Efficient hybrid multiobjective optimization of pressure swing adsorption

Hao, Zhimian; Caspari, Adrian; Schweidtmann, Artur M.; Vaupel, Yannic; Lapkin, Alexei A.; Mhamdi, Adel

DOI
10.1016/j.cej.2021.130248

Publication date
2021

Document Version
Final published version

Published in
Chemical Engineering Journal

Chemical Engineering Journal 423 (2021) 130248

Contents lists available at ScienceDirect

Chemical Engineering Journal

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

Efficient hybrid multiobjective optimization of pressure swing adsorption

Zhimian Hao [a], Adrian Caspari [b], Artur M. Schweidtmann [b,c], Yannic Vaupel [b], Alexei A. Lapkin [a,d,*], Adel Mhamdi [b,*]

[a] Department of Chemical Engineering and Biotechnology, University of Cambridge, Cambridge CB3 0AS, UK
[b] Aachener Verfahrenstechnik - Process Systems Engineering, RWTH Aachen University, Aachen 52062, Germany
[c] Department of Chemical Engineering, Delft University of Technology, Van der Maasweg 9, Delft 2629 HZ, the Netherlands
[d] Cambridge Centre for Advanced Research and Education in Singapore Ltd, 1 Create Way, CREATE Tower #05-05, 138602, Singapore

ARTICLE INFO

Keywords:
Bayesian optimization
Gradient-based deterministic optimization
Pressure swing adsorption

ABSTRACT

Pressure swing adsorption (PSA) is an energy-efficient technology for gas separation, while the multiobjective optimization of PSA is a challenging task. To tackle this, we propose a hybrid optimization framework (TSEMO + DyOS), which integrates two steps. In the first step, a Bayesian stochastic multiobjective optimization algorithm (i.e., TSEMO) searches the entire decision space and identifies an approximated Pareto front within a small number of simulations. Within TSEMO, Gaussian process (GP) surrogate models are trained to approximate the original full process models. In the second step, a gradient-based deterministic algorithm (i.e., DyOS) is initialized at the approximated Pareto front to further refine the solutions until local optimality. Therein, the full process model is used in the optimization. The proposed hybrid framework is efficient, because it benefits from the coarse-to-fine function evaluations and stochastic-to-deterministic searching strategy. When the result is far away from the optima, TSEMO can efficiently approximate a trade-off curve as good as a commonly used evolutional algorithm, i.e., Nondominated Sorting Genetic Algorithm II (NSGA-II), while TSEMO only uses around 1/16th of CPU time of NSGA-II. This is because the GP-based surrogate model is utilized for function evaluations in the initial coarse search. When the result is near the optima, the searching efficiency of TSEMO dramatically decreases, while DyOS can accelerate the searching efficiency by over 10 times. This is because, in the proximity of optima, the exploitation capacity of DyOS is significantly higher than that of TSEMO.

1. Introduction

Pressure swing adsorption (PSA) is an energy-efficient gas separation technology [1–3] that has been widely used in the industry for drying [4], air separation [5,6], and hydrogen production [7,8]. Over the last two decades, academia has seen a growing interest in applying PSA for CO_2 capture [9,10]. PSA possesses significant advantages over the conventional amine-based CO_2 capture technology with regards to emissions to the environment and energy consumption [3,11]. Since no amine solvent is involved in the PSA system, no organic waste is disposed to the environment.

The optimal design and operation of PSA processes are challenging tasks due to the inherent cyclic and dynamic behavior of the system and highly nonlinear process models [12]. Since the column pressure varies over time, the PSA process can never reach a steady-state operating point. Instead, it eventually comes to a cyclic steady state (CSS), where the trajectories of state variables are the same for consecutive cycles. From an industrial operation perspective, PSA is required to operate at CSS as to achieve a constant process performance. However, it is difficult to analytically calculate CSS, which generally requires a numerical simulation [13–15]. Additionally, multiple (conflicting) objectives coexist, including product purity, recovery rate, energy consumption, and operating cost [11,16,17]. The process design and operation problems often involve nonconvex functions [18–20], where multiple local optimal solutions exist. Further, PSA may be operated in more complicated modes, *e.g.,* multiple columns integrated with recycles [3,11,12,17]. Overall, the above-mentioned factors contribute to the difficulty for the optimization of PSA processes.

In the previous literature, stochastic optimization algorithms have been used to optimize PSA processes [11,16,21]. Stochastic optimization algorithms consider the simulation as a black-box function. They vary the values of decision variables and run the PSA simulation until CSS. Following this procedure, the values of objectives and constraints are returned to the optimizer for evaluations. Haghpanah *et al.* used a

* Corresponding authors.

https://doi.org/10.1016/j.cej.2021.130248
Received 28 February 2021; Received in revised form 30 April 2021; Accepted 4 May 2021
Available online 10 May 2021

genetic algorithm (GA) to optimize the PSA operation, while the time-consuming feature of PSA simulation leads to the slow performance of the overall optimization [11]. Capra *et al.* [16] reported a multi-level coordinate search (MCS) algorithm, where the decision space is divided for parallel computing on multiple workers to speed up the overall optimization. Stochastic algorithms can search the decision space globally. However, the optimality cannot be guaranteed in finite time [22], and thus the solution found through stochastic optimization does not satisfy Karush-Kuhn-Tucker (KKT) optimality conditions [23].

Deterministic algorithms belong to another type of method that can be used for PSA optimization, where gradient information is used to guide the search direction (thus, it is often referred to 'gradient-based optimization'). There are two common approaches for the gradient-based optimization of dynamic systems, i.e., the simultaneous and the sequential approaches [23]. The simultaneous approach discretizes the state and decision variables. Herein, both temporal and spatial domains of partial differential equations (PDEs) are discretized, resulting in a large set of algebraic equations and eventually large-scale nonlinear programming (NLP) problems. Tsay *et al.* proposed a pseudo-transient optimization framework to identify the final cycle of PSA under CSS using a 'tear-recycle' method, in which the temporal domain is significantly reduced [24]. The sequential approach is well-suited to problems with a few decision variables and complex dynamic behavior. The integrator solves the differential equations and provides the gradient to the NLP solver. However, in the case of PSA, a significant amount of computational time is required to calculate the sensitivity information and its integration over many PSA cycles for the gradient. Additionally, the sensitivity integration may fail due to the highly nonlinear PSA model [13]. Jiang *et al.* focused on one PSA cycle [t_0, t_{end}] and applied the sequential approach to converge the initial conditions (t_0) to the endpoint (t_{end}) of state variables [13]. This concept can dramatically accelerate the simulation to reach CSS. However, the spatial-discretized PSA model contains over 1,000 state variables, and thus the convergency of them is still a large optimization problem.

