

MO-MCS, a Derivative-Free Algorithm for the Multiobjective Optimization of Adsorption Processes

Federico Capra,[†] Matteo Gazzani,[‡] Lisa Joss,[§] Marco Mazzotti,^{||} and Emanuele Martelli^{*,†}

[†]Politecnico di Milano, Department of Energy, Via Lambruschini 4, 20156 Milano, Italy

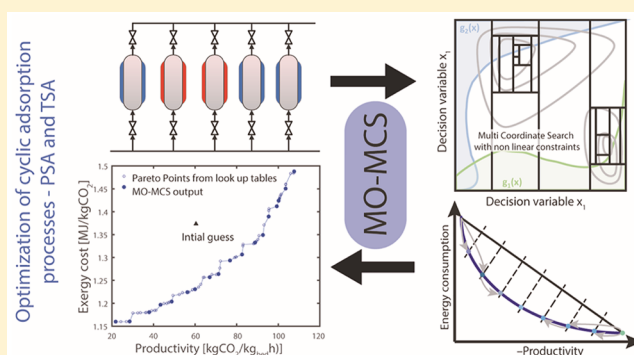
[‡]Utrecht University, Copernicus Institute of Sustainable Development, Heidelberglaan 3584CS Utrecht, The Netherlands

[§]Department of Chemical Engineering, Imperial College London, SW7 2AZ, London, United Kingdom

^{||}ETH Zurich, Institute of Process Engineering, Sonneggstrasse 3, 8092, Zurich, Switzerland

Supporting Information

ABSTRACT: Cyclic adsorption processes for gas separation, such as pressure and temperature swing adsorption (PSA and TSA), are nonstationary multicolumn processes. Their design involves many degrees of freedom, which offers a very high flexibility while calling for a systematic and rigorous optimization approach. As an additional challenge, optimization of these separation processes involves multiple objectives, e.g., minimal energy demand and maximal productivity, which have to be pursued while fulfilling given process specifications, e.g., purity and recovery of the target components. This work extends the multilevel coordinate search (MCS), a well-known model-based derivative-free algorithm, to constrained multi-objective problems. The algorithm, called MO-MCS, combines a built-in parallel computing strategy with the use of look-up tables with the goal of minimizing the computational time needed to determine the Pareto curve. The comparison with state-of-the-art optimizers indicates that MO-MCS shows better performance in terms of optimality, well spacing, and extension of the Pareto curve. Afterward, two industrially relevant case studies (TSA for CO₂ separation and PSA for H₂ and CO₂ coproduction) are tackled to demonstrate the effectiveness of the algorithm as a tool to guide the design of adsorption processes.



1. INTRODUCTION

Adsorption-based separation technologies are often regarded as key players for advanced gas separation processes: solid sorbents are nonvolatile, nontoxic, and offer the opportunity for a reduced energy penalty as compared to liquid sorbents, thanks to the lower sorption enthalpy.¹ Cyclic adsorption processes have particularly attracted interest because of their capability of achieving high separation targets, their flexibility to treat different compositions as feed, and the simplicity of the equipment and sorbents. These processes are divided depending on the nature of the regeneration step: pressure swing adsorption (PSA) cycles exploit a pressure switch, while temperature swing adsorption (TSA) works with a temperature switch. Both PSA and TSA share the common principle that the separation is achieved via using a solid medium undergoing alternating loading and regeneration phases. Accordingly, they are also hindered by very similar criticalities, especially (i) handling of impurities; (ii) development of new, cost-effective, and robust materials; (iii) design of suitable cycles; (iv) design of efficient bed geometries, including heat integration possibilities; and (v) enabling rigorous process optimization. The interplay among these challenges is especially tight when considering carbon capture and storage (CCS),^{2,3} a technology that is key for meeting the reduction of

CO₂ emissions as required by the 1.5/2 °C scenario.^{4,5} The present contribution aims at providing a new tool for the numerical optimization of cyclic adsorption processes, therefore tackling the last point of the list above.

