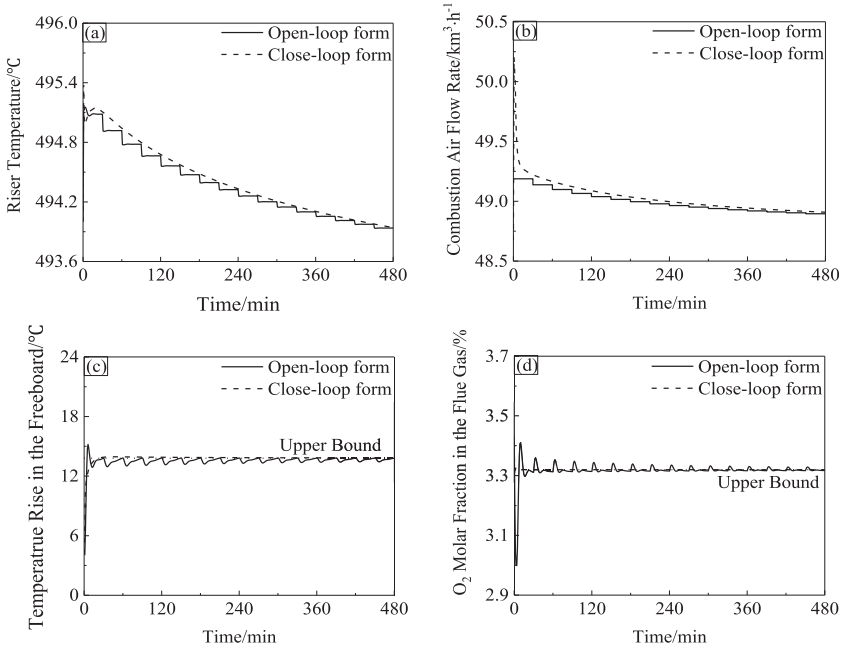

Fig. 4. optimal results in open-loop (M_{pro}=3.1864 kg) and close-loop (M_{pro}=3.4 kg) forms: (a) riser temperature, (b) combustion air flow rate, (c) temperature rise in the freeboard, (d) O_2 molar fraction in flue gas.

Table 3
Line search with respect to batch operations.

M_{pro}(kg)	J(¥)
3.1864	2728.7
3.2652	2730.4
3.3440	2731.7
3.4500	2731.5

piecewise constant control approximation for continuous operations, a high-quality solution with better precision is obtained with relatively coarse discretization. In detail, simultaneous adaptive CVP converged in 14 iterations with 43 decision variables (21 meshes), while the NSGBD converged in 5 iterations with 32 decision variables (16 meshes). Note that the scale of NLPs is different for adaptive CVP and NSGBD. Adaptive CVP needs to repetitively solve increasingly refined finite-dimensional optimization problems, while the scale of primal problems is identical for all the iterations in NSGBD.

4.3. Case 2: Combustion air and CO promoter as batch operations

In this case, the reformulated NLPs of primal problems only contain 16 decision variables, as there is only one continuous operation. Considering the length of this paper, only partial feasible primal solutions, Lagrange multipliers and master solutions are summarized in Table 4, partial solutions of infeasible minimum problems and Lagrange multipliers for supporting hyperplanes are summarized in Table 5, and partial active constraints of master problems are summarized in Table 6. Using a termination tolerance of ε_1= 1, ε_2= 0.1 and ε_3= 0.1, NSGBD algorithm converged after 10 iterations.

The parallel scheme is used in iterations 1, 2, 3 and 4. At iteration 1, 4 out of 2^5=32 corner points are calculated simultaneously. The corner points (2, 48, 48, 48, 48) and (4, 48, 48, 48, 48) in iteration 1.1 and 1.2 are feasible points and obtain two Benders cuts for master problems. The corner points (2, 50, 50, 50, 50) and (4, 50, 50, 50, 50) in iteration 1.3 and 1.4 are infeasible points, which invoke two infeasible problems simultaneously and obtain two new feasible points, Benders cuts for master problems and supporting hyperplanes of feasible region. At iteration 2.1, new batch operations obtained from the master problem are also infeasible, which invokes one infeasible problem and obtains one new feasible point, Benders cut for master problems and supporting hyperplane of feasible region. Simultaneously, 2 corner points (4, 50, 48, 48, 48) and (4, 48, 50, 48, 48) are calculated in iteration 2.2 and 2.3. They are also infeasible points, which invoke two infeasible problems simultaneously and obtain two new feasible points, Benders cuts for master problems and supporting hyperplanes of feasible region.