Besides the extensive work on applying various optimization algorithms to PSA, researchers have exerted effort on developing surrogate models to represent the dynamic behavior of PSA. Surrogate models are cheap-to-evaluate and can approximate the relationship between inputs and outputs of physical models. Jiang *et al.* employed a Lagrange interpolation polynomial to approximate the profiles of state variables, as to simplify the convergence problem. Nevertheless, such approximation was reported to introduce inaccuracy for the further optimal design of PSA process [13]. Agarwal *et al.* demonstrated that proper orthogonal decomposition (POD) can be employed to replace the stiff PDEs of PSA. A POD can achieve a significant reduction of state variables and thus lead to low-order surrogate models [25].

With the recent increasing attention to machine learning, Artificial Neural Networks (ANNs) and Gaussian processes (GP) surrogate models have become prominent options for replacing computationally expensive models [26–28]. Subraveti *et al.* applied the ANN-based surrogate model to represent the original model, which was coupled with non-dominated sorting genetic algorithm II (NSGA-II) for multiobjective optimization. The CPU time was reported to be 10 times shorter compared to NSGA-II coupled with the original PSA model [17]. Leperi *et al.* employed individual ANN-based surrogate models to represent typical PSA stages. Then, these surrogate-based PSA stages can synthesize different types of cycles (three-stage, four-stage or five-stage cycle) [21]. Boukouvala *et al.* applied a grey-box method to capture both the analytical information of the physical models and noise information by a GP-based surrogate model [29]. With this method, PSA processes with different materials were optimized successfully within acceptable computational time [29]. However, surrogate models are often criticized for their inaccuracy and lack of generalization [30].

In summary, prior studies on PSA optimization are based on (1) stochastic algorithms using expensive full-order models, in which optimality cannot be guaranteed, (2) deterministic algorithms which require the expensive-to-obtain gradient information, or (3) surrogate formulations in which accuracy might be compromised. A hybrid method may integrate the complementary advantages of the individual methods. The concept of hybrid optimization methods – a synthesis of a global solver with a local solver – has been proposed initially by computer scientists to solve nonconvex problems many years ago [31–33]. Similarly, a concept of 'coarse-to-fine' search also proposes to transform the original problem into a coarse approximation for the initial search and then gradually approach the actual problem for refined search [34]. The efficiency of these concepts has been proven in the areas of computer vision [34], speech signal processing [35], and image processing [36]. Nevertheless, these concepts are not frequently used in the chemical industry.

Therefore, we propose a hybrid strategy: a stochastic algorithm for the initial search and then a gradient-based algorithm for the local refinement of the solution. This work achieves efficient multiobjective optimization of the PSA system by hybrid optimization framework.

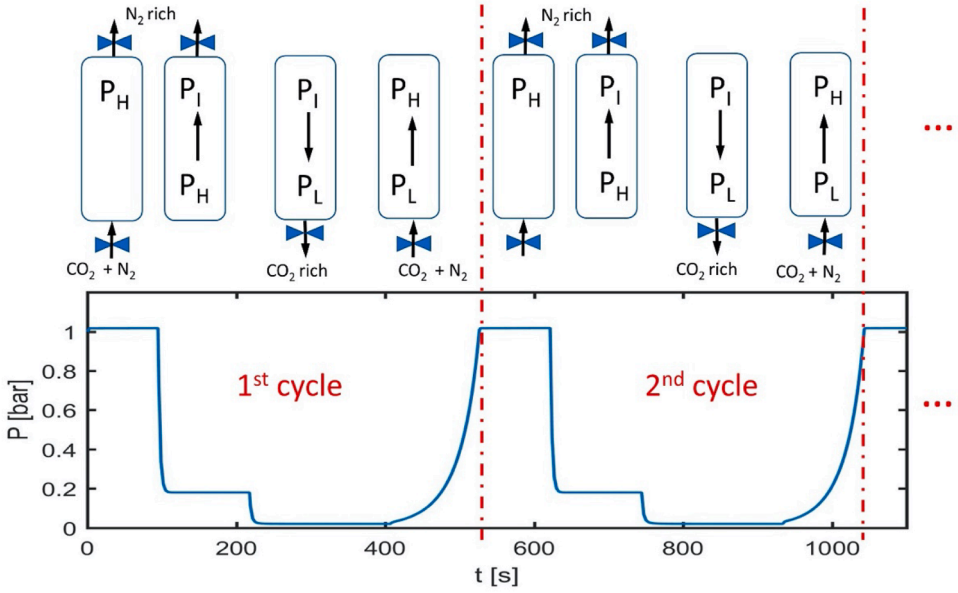

Fig. 1. Four-stage PSA for CO_2 capture.

The efficiency of the hybrid optimization framework benefits from:

- the stochastic-to-deterministic search strategy;
- the coarse-to-fine function evaluations: initially GP-based surrogate model for the rough evaluation, then the rigorous process model for the refined evaluation.

The remaining sections are structured as follows. Section 2 briefly describes the process model of PSA. Section 3 introduces the state-of-the-art algorithms used in the hybrid framework. Section 4 presents the optimization formulation of PSA using a hybrid optimization framework. Section 5 shows results, followed by the discussion on why the overall optimization efficiency of the hybrid framework is competitive in Section 6. The final section presents conclusions and outlook.

2. Model description of pressure swing adsorption

PSA is operated in a cyclic mode that alternates between adsorbing the desired gas species at a higher pressure and releasing them at a lower pressure (Fig. 1). Due to the variations in time and space, the PSA system is mathematically described by PDEs, which are based on the mass, energy and momentum balances listed in the Supplementary Information (SI, equation S1-S19). Notably, discontinuities are introduced by a sequence of frequent control actions of pressure levels, thus resulting in multiple discrete stages, *e.g.,* adsorption, blowdown, evacuation and feed pressurization, while each stage is operated continuously. Hence, the overall process belongs to a class of *combined discrete/continuous* systems, which require additional effort in the model formulation and numerical solution [37].

Table 1
Characteristics of TSEMO, DyOS and hybrid framework.