The optimization of cyclic adsorption processes is inherently complex because of the dynamic and cyclic behavior of such processes, and the tight relation between reactor design, cycle time steps, and scheduling. Because of the dynamic operation of the adsorption bed, and the steep temperature and concentration profiles, the process simulation involves partial derivative equations (PDEs). Moreover, the complexity of the resolution is increased by the cyclic steady state condition (CSS)—i.e., cycle N+1 matches cycle N within the assigned tolerances.

Solution methods can be classified into two main categories: finite difference/elements methods and shooting methods. In the first, the periodic CSS condition and the design constraints are

Special Issue: 2017 European Symposium on Computer-Aided Process Engineering

Received: January 16, 2018

Revised: May 17, 2018

Accepted: May 17, 2018

Published: May 17, 2018

solved together with the time-space discretized PDEs by a nonlinear equation solver (Nilchan and Pantelides⁶). This leads to a very large set of algebraic equations. In the second (i.e., shooting methods), the PDEs are converted into a set of differential algebraic equations (DAEs) and integrated over time until the adsorption cycle reaches CSS. The basic approach to handle the CSS condition is found by the successive substitution method, which consists of using the bed state at the end of the cycle as a starting state for the subsequent cycle. While such an approach mimics the behavior of the real system, providing insights in the CSS convergence, it also features a linear convergence rate with considerable simulation times. More sophisticated approaches use sensitivity information from the solution of the DAEs to determine a new estimate of initial bed conditions using Newton-based algorithms⁷ which can achieve quadratic or superlinear convergence rates. On the other hand, these algorithms require extra computation time to calculate the Jacobian and may fail to converge for strongly nonlinear systems and complex bed models, because the Jacobian becomes poorly approximated.⁸

As far as the optimization of adsorption cycles is concerned, three main approaches have been proposed: (i) black-box methods,^{9–12} (ii) simultaneous tailored methods,¹³ and (iii) equation oriented methods.¹⁴ Black-box approaches couple a detailed model for the simulation of the adsorption column and an external optimization algorithm, where few independent design and operation variables are optimized. The optimization problem may be extremely challenging because (i) the numerical solvers employed to solve the PDEs of the model may make objective function and constraints noisy and nonsmooth and (ii) every cycle simulation needs to reach the CSS condition, making the computational time considerable (about 1–10 min to reach CSS conditions). The black-box approach, integrated with metaheuristic global optimization algorithms, has been adopted by different authors belonging to the adsorption research community.^{15–20} Notably, Boukouvala and Floudas^{9,21,22} have recently devised a methodology for the single-objective optimization of PSA processes where the problem is formulated as a constrained gray-box problem. The methodology couples in an original approach the use of surrogate functions,²³ deterministic global optimization algorithms (e.g., ref 24 or 25 depending on the type of problem), and a clustering algorithm to reduce the search space.

In the simultaneous tailored approach,^{13,26,27} the CSS condition is included as a constraint of the optimization problem in which both design variables and initial states are optimized. In this way, the CSS condition must be guaranteed only at the optimal point with a considerable savings of computational time. On the other hand, since also the initial state variables need to be optimized, the number of optimization variables is larger, and a gradient-based optimization algorithm is required. Several approaches are available to deal with the additional CSS constraint: in Kim et al.,²⁸ the difference between the initial and final state of the reactor (temperature, pressure, gaseous component concentration, and the solid component concentration) is constrained to be lower than a given, very small, positive variable (ϵ). On the other hand, other authors²⁹ adopt the approach proposed by Ko,³⁰ i.e., including ϵ as a penalty term in the objective function, which is then minimized.

In the equation oriented approach,¹⁴ the partial differential equations (PDEs) of the column model are expressed as constraints of the optimization problem, such as to impose the CSS condition. Because of the discretization of the PDE

equations, the authors have to deal with a large scale nonlinear optimization problem (with up to 50 000 variables and constraints) and adopt an interior point optimization algorithm. However, only simple adsorption models can be tackled with this approach, due to the numerical instabilities introduced by steep front profiles,³¹ and the nonlinear and nonconvex nature of the objectives and constraints that may lead to an unfeasible solution or to a local minimum.