At iterations 3.1 and 4.1, new batch operations are obtained from master problems, and parallel scheme is used to generate 3 extra neighbour points, i.e. iteration 3.2, 3.3 and 3.4 and iteration 4.2, 4.3, and 4.4, which are all infeasible points. Both iterations (3 and 4) invoke four infeasible problems. From iteration 5, new batch operations obtained from master problems are feasible. However, at iterations 5 and 8, master problems give false estimates of lower bounds, i.e. LBD > UBD, which indicates the nonconvexity of the FCCU problem. For iteration 5, the active Benders cuts are the ones generated by iterations 1.2 and 4.2 (shown at Table 6), whose Lagrange multipliers are multiplied by γ = 1.16. The same operation is conducted at iteration 8, and the algorithm converges at iteration 10. Since there are active border constraints at iteration 10 (shown at Table 6), (M_{pro}^{10}, $V_1^{10}, V_2^{10}, V_3^{10}, V_4^{10}$) is a boundary point. Then Problem (P8) is solved and convergence criterion Eq. (24b) holds. Hence, optimal solution of NSGBD is: $M_{pro}^{10} = 3.3223$kg, $V_1^{10} = 49.068$km^3/h, $V_2^{10} = 48.981$km^3/h, $V_3^{10} = 48.931$km^3/h, $V_4^{10} = 48.896$km^3/h, J^{10} = 1279.8¥, and the solution of some other variables at iteration 10 are given in the solid lines of Fig. 5.

As shown in Fig. 5c, the upper bound constraint of temperature rise in the freeboard is active throughout the entire time, which means the optimal solution is only composed by constraint-seeking arcs. Hence, according to NCO-tracking scheme, the optimal continuous control variable, namely riser temperature set-points, can be determined by this constraint. Hence, one extra regulatory controller is needed to maintain this upper bound constraint active by adjusting the set-points of riser temperature. Moreover, define $P = \{k \in \mathrm{N} : \mu_M^k > 0\}$, $\{N_i = \{k \in \mathrm{N} : \mu_{Vi}^k < 0\}\}_{i=1}^4$. As μ_M and $\{\mu_{Vi}\}_{i=1}^4$ are the gradients of projection function, the rough estimations of

Table 4
Partial feasible primal solutions.

Iteration	4.4	5	6	7	8	9	10
M_{pro}(kg)	2.5865	2.4590	2.7855	2.8738	2.8890	3.2639	3.3223
μ_M	8.9744	−21.541	27.047	25.754	20.927	53.115	58.887
V_1(km^3/h)	49.034	49.061	49.064	49.064	49.065	49.068	49.068
μ_{V1}	−3480.3	−3515.4	−3471.9	−3479.8	−3490.8	−3480.0	−3468.3
V_2(km^3/h)	48.952	48.977	48.978	48.979	48.979	48.980	48.981
μ_{V2}	−3456.5	−3462.7	−3447.1	−3476.1	−3461.9	−3428.1	−3434.2
V_3(km^3/h)	48.928	48.927	48.929	48.929	48.929	48.930	48.931
μ_{V3}	−3402.3	−3458.2	−3441.6	−3452.5	−3465.0	−3479.2	−3441.8
V_4(km^3/h)	48.872	48.894	48.895	48.895	48.895	48.896	48.896
μ_{V4}	−3461.3	−3505.7	−3407.5	−3426.8	−3469.4	−3415.2	−3441.1
J(¥)	−989.35	−1258.4	−1271.8	−1274.4	−1275.3	−1279.8	−1279.8
LBD(¥)	–	−1258.3	−1278.4	−1275.5	−1275.3	−1290.9	−1280.5
J-LBD(¥)	–	−0.10351	6.5509	1.0377	−0.047900	11.125	0.66554

Table 5
Partial infeasible primal solutions.

