Supplementary Materials

An Enhanced Logic-Based Benders Decomposition Algorithm with Proximity-principle for Simulator-based Distillation Process Optimization

Chenshan Tian [a], Xiaodong Zhang [b,*], Yang Lan [a], Jinsheng Sun [a,*]

[a] *School of Chemical Engineering and Technology, Tianjin University, Tianjin, 300072, P. R. China*

[b] *College of Smart Energy, Shanghai Jiao Tong University, Shanghai, 200240, P. R. China*

The supplementary materials include five sections, namely, the number of selectors for Theorems and their proofs, Specific parameters in solvers and simulators, Information of case 1, Information of case 2 and Information of case 3.

*Corresponding authors.

E-mail addresses: xiaodongzhang@sjtu.edu.cn (X. Zhang); jssun2006@vip.163.com (J. Sun).

Supporting information Section 1 - Theorems and their proofs

Theorem 1 (Proof of LB-BD Convergence): If the MILP master problem is convex programming and $f_{MILP} \geq f_{NLP}$, the algorithm converges to the global optimal solution, with the optimal objective value being $\min f_{NLP}$.

Proof:

For convex functions, there is definition formula:

$$\begin{gathered}\theta_1 f(\boldsymbol{x_1}) + (1-\theta_1) f(\boldsymbol{x_2}) \geq f(\theta_1 \boldsymbol{x_1} + (1-\theta_1)\boldsymbol{x_2}) \\ 0 \leq \theta_1 \leq 1, \boldsymbol{x_1}, \boldsymbol{x_2} \in \mathbb{D}^n \subset \mathbb{R}^n, f: \mathbb{D}^n \to \mathbb{D}'\end{gathered} \tag{S1}$$

Furthermore, there are:

$$\sum_{k=1}^{\mathrm{n}+1} \theta_k = 1, 0 \leq \theta_k \leq 1 \tag{S2}$$

$$\begin{aligned} f\left(\sum_{k=1}^{\mathrm{n}+1} \theta_k \boldsymbol{x_k}\right) &= f\left((1-\theta_{n+1}) \sum_{k=1}^{\mathrm{n}} \frac{\theta_k}{(1-\theta_{n+1})} \boldsymbol{x_k} + \theta_{n+1} \boldsymbol{x_{n+1}}\right) \\ &\leq (1-\theta_{n+1}) f\left(\sum_{k=1}^{\mathrm{n}} \frac{\theta_k}{(1-\theta_{n+1})} \boldsymbol{x_k}\right) + \theta_{n+1} f(\boldsymbol{x_{n+1}}) \\ &= (1-\theta_{n+1}) f\left(\left(1 - \frac{\theta_n}{(1-\theta_{n+1})}\right) \sum_{k=1}^{\mathrm{n}-1} \frac{\theta_k}{(1-\theta_n-\theta_{n+1})} \boldsymbol{x_k} \right. \\ &\qquad \left. + \frac{\theta_n}{(1-\theta_{n+1})} \boldsymbol{x_n}\right) + \theta_{n+1} f(\boldsymbol{x_{n+1}}) \\ &\leq (1-\theta_n-\theta_{n+1}) f\left(\sum_{k=1}^{\mathrm{n}-1} \frac{\theta_k}{(1-\theta_n-\theta_{n+1})} \boldsymbol{x_k}\right) + \theta_n f(\boldsymbol{x_n}) \\ &\qquad + \theta_{n+1} f(\boldsymbol{x_{n+1}}) \\ &\leq \cdots \leq \sum_{k=1}^{\mathrm{n}+1} \theta_k f(\boldsymbol{x_k}) \end{aligned} \tag{S3}$$

Then:

$$\begin{aligned} f(\boldsymbol{x_{n+1}}) &\geq \frac{1}{\theta_{n+1}} f\left(\sum_{k=1}^{\mathrm{n}+1} \theta_k \boldsymbol{x_k}\right) - \sum_{k=1}^{\mathrm{n}} \frac{\theta_k}{\theta_{n+1}} f(\boldsymbol{x_k}) \\ &= f\left(\sum_{k=1}^{\mathrm{n}+1} \theta_k \boldsymbol{x_k}\right) + \sum_{k=1}^{\mathrm{n}} \left(-\frac{\theta_k}{\theta_{n+1}}\right) \left(f(\boldsymbol{x_k}) - f\left(\sum_{k=1}^{\mathrm{n}+1} \theta_k \boldsymbol{x_k}\right)\right) \end{aligned} \tag{S4}$$

Where:

$$\boldsymbol{x_{n+1}} = \sum_{k=1}^{n+1} \frac{\theta_k}{\theta_{n+1}} \boldsymbol{x_k} + \sum_{k=1}^{n} \left(-\frac{\theta_k}{\theta_{n+1}} \right) \boldsymbol{x_k} \tag{S5}$$

Specially, replace $\sum_{k=1}^{n+1} \frac{\theta_k}{\theta_{n+1}} \boldsymbol{x_k}$ with $\boldsymbol{y_n^*}$ and $\boldsymbol{x_k}$ with $(\boldsymbol{y_n^*} - \boldsymbol{n_p})$ or $(\boldsymbol{y_n^*} + \boldsymbol{n_p})$ in the above formula, and the right-hand of the above formula will become the equation (13), and further replace f with g_m, that will become equation (14), so if the MILP master problem is convex programming without the constraint of formula (12). Formula (7-9) are equivalent to the following equation:

$$f_{MILP} = \min_{y} \left\{ \max_{n} \left(K_n(\boldsymbol{y}, \boldsymbol{y_n^*}, \boldsymbol{n_p}, B_{p,q}, q) + M \sum_{m \in \mathcal{N}_1} W_{m,n}(\boldsymbol{y}, \boldsymbol{y_n^*}, \boldsymbol{n_p}, B_{p,q}, q) \right) \right\} \tag{S6}$$

f_{MILP} must be less than or equal to f_{NLP}. Similar theories can also refer to the proof process of LB-OA algorithm[34]. When the formula (12) is included, the integer combinations of f_{NLP} will not become the solution of f_{MILP}.