As an additional challenge, adsorption processes feature multiple contrasting performances. For example, purity and recovery of the target component identify an attainable region of separation, whose upper limit is defined by a Pareto front. A similar behavior is found for the productivity (defined as the amount of desired product per unit cycle time per amount of adsorbent material) and the energy consumption required for the separation.³² While cost evaluation would enable the weighing of productivity and energy consumption, it would also introduce significant uncertainties, especially for new materials and processes. Clearly, the identification of the most performing design and operating variables of the process calls for a multiobjective optimization approach. On the other hand, multiobjective optimization is significantly more computationally demanding compared to single objective optimization, because of the need of finding a set of optimal solutions.

In this work, we tackle the multiobjective optimization of adsorption cycles. While most of the studies use multiobjective evolutionary algorithms for this purpose,^{18,20} we devise a potentially more efficient algorithm. It is an adaptation of the model-based derivative-free algorithm called multilevel coordinate search (MCS) originally proposed by Huyer and Neumaier.³³ Notably, the advantage of model-based derivative-free algorithms compared to evolutionary algorithms is the exploitation of local models of the computationally expensive functions so as to save evaluations.³⁴ In addition, MCS is deterministic. It features convergence guarantees on continuous functions (but only for an infinite number of “splitting levels,” see subsection 4.1), and according to the benchmarking work reported in Rios and Sahinidis,³⁵ it is the best performing noncommercial algorithm for the global optimization of bound-constrained nonconvex smooth problems. In this work, MCS is extended to handle general nonlinear constraints (both algebraic and simulation-based constraints) in addition to bound constraints, and to efficiently identify the set of Pareto optimal solutions (coupling parallel computing with look-up tables). We denote this new algorithm MO-MCS.

This paper is organized as follows: section 2 describes the multiobjective optimization problem tackled in this work. The model of the adsorption bed and the numerical solution approach are outlined in section 3. The optimization algorithm is detailed in section 4. Finally, sections 5 and 6 show the performance of the MO-MCS algorithm for the optimization of TSA and PSA cycles.

2. PROBLEM STATEMENT

The constrained, nonlinear, multiobjective optimization problem tackled in this work can be stated as follows

$$\begin{aligned}
 & \begin{cases} \text{Min } e(\underline{x}) \\ \text{Max } Pr(\underline{x}) \end{cases} \\
 & \text{s.t.:} \\
 & r_{\text{CO}_2} \geq r_{\text{CO}_2, \min} \\
 & \phi_{\text{CO}_2} \geq \phi_{\text{CO}_2, \min} \\
 & \underline{\text{lb}} \leq \underline{x} \leq \underline{\text{ub}} \\
 & \underline{x} \in R^n
 \end{aligned} \quad (1)$$

where e denotes energy-related objective function, Pr denotes the productivity (defined as the amount of CO_2 that is produced by a unit mass of sorbent in a unit of time), and \underline{x} is the vector of decision variables (duration of the time steps, volumetric feed flow rate, and operating temperatures/pressures); r_{CO_2} and ϕ_{CO_2} denote, respectively, the CO_2 recovery (defined as the amount of CO_2 produced over the amount of CO_2 input) and the CO_2 purity; $\underline{\text{lb}}$ and $\underline{\text{ub}}$ define the upper and lower bounds on the decision variables \underline{x} . Concerning the energy need e , the energy consumption is considered when dealing with the PSA process, while the exergy consumption is considered for TSA processes, as it combines electric and thermal power consumption. In both cases, the energy need is an indication of the operational costs. On the other hand, the productivity serves as an indirect index of the capital investment of the adsorption processes. Larger productivity enables the use of more compact equipment units for the same yield; hence it allows minimizing the equipment cost regardless of the uncertainty of cost related data.

As explained in section 1, in the black-box optimization approach, the functions $e(\underline{x})$, $Pr(\underline{x})$, $r_{\text{CO}_2}(\underline{x})$, and $\phi_{\text{CO}_2}(\underline{x})$ are essentially black-box functions whose output is determined by executing the cycle simulation code (see next section).