Iteration	4.1	4.2	4.3	4.4
M_{pro}(kg)	2.4295	2.5865	2.5865	2.5865
M'_{pro} (kg)	2.4295	2.58652	2.5865	2.5865
μ'_M	−4.3090	-1.4923×10	−8.7590	−5.7774
V_1(km^3/h)	49.061	49.154	49.034	49.0342
V'_1 (km^3/h)	49.061	49.062	49.034	49.0342
μ'_{V1}	9.6649×10^{-5}	1.4561×10^{5}	-3.8917×10^{-6}	1.8691×10^{-3}
V_2(km^3/h)	48.977	48.952	49.079	48.952
V'_2 (km^3/h)	48.970	48.952	48.970	48.952
μ'_{V2}	-2.2725×10^{-5}	1.2098×10^{-2}	1.4636×10^{5}	-4.2456×10^{-3}
V_3(km^3/h)	48.927	48.904	48.904	49.034
V'_3 (km^3/h)	48.924	48.904	48.904	48.928
μ'_{V3}	-6.8183×10^{-6}	-1.7140×10^{-3}	-2.5470×10^{-4}	1.4713×10^{5}
V_4(km^3/h)	48.894	48.872	48.872	48.872
V'_4 (km^3/h)	48.894	48.872	48.872	48.872
μ'_{V4}	1.4714×10^{5}	-3.6467×10^{-3}	-1.3657×10^{-4}	-3.1264×10^{2}

Table 6
Partial active constraints of master problems.

Iteration	5	8	10
Active constraints	1.2, 4.2	2.2, 7	1.4, 3.4, 4.1, 4.2, 4.3

optimal batch operations can be given as follows:

$$M^*_{pro} \in (\min_{k\in p} M^k_{pro}, M^{10}_{pro}) = (2.5865, 3.3223)\text{kg} \tag{28a}$$

$$V^*_1 \in (V^{10}_1, \max_{k\in N_1} V^k_1) = (49.068, 49.072)\text{km}^3/\text{h} \tag{28b}$$

$$V^*_2 \in (V^{10}_2, \max_{k\in N_2} V^k_2) = (48.981, 48.983)\text{km}^3/\text{h} \tag{28b}$$

$$V^*_3 \in (V^{10}_3, \max_{k\in N_3} V^k_3) = (48.931, 48.932)\text{km}^3/\text{h} \tag{28b}$$

$$V^*_4 \in (V^{10}_4, \max_{k\in N_4} V^k_4) = (48.896, 48.897)\text{km}^3/\text{h} \tag{28b}$$

Then a line search with the implementation of NCO-tracking can be conducted as the simulations shown in Table 7.

As shown at the first row of Table 7, i.e. a simulation for $(M^{10}_{pro}, V^{10}_1, V^{10}_2, V^{10}_3, V^{10}_4)$, better performances than primal problems of NSGBD are also obtained with the help of NCO-tracking scheme. By the last two rows of Table 7, the optimal solution of batch operations should be taken as: $M_{pro} \in (3.3223, 3.3223)$kg, $V_1 \in (49.076, 49.080)$km^3/h, $V_2 \in (48.988, 48.992)$km^3/h, $V_3 \in (48.938, 48.942)$km^3/h, $V_4 \in (48.904, 48.908)$km^3/h. Then optimal implementation of batch operations can be $M_{pro} = 3.3223$kg, $V_1 = 49.078$km^3/h, $V_2 = 48.990$km^3/h, $V_3 = 48.940$km^3/h, $V_4 = 48.906$km^3/hwith $J = 1890.4$¥, and the corresponding simulation with NCO-tracking scheme is given by the dash lines of Fig. 5. As can be seen at the dash line of Fig. 5c, the safety margin left at every constant control section has been largely eliminated by NCO-tracking. The batch operation of combustion air flow rate introduces a cycle 2 h into the optimization horizon 8 h, while there is an artificial cycle 0.5 h in the solution of dynamic optimization (illustrated by the solid lines of Fig. 5). Moreover, a better solution of batch operation is obtained by a line search method.

For Cases 3 and 4, only the values of optimized cost functions under NCO-tracking are given here for brevity, which are provided in Table 8. Comparing Case 2 to 1 and 4 to 3, the batch operation of combustion air deteriorates the economic performance. The reason for this can be explained by the local enlarged

Table 7
Line search with respect to batch operations.