	Searching strategy	Function evaluations	Optimality
TSEMO (Bayesian optimization)	Stochastic (global search)	GP-based surrogate model	NO
DyOS	Gradient-based (deterministic)	Rigorous model	YES
Hybrid framework (TSEMO + DyOS)	Stochastic to deterministic	Surrogate to rigorous model (coarse-to-fine)	YES

The process model of PSA is based on the work of Haghpanah *et al.* [11] and implemented in Modelica using Dymola. The weighted essentially nonoscillatory (WENO) method, a finite volume method, is applied to discretize the PDEs into DAEs using 30 finite volumes. The *combined discrete/continuous* feature of PSA can first be described by a superstructure formulation of all PSA stages (SI, equation S.19), and then external controls (binary variables, refers to Table S2 in SI) are imposed to determine which stage to execute. As such, the *combined discrete/continuous* PSA is transformed into a set of continuous subsystems. Each subsystem is mathematically described by DAEs. The simulation of PSA requires the numerical integration of a series of initial value problems (IVP). The PSA cycle is repeatedly simulated and eventually reaches CSS. The simulation result is listed in the S3 section in the supplementary information (SI), because it is not the key finding in this work. Haghpanah's model has been validated experimentally [38,39] and our simulation result is in good agreement (SI, Table S2) with those reported by Haghpanah *et al.* [11].

3. State-of-the-Art of hybrid optimization framework

The hybrid optimization framework integrates TSEMO [40] with DyOS [41]. The characteristics of the methods are summarized in Table 1. TSEMO uses the input–output dataset of simulation results to train a GPs-based surrogate model, which is refined iteratively by sampling new input data points for more simulation results. Thompson sampling is the acquisition function for updating the dataset. In each iteration, the surrogate model is used as the evaluation function for multiobjective optimization [40]. With these characteristics, TSEMO belongs to Bayesian optimization [42]. NSGA-II is the optimizer within TSEMO, so the searching strategy of TSEMO is stochastic and the optimality cannot be guaranteed. DyOS contains a local sequential dynamic optimization solver, so the searching strategy belongs to gradient-based (deterministic) optimization and the optimality can be secured. The original dynamic process model is required to calculate the gradient information, and thus the function evaluations of DyOS are based on the rigorous process model.

The proposed hybrid optimization framework consists of two steps. In Step 1, TSEMO searches the decision space globally to generate an approximate trade-off curve, which contains the best points obtained by TSEMO. In Step 2, DyOS is initialized at one of the best points obtained in Step 1 and improves the solution until local optimality is reached.

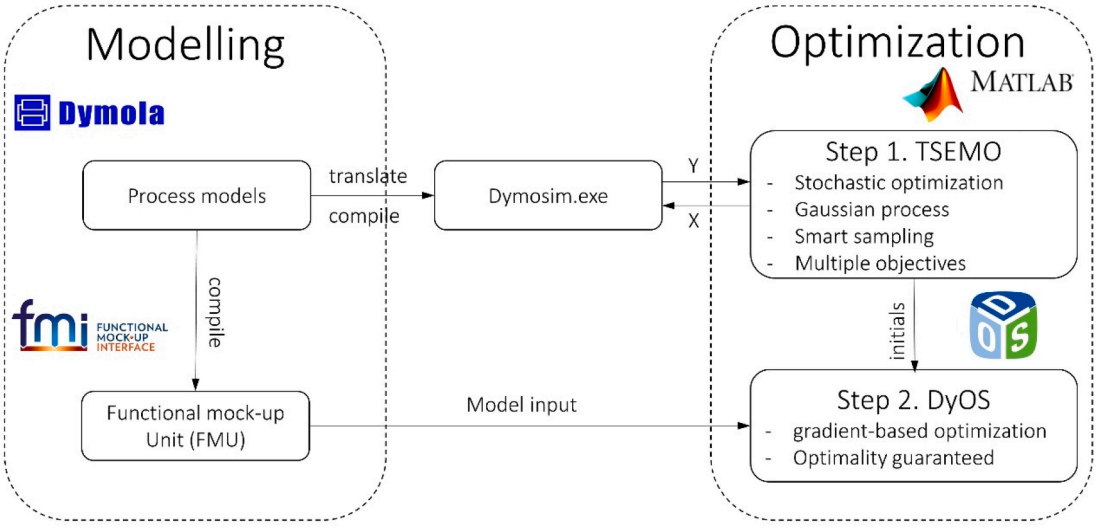

Fig. 2. Illustration of the integrated platform for modeling and optimization of PSA. Process models of PSA are programmed using Modelica language in Dymola. The Modelica model can be translated and compiled into an executable Dymosim.exe and called directly from Matlab. Alternatively, the Modelica model can be compiled as a functional mock-up unit (FMU) [43]. TSEMO runs Dymola through Dymosim.exe for simulation-based stochastic optimization, while DyOS takes an FMU as a model input for gradient-based optimization.

DyOS can only improve one point per time, so the second step needs to be repeated to 'one-by-one' improve all of the best points obtained in Step 1. Overall, the searching strategy is stochastic-to-deterministic, and the function evaluations are 'coarse-to-fine' type: initially the GP-based surrogate for rough evaluations, then the rigorous model for the refined evaluations. The overall optimization framework is implemented in MATLAB, as illustrated in Fig. 2. The model in Dymola can be compiled into an executable file (Dymosim.exe) and Functional mock-up Unit (FMU), which can be seamlessly integrated into the MATLAB environment. In Step 1, the PSA model is coupled to TSEMO as an executable. In Step 2, the model is coupled to DyOS through the FMU, and then MATLAB calls DyOS through a mex interface.

As a reference, we also employ the NSGA-II, a well-established evolutionary algorithm, to optimize the original process model of PSA.

4. Optimization formulation of PSA using the hybrid framework

One of the challenges in PSA optimization is owed to multiple (conflicting) criteria for the final product. In this work, we employ PSA for CO_2 capture, and two optimization objectives are considered: (i) the recovery rate and (ii) the purity of the product gas CO_2 are maximized.

$$\text{Recovery} = \frac{CO_2 \text{ in product within a CSS cycle}}{CO_2 \text{ fed into column within a CSS cycle}} \times 100\% \quad (1)$$

$$\text{Purity} = \frac{CO_2 \text{ in product within a CSS cycle}}{\text{total gas in product within a CSS cycle}} \times 100\% \quad (2)$$

The details of the hybrid approach (1st TSEMO + 2nd DyOS) are formulated in this section.