3. MODELING CYCLIC ADSORPTION PROCESSES

The cyclic adsorption process is simulated by using an in-house model (named FAST) specifically developed for adsorption processes within fixed beds; the main features of the tool are reported in the following, while details can be found elsewhere.^{36–38} The model consists of a system of partial differential equations (PDE) that account for the mass, energy, and momentum balances within the column. The set of PDEs, which is summarized in the Supporting Information, is solved by discretizing the column along the axial coordinate in finite volumes of constant width. The cycle is described by simulating a single column undergoing each step of the cycle; therefore, the model iterates until cyclic steady state (CSS) conditions are reached, namely, once the change in composition, temperature, and material balance between two different iterations is below a predefined tolerance and the overall material balance of the cycle is satisfied. The model is written in FORTRAN, which limits the computation time while also providing the developer with great flexibility. The set of PDEs is solved calling the LSODE solver,³⁹ which replaces the ODEs with difference equations and solves them step by step. In our implementation, LSODE uses the backward differentiation formula (BDF) method, which is recommended for stiff problems, and makes use of a modified Newton iteration method to correct the initial estimate of the solution.

The main assumptions of the adsorption model are

- Radial gradients are negligible
- The gas phase and solid phase are in thermal equilibrium

- Axial dispersion and conductivity are neglected
- Mass and heat transfer coefficients, heat of adsorption, and molar heat capacities are assumed to be constant and do not depend on temperature, pressure, or composition
- The mass transfer kinetics are described with the linear driving force (LDF) model

The model has proven to be a very powerful tool in both reproducing experimental results and designing new adsorption cycles. As far as the experimental validation is concerned, the software was employed for PSA-based separation of H_2/CO_2 mixtures⁴⁰ and multicomponent $\text{H}_2/\text{CO}_2/\text{N}_2$ mixtures⁴¹ and TSA-based separation of N_2/CO_2 .⁴² Furthermore, the model has been employed for the design of PSA and TSA adsorption cycles^{32,37} and for the assessment of new sorbents.^{43,44}

The column design considered for all calculations matches the column adopted in the P/TSA test rig present at ETH Zurich and thoroughly described elsewhere.^{41,42} It is worth mentioning that when running PSA simulations, the column is assumed to be adiabatic, while a constant heat transfer coefficient between the bed and the column wall is assumed for TSA applications (two constant values are assumed, which mimic stagnant conditions or gas flowing within the bed, respectively). The axial discretization considers volumes of approximately 5 cm width, while the time discretization is implicit and defined by the BDF algorithm. The convergence to CSS conditions is guaranteed by ensuring that both the overall mass balances are satisfied and that the variation of temperature and concentration profiles between successive iterations is negligible. The resulting computational time for achieving CSS is between 30 and 180 s for PSA and ranges from 10 to 90 s for TSA.

The operational variables that have to be provided as input consist of (i) the time length of each step of the cycle, (ii) the gas flow rate (i.e., the gas inlet velocity), (iii) the pressure levels in the cycle, especially for PSA, and (iv) the temperature of the heating and cooling fluid for TSA operation. Even though the simulation tools have proven to be robust in converging, there could be combinations of operational variables that prevent the solver from finding a feasible solution. Notably, 2.5% and 5.3% of the simulations did not converge for PSA and TSA, respectively.

4. MO-MCS ALGORITHM

In this section, the MO-MCS algorithm is presented. In subsection 4.1, we briefly review the original MCS algorithm, as proposed by Huyer and Neumaier.³³ In subsection 4.2, we present the quadratic penalty approach devised to handle nonlinear constraints. In subsection 4.3, we propose an approach for adapting the original MCS algorithm to multiobjective optimization problems, including an ad-hoc parallel computing architecture tightly coupled with the multiobjective formulation of the problem.

4.1. Multilevel Coordinate Search (MCS) Algorithm.

Let's consider a bound constrained problem, formulated as follows:

General bound constrained problem (P1)

$$\begin{aligned}
 & \min f(\underline{x}) \\
 & \text{s.t.:} \\
 & \underline{\text{lb}} \leq \underline{x} \leq \underline{\text{ub}} \\
 & \underline{x} \in R^n
 \end{aligned} \quad (2)$$