$$\begin{gathered} if\ \exists \boldsymbol{y_{n+1}^*}, \boldsymbol{y^{lb}} \leq \boldsymbol{y_{n+1}^*} \leq \boldsymbol{y^{ub}} \wedge \boldsymbol{y_{n+1}^*} \neq \boldsymbol{y_k^*}\ (k = \forall \mathbb{N}^+ \wedge k \leq n) \\ \text{s.t.}\ f_{MILP}(\boldsymbol{y_{n+1}^*}) \geq \min_{k \in \mathcal{N}_3} f_{NLP}(\boldsymbol{y_k^*}) \end{gathered} \tag{S7}$$

Existence of constant formula:

$$f_{NLP}(\boldsymbol{y_{n+1}^*}) \geq f_{MILP}(\boldsymbol{y_{n+1}^*}) \geq \min_{k \in \mathcal{N}_3} f_{NLP}(\boldsymbol{y_k^*}) = \min_{k \in \{\mathcal{N}_3, n+1\}} f_{NLP}(\boldsymbol{y_k^*})$$

$$\mathcal{N}_1 = \{1,2, \dots, \text{Number of constraints}\} \tag{S8}$$

$$\mathcal{N}_3 = \{1,2, \dots, \text{Number of known points in MILP}\}$$

The algorithm converges to the global optimal solution, with the optimal objective value being $\min f_{NLP}$.

Inference 1: If the MILP master problem is convex programming, as the number of iterations increases, the objective function value of MILP shows a single increasing trend.

Proof:

As the iteration progresses, $f_{MILP,n}$ represents the objective function result of the n-th generation MILP,

Due to the limitations of formula (12), there must be:

$$\boldsymbol{y^{lb}} \leq \boldsymbol{y^{*}_{n+1}} \leq \boldsymbol{y^{ub}} \wedge \boldsymbol{y^{*}_{n+1}} \neq \boldsymbol{y^{*}_{k}} \ (k = \forall \mathbb{N}^{+} \wedge k \leq n) \tag{S9}$$

If the MILP master problem is convex programming, then there will be:

$$\begin{aligned} f_{MILP,n+1} &= \min_{y}\left\{\max_{n+1}\left(K_n\left(\boldsymbol{y}, \boldsymbol{y^{*}_{n}}, \boldsymbol{n_p}, B_{p,q}, q\right) + M\sum_{m\in N_1} W_{m,n}\left(\boldsymbol{y}, \boldsymbol{y^{*}_{n}}, \boldsymbol{n_p}, B_{p,q}, q\right)\right)\right\} \\ &\geq \min_{y}\left\{\max_{n}\left(K_n\left(\boldsymbol{y}, \boldsymbol{y^{*}_{n}}, \boldsymbol{n_p}, B_{p,q}, q\right) + M\sum_{m\in N_1} W_{m,n}\left(\boldsymbol{y}, \boldsymbol{y^{*}_{n}}, \boldsymbol{n_p}, B_{p,q}, q\right)\right)\right\} \\ &= f_{MILP,n} \end{aligned} \tag{S10}$$

As the number of iterations increases, the objective function value of MILP shows a single increasing trend.

Theorem 2: With same known points $\boldsymbol{y_n^*}$, The f_{MILP} solved by formulas (24-30) must be less than or equal to the f_{MILP} solved by formulas (7-11).

Proof:

For formula (8), assuming $\boldsymbol{y_{BD}}$ is the solution of MILP master problem:

$$\alpha_{BD}(\boldsymbol{y_{BD}}) = \text{minmax}\{K_n(\boldsymbol{y}, \boldsymbol{y_n^*}, \boldsymbol{n_p}, B_{p,q}, q)\}, n \in \mathcal{N}_3$$
$$\mathcal{N}_3 = \{1,2,\dots,\text{Number of known points in MILP}\} \tag{S11}$$

For equations (25) and (27), assuming $\boldsymbol{y_{PBD}}$ is the solution of MILP master problem:

$$\exists n_1 \in \mathcal{N}_3, s.t. \forall n_2 \in \mathcal{N}_3, \sum_{p\in\mathcal{N}_2}\left(\frac{y_{p,PBD}-y_{p,n_1}}{y_p^{ub}-y_p^{lb}}\right)^2 \le \sum_{p\in\mathcal{N}_2}\left(\frac{y_{p,PBD}-y_{p,n_2}}{y_p^{ub}-y_p^{lb}}\right)^2$$
$$\alpha_{PBD}(\boldsymbol{y_{PBD}}) = K_{n_1}(\boldsymbol{y}, \boldsymbol{y_{n_1}^*}, \boldsymbol{n_p}, B_{p,q}, q) \le \alpha_{BD}(\boldsymbol{y_{BD}}) \tag{S12}$$
$$\mathcal{N}_2 = \{1,2,\dots,\dim(\boldsymbol{y})\}$$

Similarly, there is:

$$\beta_{m,PBD} \le \beta_{m,BD} \tag{S13}$$

Then:

$$f_{MILP,PBD} = \alpha_{PBD} + M\sum_{m\in\mathcal{N}_1}\beta_{m,PBD} \le \alpha_{BD} + M\sum_{m\in\mathcal{N}_1}\beta_{m,BD} = f_{MILP,BD}$$
$$\mathcal{N}_1 = \{1,2,\dots,\text{Number of constraints}\} \tag{S14}$$

Inference 2: For cases where the global optimal solution of MILP master problem is within a local convex domain, as long as each local convex domain has at least one known integer combination, LB-PBD can definitely find the global optimal solution.