4.1. First step: optimization formulation using TSEMO

TSEMO can deal with multiobjective optimization problems directly, and two objectives can be inserted in the solver without any further reformulation. The formulation is constrained by the process equations (cf. SI, S.1-S.19). The evaluation and optimization of PSA are only meaningful after the process reaches CSS. As an evaluation method for CSS, a small tolerance value, δ, is used to check the difference between state variables, x, over one cycle. When $\left|x(t) - x(t + t_{cycle})\right| \leq \delta$, PSA is deemed to be under CSS. Overall, in the TSEMO optimization framework, the PSA optimization problem is formulated as follows, Eqs. (3)-(5):

$$\max_{\theta} (\text{Recovery}, \text{Purity}) \quad (3)$$

$$\text{s.t. Dynamic process model} (\text{SI}, \text{S.1} - \text{S.19}) \quad (4)$$

$$\text{CSS} = \left|x(t) - x\left(t + t_{cycle}\right)\right| \leq \delta \quad (5)$$

where θ is a vector of six decision variables of four-stage PSA system including the duration of the first stage - adsorption (t_{ads}), the duration of the second stage - blowdown (t_{bd}), the duration of the third stage - evacuation (t_{evac}), two pressure setpoints - intermediate pressure (P_I), low pressure (P_L), respectively as well as feed velocity (v_{feed}). The lower and upper bounds of the decision variables are given in Table 2. In this work, the highest pressure is fixed at 1 bar. The duration of the pressurization stage (the fourth stage) is reported to have a negligible effect on the operation of PSA; therefore, it is fixed to 20 s [11].

Table 2
The ranges of the decision variables in the PSA optimization *via* TSEMO.

θ	t_{ads} [s]	t_{bd} [s]	t_{evac} [s]	P_I [bar]	P_L [bar]	v_{feed} [m/s]
range	20–100	30–200	30–200	0.07–0.5	0.005–0.05	0.1–2

Table 3
The ranges of the decision variables in the PSA optimization *via* DyOS.

θ_i	P_I [bar]	P_L [bar]	v_{feed} [m/s]
Range	0.07–0.5	0.005–0.05	0.1–2

4.2. Second step: optimization formulation of PSA using DyOS

DyOS is designed to solve single-objective optimization problems. Herein, we reformulate our multiobjective optimization problem into a series of single-objective optimization problems *via* the epsilon-constrained method [44]. In other words, the recovery remains to be the objective, while the purity is reformulated as an inequality constraint. Following the results from the first step, the constraint and the initial values of decision variables are based on the results obtained from TSEMO. In case that the constraint is too tight, a relaxation coefficient ($\eta = 0.99$) is given for the purity constraint (Eq. (8)). When optimizing PSA using DyOS, the system is assumed to reach CSS at the same number of cycles as the optimization using TSEMO (Eq. (9)). The set-up of DyOS for PSA optimization is illustrated in Figure S5 (SI). The formulation of PSA optimization in DyOS is as follows, Eqs. (6)-(9):

$$\max_{\theta_i} \text{Recovery} \quad (6)$$

$$\text{s.t. Dynamic process model} (\text{SI}, \text{S.1} - \text{S.19}) \quad (7)$$

$$\text{purity} \geq \text{purity}_{\text{TSEMO}} \cdot \eta \quad (8)$$

$$N = N_{TSEMO} \quad (9)$$

The PSA optimization *via* DyOS is conducted with respect to three decision variables: intermediate pressure, low pressure and inlet flowrate, as shown in Table 3. In the initial trials with DyOS we included the duration variables, which caused the method not to converge, likely because sensitivity integration over time is highly related to duration variables. Since the reason for unsuccessful termination is unclear at this time, we did not include the duration variables into the optimization.

5. Results

5.1. First step: optimization using TSEMO

To initialize TSEMO, 30 random sets of inputs were sampled using a Latin Hypercube Sampling (LHS) method, and then the simulation inputs and outputs (i.e., recovery and purity) were used to train the initial GPs. Then, random samples were drawn from the GPs and multiobjective optimization was performed. Following this, new inputs for simulations were recommended by the algorithm to improve the objectives. Then, the new data points were added to the whole dataset for GP surrogate training in the next iteration. In this case study, we discuss the optimization results after 50, 100, 200, 300, 400, 500, and 600 PSA simulations, which were recommended by TSEMO. Fig. 3(a) shows the obtained Pareto front, which represents the trade-off between recovery and purity through different numbers of simulations. The hypervolume can be used as an indicator to quantify the performance of multiobjective optimization [45,46]. Fig. 3(b) shows that the hypervolume improves with the increase in the number of simulations. A significant improvement for the estimated Pareto front between 50 and 100 simulations is observed while only moderate change is observed when further increasing the number of simulations. The growth in the hypervolume is negligible once the number of simulations is above 200 (Fig. 3b). This result might be explained in two ways: one explanation is that the estimated Pareto front is almost close to the actual Pareto front and leaves little space for further improvement; an alternative explanation is