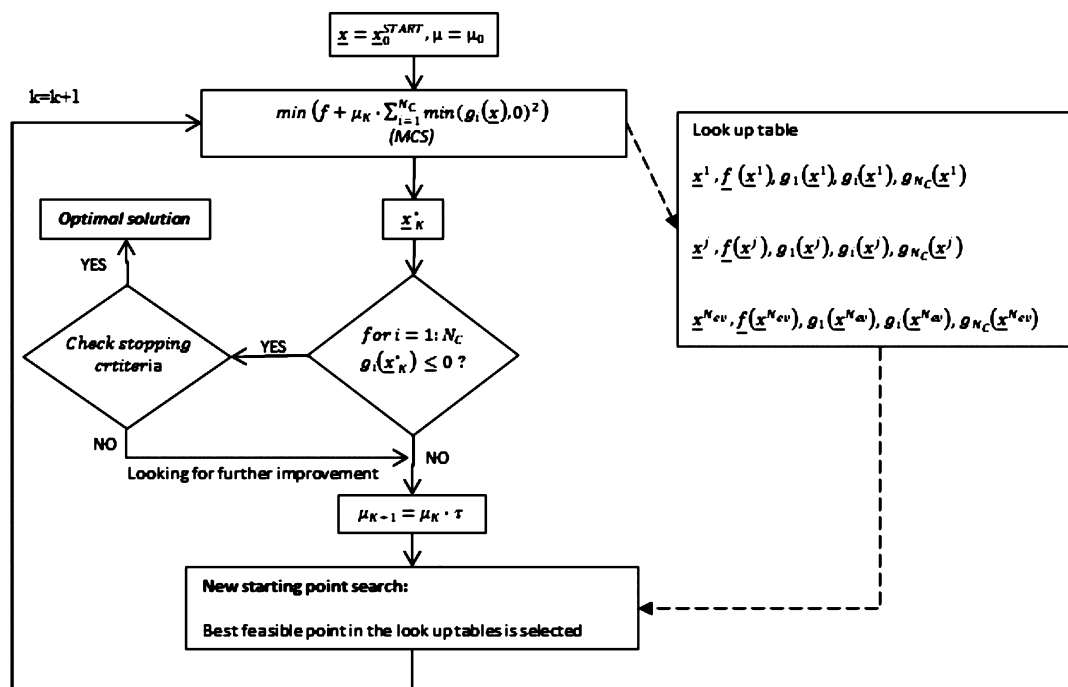


Figure 1. Block-flow diagram of the overall constrained MCS algorithm.

where \underline{x} denotes the vector of n decision variables, \underline{lb} and \underline{ub} are respectively the lower and upper bounds on the variables, and f is the objective function to be minimized.

The MCS algorithm aims at finding the global minimum of the problem by following a “branching without bounding” approach, which proceeds by dividing the search space into boxes and uses local optimization routines to refine the search in the most promising boxes. Multiple search techniques (i.e., dividing rectangle, coordinate search, “triple search,” model-based trust-region optimization) are iteratively called by the algorithm. For further details, the reader is referred to the work of Huyer and Neumaier.³³

The MCS algorithm features a convergence guarantee proved by the authors:³³ if the maximum number of splitting levels goes to infinite and the objective function f is continuous in the neighborhood of the global minimum, MCS is guaranteed to converge to the global minimum. The basic reason is that the points sampled by the algorithm would form a dense subset of R^n .

4.2. Extension of MCS to Constrained Optimization. In order to apply MCS for the optimization of adsorption cycles (see section 6), it is first necessary to include nonlinear constraints. The general formulation of the resulting constrained problems, as considered in this work, is shown in eq 3:

Nonlinear constrained problem (P2)

$$\min f(\underline{x})$$

s.t.:

$$1. g_i(\underline{x}) \leq 0 \quad \forall i = 1: N_c$$

$$2. \underline{lb} \leq \underline{x} \leq \underline{ub}$$

$$\underline{x} \in R^n$$

(3)

where f is the objective function, \underline{x} is the set of decision variables, g_i are the nonlinear constraints, and N_c is the number of constraints.