Proof:

For cases where the global optimal solution f_{NLP_k} is within a local convex domain, if each local convex domain has at least one known integer combination, there will be a f_{MILP,PBD_n} and a f_{NLP_n} for every local convex: (n is the number of local convex)

$$f_{MILP,PBD} = \min(f_{MILP,PBD_n}) \leq f_{MILP,PBD_k} \leq f_{NLP_k} = \min(f_{NLP_n}) \tag{S15}$$

So, under the constraints of (31) and similar as **Theorem 1**, if $f_{MILP,PBD} \geq \min(f_{NLP})$ occurs, the algorithms must have found the global optimal solution.

Supporting information Section 2 - Specific parameters in solvers and simulators

Specifically, FMINCON uses the inlier method with the following special settings and default parameters for all others:

1. Due to the use of a penalty function in the interior point method to evaluate the degree of constraint violation. Normalization is performed based on the upper and lower bounds after each simulator evaluation, at the same time, a fixed scaling ratio is used to balance the objective function value size and constraint functions value size, obviously, the corresponding derivative also needs to be scaled accordingly. While inverse normalization is performed based on the upper and lower bounds before each simulator evaluation. In our long-term experiments, we have found that this improvement is very necessary because of the significant absolute difference between the independent variables, objective function value and constraint function values.

2. For the objective function values that have not converged in the simulator, set NaN, and the solver will automatically reduce the step size until a feasible solution is found. This scheme has been proven to be very effective for simulators[23].

3. The minimum relative step $\frac{|x_{i+1}-x_i|}{1+|x_i|}$ is set to 1e-4 to prevent meaningless searches.

4. Robustness setting: If the initial values given to the simulator cannot converge, the continuous variables will be automatically updated randomly, and the solution will be restarted until the NLP algorithm can proceed, with a maximum of 10 attempts to update. Otherwise, the target values and constraint target values will be directly set to NaN and the NLP solution will be stopped.

We use the formula (S16-S17) to obtain derivative information in ASPEN PLUS. After expansion, the left matrix is obtained through Open Object Model Framework (OOMF)[12,13], and the right matrix is obtained through numerical or analytical methods.

ASPEN needs to build a calculation equation system (Synchronize EO) based on the calculation results (convergence success or failure) of the sequential modular method (SM) before using the equation oriented method (EO) to calculate. The coefficients extracted from this equation system by OOMF can directly calculate the sensitivity of the left matrix (EO Sensitivity), which is much faster than performing a complete SM calculation. The main time-consuming process will come from communication between software, as OOMF must use txt scripts or ACSII files for data transfer. But overall, using this method to obtain derivatives is much faster than perturbation method, and the derivative obtained from conventional modules (such as distillation tower, flash evaporator, etc.) through this method is an analytical solution.

$$\frac{df}{d\vec{\boldsymbol{x}}} = \begin{bmatrix} \dfrac{df}{dx_1} \\ \dfrac{df}{dx_2} \\ \vdots \\ \dfrac{df}{dx_{\dim(\vec{\boldsymbol{x}})}} \end{bmatrix}$$

$$= \begin{bmatrix} \dfrac{\partial u_1}{\partial x_1} & \dfrac{\partial u_2}{\partial x_1} & \cdots & \dfrac{\partial u_{\dim(\vec{\boldsymbol{u}})}}{\partial x_1} \\ \dfrac{\partial u_1}{\partial x_2} & \dfrac{\partial u_2}{\partial x_2} & \cdots & \dfrac{\partial u_{\dim(\vec{\boldsymbol{u}})}}{\partial x_2} \\ \vdots & & \ddots & \vdots \\ \dfrac{\partial u_1}{\partial x_{\dim(\vec{\boldsymbol{x}})}} & \dfrac{\partial u_2}{\partial x_{\dim(\vec{\boldsymbol{x}})}} & \cdots & \dfrac{\partial u_{\dim(\vec{\boldsymbol{u}})}}{\partial x_{\dim(\vec{\boldsymbol{x}})}} \end{bmatrix} \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} \begin{bmatrix} \dfrac{\partial f}{\partial u_1} \\ \dfrac{\partial f}{\partial u_2} \\ \vdots \\ \dfrac{\partial f}{\partial u_{\dim(\vec{\boldsymbol{u}})}} \\ \dfrac{\partial f}{\partial x_1} \\ \dfrac{\partial f}{\partial x_2} \\ \vdots \\ \dfrac{\partial f}{\partial x_{\dim(\vec{\boldsymbol{x}})}} \end{bmatrix} \tag{S16}$$

$$\frac{d\,\vec{\boldsymbol{G}}}{d\vec{\boldsymbol{x}}} = \begin{bmatrix} \dfrac{dg_1}{dx_1} & \dfrac{dg_2}{dx_1} & \cdots & \dfrac{dg_{\dim(\vec{\boldsymbol{G}})}}{\partial x_1} \\ \dfrac{dg_1}{dx_2} & \dfrac{dg_2}{dx_2} & \cdots & \dfrac{dg_{\dim(\vec{\boldsymbol{G}})}}{dx_2} \\ \vdots & & \ddots & \vdots \\ \dfrac{dg_1}{dx_{\dim(\vec{\boldsymbol{x}})}} & \dfrac{dg_2}{dx_{\dim(\vec{\boldsymbol{x}})}} & \cdots & \dfrac{dg_{\dim(\vec{\boldsymbol{G}})}}{dx_{\dim(\vec{\boldsymbol{x}})}} \end{bmatrix}$$