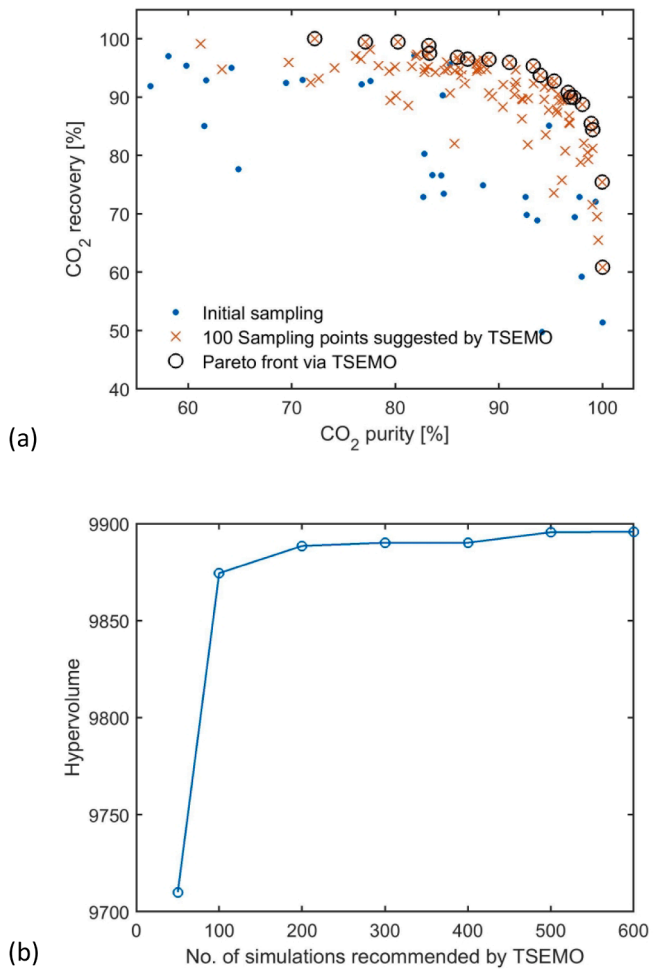

Fig. 3. Multiobjective optimization of PSA *via* TSEMO. (a) optimization results through 100 simulations recommended by TSEMO: to initialize TSEMO, LHS generated 30 simulations, shown as the blue points; the algorithm recommended additional 100 simulations, shown as the red crosses. The estimated Pareto front was evolved, shown as the black circles. (b) hypervolume quantification (reference point is [0, 0]) varying from 50 to 600 simulations recommended by TSEMO. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

that the searching efficiency of TSEMO considerably drops when the identified solutions are approaching optimality. This is a known issue of any stochastic search algorithm: the convergence is only guaranteed in the limit of an infinite number of function evaluations.

5.2. Second step: optimization using DyOS

One issue with the stochastic global search is the lack of local refinement of the identified solutions. In particular, TSEMO does not use gradient information to improve the approximate solutions further. Hence, it is desired to perform further gradient-based optimization that is initialized from the approximate solution points obtained in the first step. Following 600 simulations *via* TSEMO, we selected 22 non-dominated points with purity over 80% and recovery over 75%, which are the starting points in the second step. For every individual point, DyOS is called to perform gradient-based optimization using the full model. As shown in Fig. 4, DyOS slightly improves the estimated Pareto front until local optimality is satisfied. When referring to the hypervolume in Table 4, the improvement is not significant, which indicates that the estimated Pareto front based on the limited number of TSEMO simulations is very close to the local refined solution by gradient-based optimization.

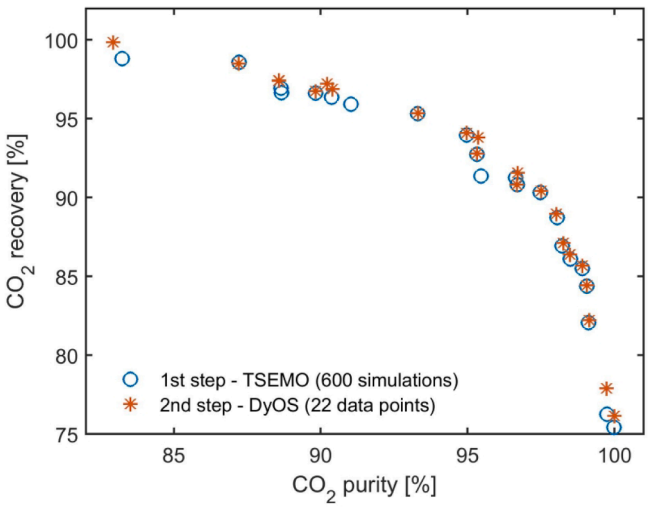

Fig. 4. The result of the hybrid approach for the multiobjective optimization of PSA.

Table 4
Optimization performance *via* TSEMO and DyOS (reference point of hypervolume quantification is [0, 0]).

	First step -TSEMO (600 simulations)	Second step - DyOS
CPU time [h]	29.5	81.7
Hypervolume [-]	9,896	9,932

Table 4 presents the optimization performance. The hypervolume quantification indicates that DyOS does further improve the results from TSEMO. Nevertheless, the CPU time of DyOS is almost three times that of TSEMO. This is because TSEMO uses cheap-to-evaluate surrogate models and parallel computing is possible for surrogate models. By contrast, DyOS relies on gradients calculated from the sensitivity integration over all PSA cycles, and thus a large percentage of time is consumed to obtain the gradient information. Notably, the full-order physical model is evaluated to ensure the result's accuracy, which further increases the CPU cost in the second step.

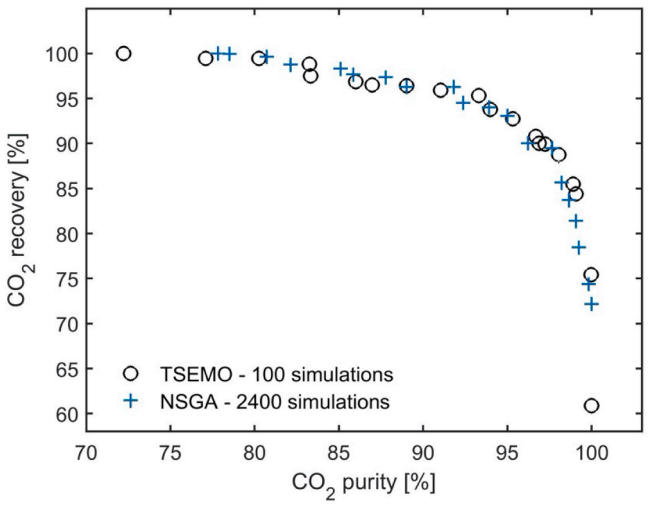

Fig. 5. Comparison between Pareto set of solutions obtained by TSEMO – 100 simulations and NSGA-II – 2,400 simulations.

Table 5
Optimization performance between NSGA-II and TSEMO (reference point of hypervolume quantification is [0, 0]).

	NSGA-II 2,400 simulations	TSEMO 100 simulations
CPU time [h]	63.2	3.9
Hypervolume [-]	9,877	9,875

6. Discussion

To demonstrate the efficiency of this hybrid framework, we firstly compare the performance of TSEMO with that of NSGA-II. As shown in Fig. 5, the estimated Pareto front from TSEMO is comparable to that of NSGA-II, while NSGA-II requires a significantly larger number of simulations than TSEMO. As shown in Table 5, TSEMO with 100 simulations has a hypervolume value almost the same as the NSGA-II with 2,400 simulations, while TSEMO only uses around 1/16th of the CPU time of NSGA-II. This is reasonable because TSEMO trains the GP-surrogate for the function evaluations during optimization, so it is not CPU-intensive as the rigorous model. NSGA-II is actually the optimizer within the TSEMO framework, so TSEMO has a similar exploration capacity as NSGA-II. TSEMO also employs Thompson sampling (acquisition function) to choose new sampling points, thus improving the exploitation capability. Therefore, the efficiency of TSEMO is higher than NSGA-II.