M_{pro}(kg)	V_1(km^3/h)	V_2(km^3/h)	V_3(km^3/h)	V_4(km^3/h)	Violation	J(¥)
3.3223	49.068	48.981	48.931	48.896	No	1761.1
3.3223	49.072	48.985	48.934	48.900	No	1812.8
3.3223	49.076	48.988	48.938	48.904	No	1864.3
3.3223	49.080	48.992	48.942	48.908	Yes	1915.6

Table 8
Comparison of different operation modes.

Case	M_{pro}(kg)	V_1(km^3/h)	V_2(km^3/h)	V_3(km^3/h)	V_4(km^3/h)	J(¥)
1	3.4000	–	–	–	–	2732.3
2	3.3223	49.078	48.990	48.940	48.906	1890.4
3	4.0000	–	–	–	–	2723.2
4	4.0000	49.081	48.993	48.941	48.906	1866.6

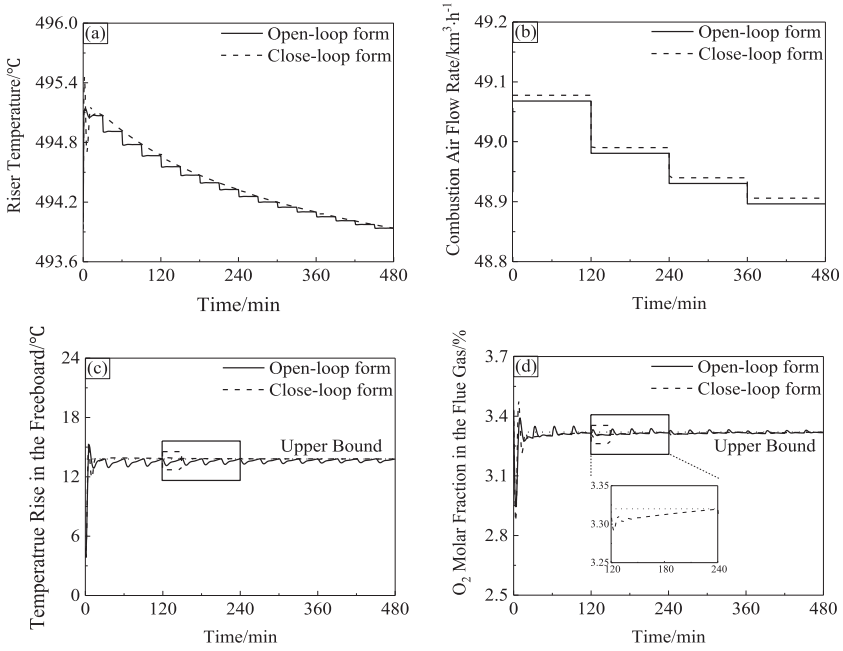

Fig. 5. optimal results in open-loop and close-loop forms (M_{pro}=3.3223 kg): (a) riser temperature and its set points, (b) combustion air flow rate, (c) temperature rise in the freeboard, (d) O_2 molar fraction in flue gas.

drawing in Fig. 5d. While the upper bound constraint for temperature rise in the freeboard is active most of the time by an extra PID controller, the upper bound constraint for O_2 molar fraction in flue gas is active only at the end-point of each cycle (2 h), which leaves much safety margin unexploited. Comparing Case 3 to 1 and Case 4 to 2, the prescribed operation of CO promoter mildly diminishes the economic performance, which suggests that the economic benefits obtained by integrated optimization mainly come from the continuous operations of riser temperature and combustion air.

5. Conclusions

This paper proposes a novel integrated optimization framework for the optimization of the continuous process with batch operations, which is illustrated by a commercial FCCU with CO promoter. This kind of integrated optimization must tackle an extremely difficult infinite dimensional optimization, namely dynamic optimization. It is still a challenge to obtain a high-quality solution for such problems efficiently, especially when the problem formulation contains large-scale models, which is normally solved by adaptive methods. However, this paper exploits the decomposable structure of hybrid parametric dynamic optimization, which is solved by NSGBD algorithm. In detail, by designating the batch operations as complicating variables, a sequence of primal and master problems is solved until convergence. Then the open-loop optimal continuous operations are implemented as extra close-loop controllers by NCO-tracking scheme to account for the uncertainty, while the optimal batch operations are improved by a line search method using the sensitivity information provided by NSGBD algorithm. For the case study of FCCU, the results show that a high-quality solution could be obtained by the proposed framework with relatively coarse discretization. Moreover, several cases have been covered to show that the batch operation of combustion air largely diminishes the economic performance, which is caused by the safety margin left at every constant control section, and the preset operation of CO promoter mildly diminishes the economic performance, which indicates that the economic benefits of integrated optimization mostly come from the continuous operations.