$$= \begin{bmatrix} \dfrac{\partial u_1}{\partial x_1} & \dfrac{\partial u_2}{\partial x_1} & \cdots & \dfrac{\partial u_{\dim(\vec{\boldsymbol{u}})}}{\partial x_1} \\ \dfrac{\partial u_1}{\partial x_2} & \dfrac{\partial u_2}{\partial x_2} & \cdots & \dfrac{\partial u_{\dim(\vec{\boldsymbol{u}})}}{\partial x_2} \\ \vdots & & \ddots & \vdots \\ \dfrac{\partial u_1}{\partial x_{\dim(\vec{\boldsymbol{x}})}} & \dfrac{\partial u_2}{\partial x_{\dim(\vec{\boldsymbol{x}})}} & \cdots & \dfrac{\partial u_{\dim(\vec{\boldsymbol{u}})}}{\partial x_{\dim(\vec{\boldsymbol{x}})}} \end{bmatrix} \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} \begin{bmatrix} \dfrac{\partial g_1}{\partial u_1} & \cdots & \dfrac{\partial g_{\dim(\vec{\boldsymbol{G}})}}{\partial u_1} \\ \dfrac{\partial g_1}{\partial u_2} & & \dfrac{\partial g_{\dim(\vec{\boldsymbol{G}})}}{\partial u_2} \\ \vdots & & \vdots \\ \dfrac{\partial g_1}{\partial u_{\dim(\vec{\boldsymbol{u}})}} & \cdots & \dfrac{\partial g_{\dim(\vec{\boldsymbol{G}})}}{\partial u_{\dim(\vec{\boldsymbol{u}})}} \\ \dfrac{\partial g_1}{\partial x_1} & & \dfrac{\partial g_{\dim(\vec{\boldsymbol{G}})}}{\partial x_1} \\ \dfrac{\partial g_1}{\partial x_2} & & \dfrac{\partial g_{\dim(\vec{\boldsymbol{G}})}}{\partial x_2} \\ \vdots & & \vdots \\ \dfrac{\partial g_1}{\partial x_{\dim(\vec{\boldsymbol{x}})}} & & \dfrac{\partial g_{\dim(\vec{\boldsymbol{G}})}}{\partial x_{\dim(\vec{\boldsymbol{x}})}} \end{bmatrix} \tag{S17}$$

GUROBI uses the following special settings and default parameters for all others:

1. Adopt the highest level of solution accuracy (NumericFocus=3) to avoid numerical problems as much as possible.

2. Based on our observations and experiments[30], the Big-M value only needs to be large enough, unless otherwise specified, the Big-M value is set to be two orders of magnitude larger than the maximum value that may occur in the equation. If it is too high, it will have almost no effect on the convergence of the algorithm, but it will affect the solving time.

3. For the objective function and constraint functions values that have not converged in the simulator, set to the sufficiently large values to ensure that it is always "the worst solution", while ensuring that the Big-M value in the previous statement is meaningful.

Table S1 The binary interaction parameters used by simulators

Comp.i	Comp.j	Aij	Aji	Bij	Bji	Cij
ANE	ENE	0	0	-222.007	279.768	0.3
ANE	DMAC	0	0	699.1038	197.4781	0.3
ENE	DMAC	0	0	572.0253	102.6018	0.3
CH_4O	H_2O	-0.693	2.7322	172.9871	-617.269	0.3
CH_4O	C_2H_6O	4.7119	-2.3127	-1162.29	483.8436	0.3
C_2H_6O	H_2O	-0.8009	3.4578	246.18	-586.081	0.3

ANE is the cyclohexane, ENE is the cyclohexene, DMAC is the dimethylacetamide, CH_4O is the methanol, C_2H_6O is the ethanol and H_2O is the water.

Supporting information Section 3 – Information of case 1

Table S2 Algorithm numerical test function (1) results

Algorithm name	Accuracy	Average number of iterations	Average number of function evaluations
SPDDE($m = 400, k = 40$)	0.00	2.84	16000
SPDDE($m = 100, k = 160$)	0.00	3.38	16000
SPDDE($m = 100, k = 4000$)	0.00	3.47	400000
LB-OA	0.98	3.00	365.3
LB-BD($i = 1, h = 1$)	0.20	6.53	3322.4
LB-BD($i = 1, h = 3$)	0.05	3.26	2620.1
LB-BD($i = 1, h = 5$)	0.03	2.15	3204.9
LB-BD($i = 1, h = 7$)	0.06	2.02	4363
LB-BD($i = 1, h = 9$)	0.05	2.00	5586.9
LB-BD($i = 1, h = 11$)	0.04	2.00	6810.9
LB-BD($i = 1, h = 13$)	0.04	2.00	8022.7
LB-BD($i = 1, h = 15$)	0.06	2.00	9250.7
LB-BD($i = 3, h = 1$)	0.25	8.38	3802.3
LB-BD($i = 3, h = 3$)	0.10	5.29	3524.8
LB-BD($i = 3, h = 5$)	0.03	4.35	4299.2
LB-BD($i = 3, h = 7$)	0.02	4.03	5376.4
LB-BD($i = 3, h = 9$)	0.05	4.00	6557.6
LB-BD($i = 3, h = 11$)	0.06	4.00	7802.2
LB-BD($i = 3, h = 13$)	0.02	4.10	9006
LB-BD($i = 3, h = 15$)	0.08	4.13	10252
LB-BD($i = 5, h = 1$)	0.24	10.75	4328.6
LB-BD($i = 5, h = 3$)	0.10	7.35	4083.9
LB-BD($i = 5, h = 5$)	0.04	6.33	4873.5
LB-BD($i = 5, h = 7$)	0.03	6.01	5951.9
LB-BD($i = 5, h = 9$)	0.06	6.06	7209
LB-BD($i = 5, h = 11$)	0.05	6.05	8366.7
LB-BD($i = 5, h = 13$)	0.11	6.07	9654.9
LB-BD($i = 5, h = 15$)	0.05	6.04	10781
LB-PBD($i = 1, h = 1$)	0.76	22.24	12602