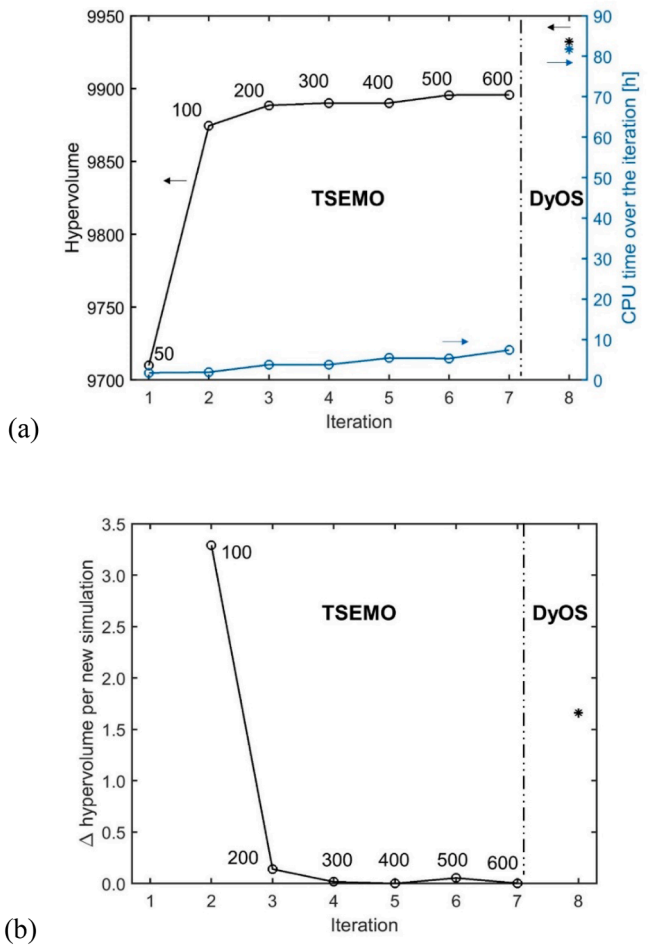

Fig. 6. (a) Hypervolume and CPU time via TSEMO and DyOS (The reference point of hypervolume quantification is [0, 0]). (b) the average hypervolume improvement when a new simulation is added. Iterations 1 – 7 refer to the influence of TSEMO, which recommends 50, 100, 200, 300, 400, 500, and 600 simulations, respectively. Iteration 8 refers to the influence of DyOS based on 22 data points.

Table 6
Searching efficiency *via* TSEMO and DyOS (reference point of hypervolume quantification is [0, 0]).

	First step -TSEMO (7th iteration)	Second step - DyOS
CPU time [h]	7.38	81.7
Hypervolume improvement [-]	0.3	36.5
Searching efficiency [1/h]	0.04	0.45
No. of updated data points [-]	100	22
Hypervolume improvement per point [-]	0.003	1.66

From Table 4, we noticed that the optimization result from TSEMO is close to that of DyOS, but DyOS costs significantly more CPU time. However, it is important to notice that the deterministic local search also offers distinct advantages for the considered case study. Firstly, DyOS verifies that the optimization result of TSEMO is 'good enough'. Without the verification, there are no criteria to check the optimality only by TSEMO. Secondly, DyOS indeed improves the optimization result. A slight improvement of operating condition may only introduce little difference in one hour for a laboratory set-up. However, such improvement can be significant for an annually operated industrial PSA plant. Last but not least, the searching efficiency of DyOS is higher than TSEMO when the optimization result is near optima. We introduce a value to quantify the searching efficiency:

$$searching\ efficiency = \frac{hypervolume\ improvement}{CPU\ time} \tag{10}$$

As shown in Fig. 6a, the growth of hypervolume slows down with the increase of iteration of TSEMO, while the CPU time starts to increase gradually. Thus, the search efficiency of TSEMO dramatically decreases after 3rd iteration. DyOS is initialized based on the result of the 7th iteration of TSEMO. The searching efficiency of DyOS is over 11 times that of TSEMO on its 7th iteration (Table 6). This means that TSEMO requires much more than 11 times CPU time to achieve the same trade-off curve calculated from DyOS, given the searching efficiency of TSEMO keeps going down.

TSEMO belongs to a stochastic search algorithm. Theoretically, TSEMO can only converge to optimality in an infinite number of function evaluations. In other words, the searching efficiency of TSEMO declines inevitably and approaches 0 eventually. That is an inherent characteristic of any stochastic method – focusing on space-filling, rather than the improvement of individual points as gradient-based methods. Both TSEMO and DyOS tend to find better results than the last iteration, but the improvement on individual points is quite different. As shown in Fig. 6b, the average hypervolume improvement on an individual point drops significantly with the increase of TSEMO iteration, while DyOS can still take advantage of the gradient to further optimize the individual point (operating conditions for new simulation). As shown in Table 6, the difference can be 553 times when comparing between DyOS and the last iteration of TSEMO, regarding the hypervolume improvement of an individual point. In other words, in the proximity of an optimal solution, DyOS possesses a significantly higher exploitation capacity than TSEMO.

7. Conclusions and outlook

When solving the multiobjective optimization problem of PSA deterministically, the main challenge is the high computational cost. In this work, a hybrid (TSEMO + DyOS) optimization framework is developed to secure a high searching efficiency and accuracy for a four-stage PSA system with an application in CO_2 capture.

In the hybrid optimization framework, the first step employs our open-source Bayesian optimization algorithm, TSEMO, to search the full decision space efficiently. This step identifies an approximate Pareto front of two objectives, CO_2 purity and recovery. In the second step,

DyOS starts from the most promising objective points obtained in the first step and further improves the optimization result of PSA until optimality. The small improvement in the 2nd step indicates that TSEMO can achieve nearly optimal operation conditions of PSA within the limited number of simulations.