Declaration of Competing Interest

None.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (21676295).

Reference

[1] Wassick JM, Agarwal A, Akiya N, Ferrio J, Bury S, You F. Addressing the operational challenges in the development, manufacture, and supply of advanced materials and performance products. Comput Chem Eng 2012;47(20):157–69.
[2] Shi H, Chu Y, You F. Novel optimization model and efficient solution method for integrating dynamic optimization with process operations of continuous manufacturing processes. Ind Eng Chem Res 2015;54(7):2167–87.
[3] Nie Y, Biegler LT, Wassick JM. Integrated scheduling and dynamic optimization of batch processes using state equipment networks. AICHE J 2012;58(11):3416–32.
[4] Chu Y, You F. Integration of scheduling and dynamic optimization of batch processes under uncertainty: two-stage stochastic programming approach and enhanced generalized benders decomposition algorithm. Ind Eng Chem Res 2013;52(47):16851–69.
[5] Nie Y, Biegler LT, Villa CM, Wassick JM. Discrete time formulation for the integration of scheduling and dynamic optimization. Ind Eng Chem Res 2015;54(16):4303–15.
[6] Chu Y, You F. Integrated scheduling and dynamic optimization of complex batch processes with general network structure using a generalized benders decomposition approach. Ind Eng Chem Res 2013;52(23):7867–85.
[7] Lin J, Xu F, Luo X. Dynamic optimization of continuous-batch processes: a case study of an FCCU with CO promoter. Ind Eng Chem Res 2019;58(51):23187–200.
[8] Geoffrion AM. Generalized benders decomposition. J Optimiz Theory App 1972;10(4):237–60.
[9] Sahinidis NV, Grossmann IE. Convergence properties of generalized Benders decomposition. Comput Chem Eng 1991;15(7):481–91.
[10] Bansal V, Sakizlis V, Ross R, Perkins JD, Pistikopoulos EN. New algorithms for mixed-integer dynamic optimization. Comput Chem Eng 2003;27(5):647–68.
[11] Li X, Chen Y, Barton PI. Nonconvex generalized benders decomposition with piecewise convex relaxations for global optimization of integrated process design and operation problems. Ind Eng Chem Res 2012;51(21):7287–99.
[12] Li X, Tomasgard A, Barton PI. Nonconvex generalized benders decomposition for stochastic separable mixed-integer nonlinear programs. J Optimiz Theory App 2011;151(3):425–54.
[13] Lin J, Xu F, Luo X. Nonconvex sensitivity-based generalized Benders decomposition. J Glob Optim (Under Review)
[14] Secchi AR, Santos MG, Neumann GA, Trierweiler JO. A dynamic model for a FCC UOP stacked converter unit. Comput Chem Eng 2001;25(4–6):851–8.