LB-PBD($i=1, h=3$)	0.76	18.94	11872
LB-PBD($i=1, h=5$)	0.84	19.10	13177
LB-PBD($i=1, h=7$)	0.90	18.54	14073
LB-PBD($i=1, h=9$)	0.87	16.03	13853
LB-PBD($i=1, h=11$)	0.86	14.86	14398
LB-PBD($i=1, h=13$)	0.91	13.99	15159
LB-PBD($i=1, h=15$)	0.87	13.05	15792
LB-PBD($i=3, h=1$)	0.76	27.76	14619
LB-PBD($i=3, h=3$)	0.65	18.97	11128
LB-PBD($i=3, h=5$)	0.84	19.53	12627
LB-PBD($i=3, h=7$)	0.81	18.62	13411
LB-PBD($i=3, h=9$)	0.84	18.21	14310
LB-PBD($i=3, h=11$)	0.90	17.50	15130
LB-PBD($i=3, h=13$)	0.88	17.29	16059
LB-PBD($i=3, h=15$)	0.89	15.85	16495
LB-PBD($i=5, h=1$)	0.76	31.35	15301
LB-PBD($i=5, h=3$)	0.67	21.91	11689
LB-PBD($i=5, h=5$)	0.78	22.20	13086
LB-PBD($i=5, h=7$)	0.81	21.30	13885
LB-PBD($i=5, h=9$)	0.88	21.12	15054
LB-PBD($i=5, h=11$)	0.89	20.26	15712
LB-PBD($i=5, h=13$)	0.84	19.32	16624
LB-PBD($i=5, h=15$)	0.89	19.14	17487

Ps: i is the hyperparameter i of the delayed convergence strategy in Section 2.3 of main text. h is the hyperparameter of the multi start points strategy in Section 2.3 of main text.

m is the maximum iterative algebra in SPDDE and k is the corresponding population size for each generation. Average number of iterations is the iteration number where the final solution first appears in SPDDE.

Table S3 Algorithm numerical test function (2) results

Algorithm name	Accuracy	Average number of iterations	Average number of function evaluations
SPDDE($m = 2000, k = 120$)	0.24	203.11	240000
SPDDE($m = 500, k = 480$)	0.81	237.15	240000
SPDDE($m = 250, k = 960$)	0.99	245.14	240000
LB-OA($i = 3, h = 1$)	0.03	4.93	9525
LB-OA($i = 10, h = 1$)	0.01	11.96	26637
LB-OA($i = 15, h = 1$)	0.01	16.94	38771
LB-BD($i = 3, h = 1$)	0.00	5.02	35961
LB-BD($i = 3, h = 8$)	0.01	4.05	112222
LB-BD($i = 3, h = 15$)	0.05	4.05	192780
LB-BD($i = 10, h = 1$)	0.01	13.47	81439
LB-BD($i = 10, h = 8$)	0.07	11.96	150413
LB-BD($i = 10, h = 15$)	0.05	11.53	229450
LB-BD($i = 15, h = 1$)	0.01	22.20	114470
LB-BD($i = 15, h = 8$)	0.08	17.71	172350
LB-BD($i = 15, h = 15$)	0.04	17.89	252103
LB-PBD($i = 3, h = 1$)	0.35	21.43	217372
LB-PBD($i = 3, h = 8$)	0.65	26.04	352762
LB-PBD($i = 3, h = 15$)	0.68	22.51	395726
LB-PBD($i = 10, h = 1$)	0.67	39.57	328624
LB-PBD($i = 10, h = 8$)	0.7	36.51	426270
LB-PBD($i = 10, h = 15$)	0.83	32.23	460920
LB-PBD($i = 15, h = 1$)	0.80	51.16	391165
LB-PBD($i = 15, h = 8$)	0.84	47.13	484376
LB-PBD($i = 15, h = 15$)	0.85	42.29	523734

Ps: i is the hyperparameter i of the delayed convergence strategy in Section 2.3 of main text. h is the hyperparameter h of the multi start points strategy in Section 2.3 of main text.

m is the maximum iterative algebra in SPDDE and k is the corresponding population size for each generation. Average number of iterations is the iteration number where the final solution first appears in SPDDE.

Table S4 Big-M test results

Test Function	Value of M_1 in formula (25-26)	Accuracy	Average number of iterations	Average number of function evaluations
(1)	10	0.74	14.34	11468
	10^2	0.83	18.38	13821
	10^3	0.86	18.35	13799
	10^4	0.86	19.10	14141
	10^5	0.84	18.68	14039
(2)	10^3	0.33	12.66	204516
	10^4	0.58	25.39	346638
	10^5	0.65	26.04	352762
	10^6	0.62	25.83	350562
	10^7	0.66	25.75	352033

Ps: The optimization algorithm adopts LB-PBD, the hyperparameter i of the delayed convergence strategy is set 3 and the hyperparameter h of the multi start points strategy is set 8.

Supporting information Section 4 - Information of case 2

In order to ensure that NLP under each integer combination can obtain the global optimal solution in **Figure 8** of case 2, each NLP solution needs to be repeated 10 times. The first initial value is a random value, and the subsequent initial values are the previous solution. Since the target value corresponding to the algorithm's solution is always decreasing, the solution obtained in this way can be used as the global optimal solution. There are two phenomena that can prove this statement: first, the solution of a general integer combination no longer changes after 3-4 repetitions of solving (within the tolerance range); second, from **Figure 8** in the main text, it can be seen that the results of most points show obvious smoothness, which conforms to general physical laws.

Table S5 Calculation formula for total annualized cost (TAC)