The hybrid optimization framework possesses an excellent optimization efficiency. Such efficiency benefits from the coarse-to-fine function evaluations and stochastic-to-deterministic searching strategy. TSEMO employs GP-surrogates for function evaluations in the initial coarse search. Hence, the efficiency of TSEMO is higher than NSGA-II. However, the searching efficiency of TSEMO dramatically drops on the nearly-optimal condition, where the hybrid framework can use DyOS to further improve the searching efficiency by over 10 times. This is because TSEMO belongs to stochastic methods, which are weaker in exploitation than deterministic methods, when the optimal solution is nearly optimal. Therefore, the overall searching efficiency on PSA optimization can be ranked as follows, hybrid (TSEMO + DyOS) framework > TSEMO > NSGA-II.

Ideally, the hybrid framework can be implemented iteratively as follows, (TSEMO → DyOS) → (TSEMO → DyOS) → (TSEMO → DyOS) … An iterative way can help balance the exploration and exploitation better, thus leading to fast convergence to the optimal solution. In the case study of PSA, the optimization result from TSEMO was thought to be 'good enough', which can be referred to the result of NSGA-II (2400 simulations / 63 h in total) and DyOS. Also, the second step on DyOS consumed significantly more time. As a result, the iterative way for the hybrid framework was set aside. In the future, two factors might make the iterative way more appealing and practical: 1) fast evaluation of PSA process model: reformulate the PSA model to make the system efficiently converge to cyclic steady state; 2) parallel computing in DyOS: initialize the exploitation for all individual points simultaneously.

This hybrid multiobjective optimization framework can be used to explore other competing criteria, such as energy consumption and productivity of PSA. Further, this approach can be extended to optimization of any other complex expensive-to-evaluate dynamic processes. TSEMO seems to already deliver a 'good-enough' trade-off curve among multiple criteria in a relatively low time cost, while the hybrid framework can be used to accelerate the trade-off curve to converge to the real 'good-enough' solution. Pursuing the optimality can be especially meaningful to high-value processes because a slight improvement of the operating condition can make a significant impact on an annually operating industrial plant.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

Authors declare no competing financial interest. ZH acknowledges the financial support from Cambridge Trust and Chinese Scholarship Council. ZH's secondment to Aachen was funded by Sustainable Reaction Engineering research group of Prof. Lapkin. We thank Mr. Tobias Ploch (RWTH Aachen) for helping to debug the PSA model in Dymola. AC gratefully acknowledges the financial support of the Kopernikus project SynErgie by the Federal Ministry of Education and Research (BMBF) and the project supervision by the project management organization Projektträger Jülich (PtJ). AAL acknowledges funding from National Research Foundation (NRF), Prime Minister's Office, Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) program as a part of the Cambridge Centre for Advanced Research and Education in Singapore Ltd (CARES).

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.cej.2021.130248.

References

[1] S. Sircar, Pressure swing adsorption, Ind. Eng. Chem. Res. 41 (6) (2002) 1389–1392.
[2] R.V. Jasra, N.V. Choudary, S.G.T. Bhat, Separation of gases by pressure swing adsorption, Sep. Sci. Technol. 26 (7) (1991) 885–930.
[3] R. Haghpanah, R. Nilam, A. Rajendran, S. Farooq, I.A. Karimi, Cycle synthesis and optimization of a VSA process for postcombustion CO_2 capture, AICHE J. 59 (12) (2013) 4735–4748.
[4] J.W. Carter, M.L. Wyszynski, The pressure swing adsorption drying of compressed air, Chem Eng Sci 38 (7) (1983) 1093–1099.
[5] D.M. Ruthven, S. Farooq, Air separation by pressure swing adsorption, Gas Sep. Purif. 4 (3) (1990) 141–148.
[6] J.-G. Jee, J.-S. Lee, C.-H. Lee, Air Separation by a small-scale two-bed medical O_2 pressure swing adsorption, Ind. Eng. Chem. Res. 40 (16) (2001) 3647–3658.
[7] A. Malek, S. Farooq, Hydrogen purification from refinery fuel gas by pressure swing adsorption, AICHE J. 44 (9) (1998) 1985–1992.
[8] S. Sircar, T.C. Golden, Purification of hydrogen by pressure swing adsorption, Sep. Sci. Technol. 35 (5) (2000) 667–687.
[9] A.L. Chaffee, G.P. Knowles, Z. Liang, J. Zhang, P. Xiao, P.A. Webley, CO_2 capture by adsorption: Materials and process development, Int. J. Greenh Gas Con. 1 (1) (2007) 11–18.
[10] L.Y. Liu, H. Gong, Z. Wang, G. Li, T. Du, Application of Pressure Swing Adsorption Technology to Capture CO_2 in Highly Humid Flue Gas, Prog. Chem. 30 (2018) 872–878.
[11] R. Haghpanah, A. Majumder, R. Nilam, A. Rajendran, S. Farooq, I.A. Karimi, M. Amanullah, Multiobjective optimization of a four-step adsorption process for postcombustion CO_2 capture via finite volume simulation, Ind. Eng. Chem. Res. 52 (11) (2013) 4249–4265.
[12] Y. Tian, S.E. Demirel, M.M.F. Hasan, E.N. Pistikopoulos, An overview of process systems engineering approaches for process intensification: State of the art, Chem. Eng. Processing - Process Intensif. 133 (2018) 160–210.
[13] L. Jiang, L.T. Biegler, V.G. Fox, Simulation and optimization of pressure-swing adsorption systems for air separation, AICHE J. 49 (5) (2003) 1140–1157.
[14] Y. Ding, M.D. LeVan, Periodic states of adsorption cycles III Convergence acceleration for direct determination, Chem. Eng. Sci. 56 (17) (2001) 5217–5230.
[15] O.J. Smith, A.W. Westerberg, Acceleration of cyclic steady state convergence for pressure swing adsorption models, Ind. Eng. Chem. Res. 31 (6) (1992) 1569–1573.
[16] F. Capra, M. Gazzani, L. Joss, M. Mazzotti, E. Martelli, MO-MCS, a derivative-free algorithm for the multiobjective optimization of adsorption processes, Ind. Eng. Chem. Res. 57 (30) (2018) 9977–9993.
[17] S.G. Subraveti, Z. Li, V. Prasad, A. Rajendran, Machine learning-based multiobjective optimization of pressure swing adsorption, Ind. Eng. Chem. Res. 58 (44) (2019) 20412–20422.
[18] W.R. Esposito, C.A. Floudas, Deterministic global optimization in nonlinear optimal control problems, J. Global Optim. 17 (2000) 97–126.
[19] S. Lee, I.E. Grossmann, A global optimization algorithm for nonconvex generalized disjunctive programming and applications to process systems, Comput. Chem. Eng. 25 (11-12) (2001) 1675–1697.
[20] H.S. Ryoo, N.V. Sahinidis, Global optimization of nonconvex NLPs and MINLPs with applications in process design, Comput. Chem. Eng. 19 (5) (1995) 551–566.
[21] K.T. Leperi, D. Yancy-Caballero, R.Q. Snurr, F. You, 110th anniversary: Surrogate models based on artificial neural networks to simulate and optimize pressure swing adsorption cycles for CO_2 capture, Ind. Eng. Chem. Res. 58 (2019) 18241–18252.
[22] J. Nocedal, S. Wright, Numerical optimization, Springer Science & Business Media, 2006.
[23] L.T. Biegler, Nonlinear programming: concepts, algorithms, and applications to chemical processes, SIAM2010.
[24] C. Tsay, R.C. Pattison, M. Baldea, A pseudo-transient optimization framework for periodic processes: Pressure swing adsorption and simulated moving bed chromatography, AICHE J. 64 (2018) 2982–2996.
[25] A. Agarwal, L.T. Biegler, S.E. Zitney, Simulation and optimization of pressure swing adsorption systems using reduced-order modeling, Ind. Eng. Chem. Res. 48 (5) (2009) 2327–2343.
[26] D. Buche, N.N. Schraudolph, P. Koumoutsakos, Accelerating evolutionary algorithms with Gaussian process fitness function models, IEEE Trans, Syst. Man Cybern. Part C-Appl. Rev. 35 (2) (2005) 183–194.
[27] A.M. Schweidtmann, A. Mitsos, Deterministic global optimization with artificial neural networks embedded, J. Optimiz. Theory Appl. 180 (3) (2019) 925–948.
[28] D. Duvenaud, Automatic model construction with Gaussian processes, University of Cambridge, 2014.
[29] F. Boukouvala, M.M.F. Hasan, C.A. Floudas, Global optimization of general constrained grey-box models: new method and its application to constrained PDEs for pressure swing adsorption, J. Global Optim. 67 (2017) 3–42.
[30] Z.T. Wilson, N.V. Sahinidis, The ALAMO approach to machine learning, Comput. Chem. Eng. 106 (2017) 785–795.
[31] C.K. Chak, G. Feng, Accelerated genetic algorithms: combined with local search techniques for fast and accurate global search, Proceedings of the 1995 IEEE