[15] Wei F, Lin S, Yang G. Analysis and simulation of highefficiency regenerator of commercial FCCU I. Analysis of a high efficiency FCC regenerator. Acta Pet Sin (Pet Process Sect) 1994;10(2):21–8.
[16] Wei F, Lin S, Yang G. Analysis and simulation of highefficiency regenerator of commercial FCCU II. Model formulation and result of simulation. Analysis of a high efficiency FCC regenerator. Acta Pet Sin (Pet Process Sect) 1994;10(3):25–35.
[17] Luo X, Wei F, Hu C, Si Z, Huang J. A model for the FCC regenerator with fast fluidization and its application. Pet Process Petrochem 1993;24(9):1–8.
[18] Huang M, Zheng Y, Li S. Rto with modifier adaptation method applied to the fcc uint. In: Proceedings of the 2017 36th Chinese control conference (CCC). IEEE; 2017. p. 2990–5.
[19] Guan H, Ye L, Shen F, Song Z. Economic operation of a fluid catalytic cracking process using self-optimizing control and reconfiguration. J Taiwan Inst Chem Eng 2019;96:104–13.
[20] Aseev SM, Kryazhimskiy AV. The Pontryagin maximum principle and transversality conditions for a class of optimal control problems with infinite time horizons. SIAM J. Control Optim. 2004;43(3):1094–119.
[21] Howard RA. Dynamic programming. Manag Sci 1966;12(5):317–48.
[22] Lin J, Luo X. Hybrid parametric minimum principle. Nonlinear Anal Hybri 2020;37:100902. doi: 10.1016/j.nahs.2020.100902.
[23] Hartwich A, Marquardt W. Dynamic optimization of the load change of a large-scale chemical plant by adaptive single shooting. Comput Chem Eng 2010;34(11):1873–89.
[24] Schlegel M, Stockmann K, Binder T, Marquardt W. Dynamic optimization using adaptive control vector parameterization. Comput Chem Eng 2005;29(8):1731–51.
[25] Biegler LT, Cervantes AM, Wachter A. Advances in simultaneous strategies for dynamic process optimization. Chem Eng Sci 2002;57(4):575–93.
[26] Biegler LT. An overview of simultaneous strategies for dynamic optimization. Chem Eng Process Process Intens 2007;46(11):1043–53.
[27] Kirches C, Wirsching L, Bock HG, Schlöder JP. Efficient direct multiple shooting for nonlinear model predictive control on long horizons. J Process Control 2012;22(3):540–50.
[28] Leineweber DB, Bauer I, Bock HG, Schlöder JP. An efficient multiple shooting based reduced SQP strategy for large-scale dynamic process optimization. Part 1: theoretical aspects. Comput Chem Eng 2003;27(2):157–66.
[29] Binder T, Cruse A, Villas C, Marquardt W. Dynamic optimization using a wavelet based adaptive control vector parameterization strategy. Comput Chem Eng 2000;24(2–7):1201–7.
[30] Huang D, Luo X. Process transition based on dynamic optimization with the case of a throughput-fluctuating ethylene column. Ind Eng Chem Res 2018;57(18):6292–302.
[31] Srinivasan B, Palanki S, Bonvin D. Dynamic optimization of batch processes: I. Characterization of the nominal solution. Comput Chem Eng 2003;27(1):1–26.
[32] Srinivasan B, Bonvin D, Visser E, Palanki S. Dynamic optimization of batch processes: ii. role of measurements in handling uncertainty. Comput Chem Eng 2003;27(1):27–44.
[33] Lin J, Xu F, Luo X. Dynamic optimization of hybrid parametric system with non-convex sensitivity-based generalized Benders decomposition. Comput. Chem. Eng.(Under Review)
[34] Wang R, Luo X, Xu F. Multiple steady states of FCCU with varying CO concentration. CIESC J 2013;64(8):2930–7.
[35] Wang R, Luo X, Xu F. Effect of co combustion promoters on combustion air partition in fcc under nearly complete combustion. Chin J Chem Eng 2014;22(5):531–7.
[36] Wang R, Luo X, Xu F. Economic and control performance of a fluid catalytic cracking unit: interactions between combustion air and co promoters. Ind Eng Chem Res 2014;53(1):287–304.
[37] Coppinger M, Swain E. Performance prediction of an industrial centrifugal compressor inlet guide vane system. P I Mech Eng A-J Pow 2000;214(2):153–64.
[38] Li X. Parallel nonconvex generalized Benders decomposition for natural gas production network planning under uncertainty. Comput Chem Eng 2013;55:97–108.
[39] Srinivasan B, Bonvin D. Real-time optimization of batch processes by tracking the necessary conditions of optimality. Ind Eng Chem Res 2007;46(2) 492-50.
[40] Gros S, Srinivasan B, Bonvin D. Neighboring extremal controllers for singular problems. In: Proceedings of the 2004 American control conference. IEEE; 2004. p. 34–9.
[41] Ye L, Cao Y, Li Y, Zong Z. Approximating necessary conditions of optimality as controlled variables. Ind Eng Chem Res 2013;52(2):798–808.
[42] François G, Srinivasan B, Bonvin D. Use of measurements for enforcing the necessary conditions of optimality in the presence of constraints and uncertainty. J Process Control 2005;15(6):701–12.
[43] Gros S, Srinivasan B, Bonvin D. Optimizing control based on output feedback. Comput Chem Eng 2009;33(1):191–8.
[44] Ye L, Shen F, Ma X, Zhang H. An active approach to space-reduced NCO tracking and output feedback optimizing control for batch processes with parametric uncertainty. J Process Control 2020;89:30–44.