Item	Calculation formula
Distillation columns	
Distillate molar flow rate D (kmol/h)	Independent variable
Mole reflux ratio R	Independent variable
Column absolute pressure P (atm)	Independent variable
The number of theoretical stages N	Independent variable
The degrees subcooled of condenser T_{sc} (K)	fixed value
Tray pressure drop P_{tray} (atm)	fixed value
Column overhead temperature T_D (K)	Aspen RadFrac results
Column bottom temperature T_R (K)	Aspen RadFrac results
Heat duty of condensers Q_C (kW)	Aspen RadFrac results
Heat duty of reboilers Q_R (kW)	Aspen RadFrac results
Column diameter D_c (m)[35]	$D_c = \left[\frac{4\cdot 22.4\cdot D\cdot (R+1)\cdot T_D}{3600\cdot 273\cdot 0.761\cdot \pi\cdot P^{0.5}}\right]^{0.5}$
Column height H_c (m)[36]	$H_c = 0.61\frac{N}{0.8} + 4.27$
Column shell cost C_{col} (US$)[36]	$C_{col} = 17640 \cdot D_c^{1.066} \cdot H_c^{0.802}$
Column tray cost C_{tray} (US$)[36]	$C_{tray} = 229 \cdot D_c^{1.55} \cdot \frac{N}{0.8}$
Vacuum systems	
Air leakage amount M (lb/h)[37]	$M = 5 + \{0.0298 + 0.03088\ln(760P) - 0.0005733[\ln(760P)]^2\} \cdot \left(\frac{\pi/4\cdot D_c^2\cdot H_c}{0.0283168}\right)^{0.66}$
Capital cost of vacuum system C_{vs} (US$)[37]	$C_{vs} = 1690 \cdot \left(\frac{M}{760P}\right)^{0.41}$
Vacuum system operating cost C_{vsopt} (US$/year)[37]	$C_{vsopt} = 400 \cdot M$
Condensers	
Logarithmic mean temperature difference of condensers $LMTD_{cond}$ (K)	$LMTD_{cond} = \frac{(T_D - T_{cond}^{out}) - (T_D - T_{sc} - T_{cond}^{in})}{\ln\left((T_D - T_{cond}^{out})/(T_D - T_{sc} - T_{cond}^{in})\right)}$

Capital cost of condensers C_{cond} (US$)[38]	$C_{cond} = 7296\left(\frac{Q_C}{0.852 \cdot LMTD_{cond}}\right)^{0.65}$
Reboilers	
Logarithmic mean temperature difference of reboilers $LMTD_{reb}$ (K)	$LMTD_{reb} = T_{steam} - T_R$
Capital cost of reboilers C_{reb} (US$)[38]	$C_{reb} = 7296\left(\frac{Q_C}{0.568 \cdot LMTD_{reb}}\right)^{0.65}$
Process to process heat exchangers	
Heat-transfer coefficient U_{P-P} (kW/(K·m^2))[38]	$U_{P-P} = 0.17$ (gas-gas) $U_{P-P} = 0.28$ (phase change-gas) $U_{P-P} = 0.57$ (phase change-phase change or liquid-liquid or liquid-gas) $U_{P-P} = 0.85$ (liquid- phase change)
Heat duty of heat exchangers Q_{P-P} (kW)	Aspen HeatX results
Cold stream inlet temperature T_{cold}^{in} (K)	Aspen HeatX results
Cold stream outlet temperature T_{cold}^{out} (K)	Aspen HeatX results
Hot stream inlet temperature T_{hot}^{in} (K)	Aspen HeatX results
Hot stream outlet temperature T_{hot}^{out} (K)	Aspen HeatX results
Logarithmic mean temperature difference of heat exchangers $LMTD_{P-P}$ (K)	$LMTD_{P-P} = \frac{(T_{hot}^{in} - T_{cold}^{out}) - (T_{hot}^{out} - T_{cold}^{in})}{\ln\left((T_{hot}^{in} - T_{cold}^{out})/(T_{hot}^{out} - T_{cold}^{in})\right)}$
Capital cost of heat exchangers C_{P-P} (US$)[38]	$C_{P-P} = 7296\left(\frac{Q_{P-P}}{U_{P-P} \cdot LMTD_{P-P}}\right)^{0.65}$
Compressors	
Work done of compressors W_{com} (kW)[39]	Aspen Compr results
Capital cost of compressors C_{com} (US$)[39]	$C_{com} = 9560 \cdot W_{com}^{0.82}$
Energy cost	
Low pressure steam cost U_{steam} (US$/GJ)[38]	$U_{steam} = 7.78$
Low pressure steam temperature $T_{steam} = 160℃$ with pressure 6 bar	
Medium pressure steam cost U_{steam}(US$/GJ)[38]	$U_{steam} = 8.22$
Medium pressure steam temperature $T_{steam} = 184℃$ with pressure 11 bar	
High pressure steam cost U_{steam} (US$/GJ)[38]	$U_{steam} = 9.88$
High pressure steam temperature $T_{steam} = 254℃$ with pressure 42 bar	
Cooling water cost U_{cond} (US$/GJ)[38]	$U_{cond} = 0.354$
Cooling water inlet temperature $T_{cond}^{in} = 25℃$ with outlet temperature $T_{cond}^{out} = 35℃$	
Chilled water cost U_{cond} (US$/GJ)[38]	$U_{cond} = 4.43$
Chilled water inlet temperature $T_{cond}^{in} = 5℃$ with outlet temperature $T_{cond}^{out} = 15℃$	
Refrigerant cost U_{cond} (US$/GJ)[38]	$U_{cond} = 7.89$
Refrigerant inlet temperature $T_{cond}^{in} = -20℃$ with outlet temperature $T_{cond}^{out} = -10℃$	
Deep refrigerant cost U_{cond} (US$/GJ)[38]	$U_{cond} = 13.11$
Deep refrigerant inlet temperature $T_{cond}^{in} = -50℃$ with outlet temperature $T_{cond}^{out} = -40℃$	
Electricity cost U_{elec} (US$/GJ)[39]	$U_{elec} = 16.9$
Payback period PBY (years)	$PBY = 3$
Annual operating time AOT (hours)	$AOT = 8000$
Total annualized cost (TAC) (US$/year)	

$$TAC = \frac{C_{col} + C_{tray} + C_{vs} + C_{cond} + C_{reb} + C_{P-P} + C_{com}}{PBY} + \left(\sum C_{vsopt} + \sum U_{steam} Q_{steam} + \sum U_{cond} Q_{cond} + \sum U_{elec} W_{com}\right) AOT$$

PS: The degrees subcooled of condenser $T_{sc} = 5\text{K}$ in case 2 and 3.

The reboiler adopts a thermosiphon type, and the outlet condition adopts a vapor fraction of 0.15.

Tray pressure drop $P_{tray} = 0.007\text{atm}$ in case 2 and $P_{tray} = 0.3\text{kPa}$ in case 3.