International Conference on Evolutionary Computation. Part 1 (of 2), 1995, pp. 378-383.
[32] G. Folino, C. Pizzuti, G. Spezzano, Parallel hybrid method for SAT that couples genetic algorithms and local search, IEEE Trans. Evol. Comput. 5 (4) (2001) 323–334.
[33] C. Junying, Q. Zheng, L. Yu, L. Jiang, Particle swarm optimization with local search, Int. Conf. Neural Netw. Brain 2005 (2005) 481–484.
[34] H. Mobahi, J.W. Fisher, Coarse-to-Fine Minimization of Some Common Nonconvexities, in: E. Bae, T.F. Chan, M. Lysaker (Eds.) Energy Minimization Methods in Computer Vision and Pattern Recognition, Emmcvpr 2015, pp. 71-84.
[35] J. Yao, A. Al-Dahle, Coarse-to-fine Optimization for Speech Enhancement, arXiv preprint arXiv: 1908.08044 (2019).
[36] C. Raphael, Coarse-to-fine dynamic programming, IEEE Trans. Pattern Anal. Mach. Intell. 23 (12) (2001) 1379–1390.
[37] P.I. Barton, C.C. Pantelides, Modeling of combined discrete/continuous processes, AICHE J. 40 (6) (1994) 966–979.
[38] L. Estupiñan Perez, P. Sarkar, A. Rajendran, Experimental validation of multi-objective optimization techniques for design of vacuum swing adsorption processes, Sep. Purif. Technol. 224 (2019) 553–563.
[39] S. Krishnamurthy, V.R. Rao, S. Guntuka, P. Sharratt, R. Haghpanah, A. Rajendran, M. Amanullah, I.A. Karimi, S. Farooq, CO_2 capture from dry flue gas by vacuum swing adsorption: A pilot plant study, AICHE J. 60 (2014) 1830–1842.
[40] E. Bradford, A.M. Schweidtmann, A. Lapkin, Efficient multiobjective optimization employing Gaussian processes, spectral sampling and a genetic algorithm, J Global Optim 71 (2) (2018) 407–438.
[41] A. Caspari, A.M. Bremen, J.M.M. Faust, F. Jung, C.D. Kappatou, S. Sass, Y. Vaupel, K. Hannemann-Tamas, A. Mhamdi, A. Mitsos, DyOS - A Framework for Optimization of Large-Scale Differential Algebraic Equation Systems, in: A.A. Kiss, E. Zondervan, R. Lakerveld, L. Ozkan (Eds.) 29th European Symposium on Computer Aided Process Engineering, Pt A2019, pp. 619-624.
[42] B. Shahriari, K. Swersky, Z. Wang, R.P. Adams, N.d. Freitas, Taking the Human Out of the Loop: A Review of Bayesian Optimization, Proceedings of the IEEE 104 (2016) 148-175.
[43] P. Fritzson, P. Bunus, S. Ieee Computer, S. Ieee Computer, Modelica - A general object-oriented language for continuous and discrete-event system modeling and simulation, 35th Annual Simulation Symposium, Proceedings, 2002, pp. 365-380.
[44] R. Wu, G. Mavromatidis, K. Orehounig, J. Carmeliet, Multiobjective optimisation of energy systems and building envelope retrofit in a residential community, Appl. Energ. 190 (2017) 634–649.
[45] A. Auger, J. Bader, D. Brockhoff, E. Zitzler, Hypervolume-based multiobjective optimization: Theoretical foundations and practical implications, Theoretic. Comput. Sci. 425 (2012) 75–103.
[46] E. Zitzler, D. Brockhoff, L. Thiele, The Hypervolume Indicator Revisited: On the Design of Pareto-compliant Indicators Via Weighted Integration, in: S. Obayashi, K. Deb, C. Poloni, T. Hiroyasu, T. Murata (Eds.) International Conference on Evolutionary Multi-Criterion Optimization, Springer, Berlin, Heidelberg, 2007, pp. 862-876.