In case 2, if the bottom pressure of the column is lower than 1 atm, a vacuum system is required, otherwise it is not necessary; while in case 3, if the top pressure of the column is below 1atm, a vacuum system is required, otherwise it is not necessary.

Table S6 The upper and lower bounds of the independent variable in Case 2

Item	Variable Range
Distillate molar flow rate D (kmol/h)	[99.7998,100.1001]*
Mole reflux ratio R	[3,20]
Column absolute pressure P (atm)	[0.2,10]
The number of theoretical stages in rectifying section N_1	[10,150]
The number of theoretical stages in stripping section N_2	[10,150]

The range of variables with * is obtained by quadratic constraint programming (QCP) calculation as follows:

$$D^{ub} = \max D\,, D^{lb} = \min D \tag{S18}$$

Subject to:

$$D + B = F \tag{S19}$$

$$Dx_{D,A} + Bx_{B,A} = Fx_{F,A} \tag{S20}$$

$$D \leq F \tag{S21}$$

$$B \leq F \tag{S22}$$

$$0.999 \leq x_{D,A} \leq 1, 0 \leq x_{B,A} \leq 0.001, D \geq 0, B \geq 0$$

Where, the feed flow F to the distillation column is 300 kmol/h with methanol, ethanol, and water each accounting for one-third. Use A, B, and C respectively to represent methanol, ethanol, and water, where the molar concentration of methanol in the feed is $x_{F,A}$, the molar concentration

of methanol in the top of the column is $x_{D,A}$, and the molar concentration of methanol in the bottom of the column is $x_{B,A}$. The same applies to other substances. The range of distillate molar flow rate D is the $[D^{lb}, D^{ub}]$.

Table S7 Test results of different algorithms

Algorithm name	Accuracy	Average CPU consumption/seconds	Average number of iterations (average number of integer combination calculations)
LB-BD($i=1,h=1$)	0/30	0.08	2.17(10.9)
LB-BD($i=1,h=5$)	0/30	0.13	2.03(30.2)
LB-BD($i=1,h=15$)	0/30	0.20	2.00(80.00)
LB-BD($i=3,h=1$)	0/30	0.17	4.20(21.0)
LB-BD($i=3,h=5$)	0/30	0.28	4.33(41.7)
LB-BD($i=3,h=15$)	0/30	0.42	4.00(90.0)
LB-BD($i=5,h=1$)	0/30	0.22	6.20(31.0)
LB-BD($i=5,h=5$)	0/30	0.38	6.23(51.2)
LB-BD($i=5,h=15$)	0/30	0.61	6.00(100.0)
LB-PBD ($i=1,h=1,K=1$)	29/30	21.16	27.67(138.4)
LB-PBD ($i=1,h=5,K=1$)	28/30	30.59	26.23(151.2)
LB-PBD ($i=1,h=15,K=1$)	30/30	42.95	20.80(174.0)
LB-PBD ($i=1,h=5,K=3$)	9/30	2.09	6.90(54.5)
LB-PBD ($i=1,h=15,K=3$)	20/30	11.48	9.83(119.2)
LB-PBD ($i=2,h=1,K=1$)	30/30	47.28	32.80(164.0)
LB-PBD ($i=2,h=5,K=1$)	30/30	27.29	26.27(151.4)
LB-PBD ($i=2,h=15,K=1$)	30/30	54.21	23.77(188.9)
LB-PBD ($i=2,h=5,K=3$)	22/30	6.65	13.57(87.9)
LB-PBD ($i=2,h=15,K=3$)	26/30	20.34	14.03(140.2)
LB-PBD ($i=3,h=1,K=1$)	30/30	29.29	29.10(145.5)
LB-PBD ($i=3,h=5,K=1$)	30/30	25.94	25.03(145.2)
LB-PBD ($i=3,h=15,K=1$)	30/30	50.43	23.50(187.5)

LB-PBD ($i=3,h=5,K=3$)	22/30	9.43	15.60(98.0)
LB-PBD ($i=3,h=15,K=3$)	30/30	30.63	18.17(160.85)
LB-PBD ($i=4,h=1,K=1$)	30/30	27.88	29.87(149.4)
LB-PBD ($i=4,h=5,K=1$)	30/30	33.76	28.60(163.0)
LB-PBD ($i=4,h=15,K=1$)	30/30	57.13	25.07(195.4)
LB-PBD ($i=4,h=5,K=3$)	22/30	14.52	18.53(112.7)
LB-PBD ($i=4,h=15,K=3$)	30/30	37.12	19.27(166.4)
LB-PBD ($i=5,h=1,K=1$)	30/30	39.70	32.87(164.4)
LB-PBD ($i=5,h=5,K=1$)	30/30	42.23	30.33(171.7)
LB-PBD ($i=5,h=15,K=1$)	30/30	60.63	26.53(202.65)
LB-PBD ($i=5,h=5,K=3$)	28/30	29.85	24.03(140.2)
LB-PBD ($i=5,h=15,K=3$)	30/30	49.52	22.17(180.9)

i is the hyperparameter i of the delayed convergence strategy in Section 2.3 of main text. h is the hyperparameter h of the multi start points strategy in Section 2.3 of main text. K is the hyperparameter K of the formula (28) in Section 2.2 of main text. The average CPU time does not include the computation time of the simulator.

Supporting information Section 5 - Information of case 3

The process flow diagram of Case 3 is shown in **Figure S1**, where all independent variables are marked. The feed is a mixture of 7405.422 kg/h cyclohexane (ANE) and 28123.3 kg/h cyclohexene (ENE), and the circulating stream is strictly connected without concentration restrictions. In order to reduce the number of non-convergent solutions and improve the robustness of the algorithm, we adopt the following strategy to handle non-convergent solutions: when the process convergence fails, consider changing the convergence algorithm of each column until convergence is successful or all algorithm combinations fail to converge. Therefore, the single evaluation time will significantly increase, thereby increasing the time consumption of NLP sub-problems. In order to reduce the overall time, NLP for each fixed integer combination will not be repeated.

The optimization goal of the process is to minimize TAC (calculated as shown in Table S5). The top of the extraction column requires the extraction of cyclohexane with a molar concentration no less than 99.9%, while the top of the regeneration column requires the extraction of cyclohexene with a molar concentration no less than 99.9%.

For comparison, SPDDE parameters are set as follows: the population size 130, maximum scale factor 0.9, minimum scale factor 0.1, maximum crossover rate 0.9, minimum crossover rate 0.1 and the maximum evolution generation is 500 generations. The results of SPDDE are shown in **Figure S1**, and the iteration diagram of SPDDE is shown in **Figure S2**.

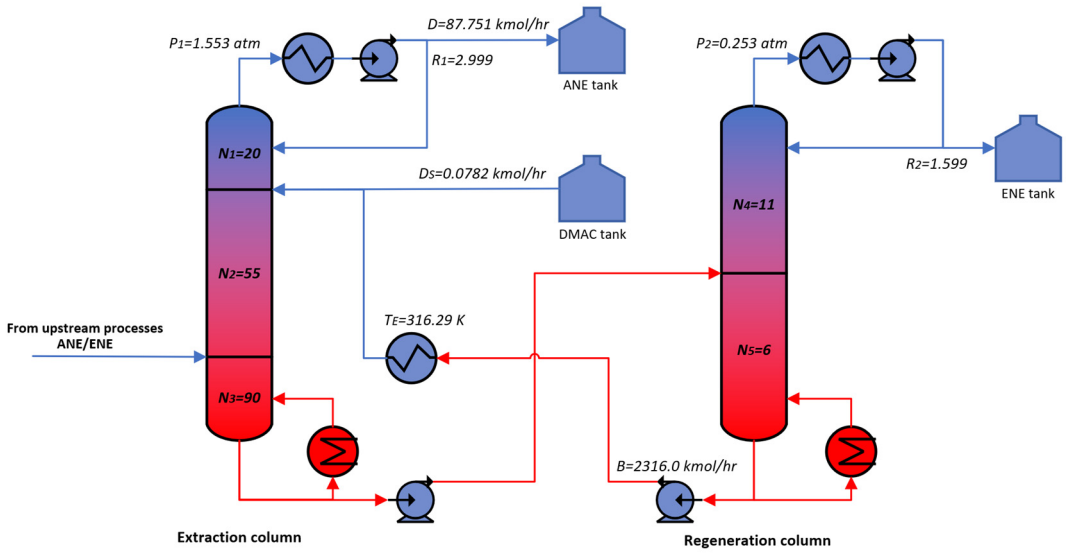


Figure S1 Process flow diagram of SPDDE for case 3

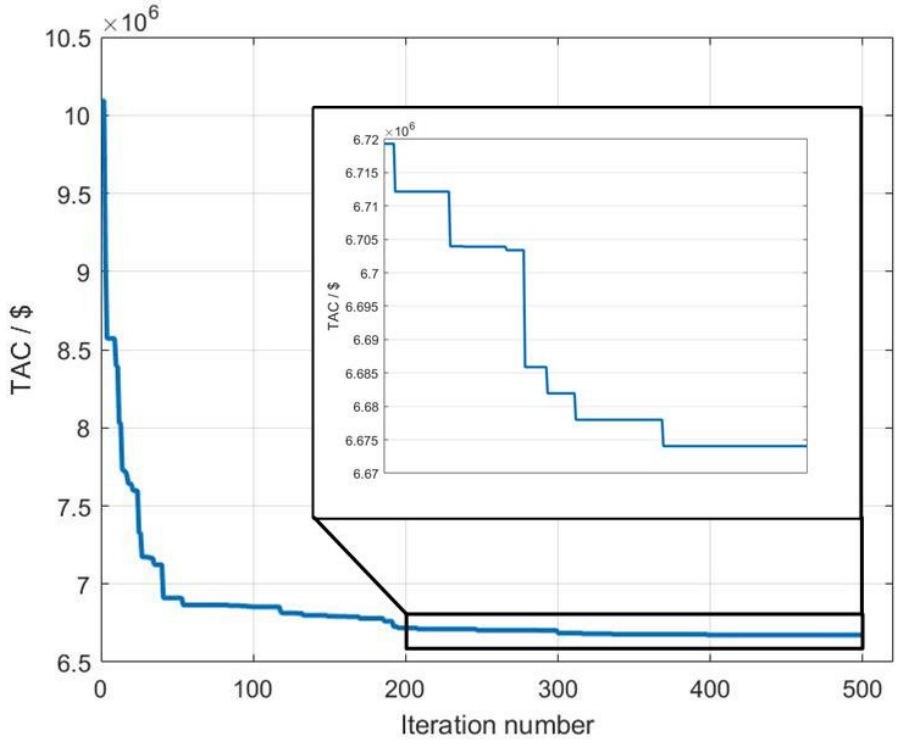


Figure S2 The iteration diagram of SPDDE for case 3

Table S8 The upper and lower bounds of the independent variable in Case 3

Item	Variable Range
Distillate molar flow rate of extraction column D (kmol/h)	[87.6482,88.079]*
Mole reflux ratio of extraction column R_1	[1,20]
Column absolute pressure of extraction column P_1 (atm)	[0.6,3]
Bottom molar flow rate of regeneration column B (kmol/h)	[2000,8000]
Mole reflux ratio of regeneration column R_2	[1,20]
Column absolute pressure of regeneration column P_2 (atm)	[0.2,1]
Supplementary flow rate of fresh solvent D_S (kmol/h)	[0,0.5]
Control temperature of cyclic extractant T_E (K)	[313.15,365]
The number of theoretical stages in rectifying section of extraction column N_1	[20,25]
The number of theoretical stages in extraction section of extraction column N_2	[55,65]
The number of theoretical stages in stripping section of extraction column N_3	[90,95]
The number of theoretical stages in rectifying section of regeneration column N_4	[8,12]
The number of theoretical stages in stripping section of regeneration column N_5	[5,10]