Several techniques have been developed to convert a nonlinear constrained optimization problem into a series of unconstrained problems, such as penalty methods, barrier methods, and augmented Lagrangian methods.⁴⁵ We have selected the quadratic penalty strategy because of the following reasons:

- When optimizing an adsorption cycle, the starting solution often does not meet the constraints (desired purity, recovery, etc.), and this would affect the effectiveness of barrier methods.⁴⁵

- In order to handle inequality constraints, augmented Lagrangian methods have to convert them into equality constraints by adding slack variables (one slack variable for each nonlinear constraints). These slack variables are added to the list of decision variables causing an increase of the problem dimensionality. While gradient-based methods can easily handle problems with hundreds of variables, the convergence rate of derivative-free methods (like MCS) is significantly affected by the number of variables.

- The nonlinear constraints in the adsorption cycle problem (section 2) are “relaxable” constraints since some degree of violation can be tolerated. For this reason, the penalty coefficients do not need to be raised to very high values so as to drive to zero the constraints’ violation. Limiting the penalty coefficients allows to moderate the ill conditioning issue of the Hessian matrix of the objective function minimized by MCS. It is worth noting that, although MCS does not directly use the Hessian of the objective function to find the search direction, it builds and optimizes (with an indefinite quadratic programming algorithm) a quadratic model which inherits the ill conditioning of the objective function. For this reason, it is advisable not to adopt excessively high penalty coefficients.

Using the quadratic penalty approach, the constrained nonlinear program (eq 3) is converted into the bound constrained problem shown in eq 4.

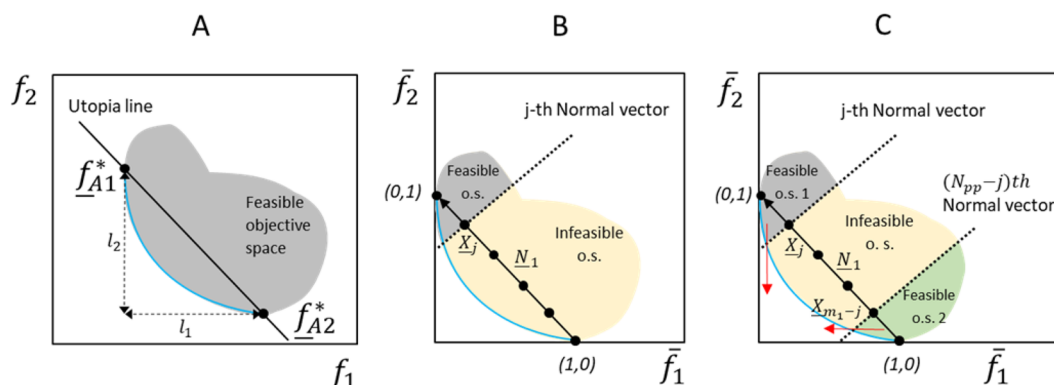


Figure 2. Implementation of the NNC method represented in the objective space (o.s.).

Bound constrained problem (P3)

$$\min Q(\underline{x}, \mu_k) = f(\underline{x}) + \mu_k \cdot \sum_{i=1}^{N_c} (\max[g_i(\underline{x}), 0])^2$$

$$\underline{lb} \leq \underline{x} \leq \underline{ub}$$

$$\underline{x} \in R^n \quad (4)$$

The original objective function is penalized by adding the summation of the square of the constraint violations times the penalty parameter μ_k (k is the iteration index counting the number of times the problem has been solved). Starting from an initial (user-defined) penalty parameter μ_0 , the bound-constrained problem is solved with the MCS algorithm. Then, the degree of violation of the constraints is checked, and if greater than the desired tolerance, the penalty parameter is increased ($\mu_{k+1} = \tau\mu_k$ with $\tau > 1$) and the penalized problem solved again. Convergence of the quadratic penalty approach to a local minimum is guaranteed even if the intermediate k -th penalized problems are not exactly solved (provided that the optimization accuracy progressively decreases tending to zero for k going to infinite⁴⁵). The drawback of this procedure is the need of solving the bound-constrained problem several times with increasing values of the penalty parameter μ_k . Due to the high computational cost of the black-box function (i.e., cycle simulation code), each optimization (iteration k) takes several hours, making this strategy not practical. On the other hand, to save computational time, one could think of avoiding the intermediate iterations and starting directly with a very high penalty parameter (static penalty approach). Beside the ill conditioning issue of the problem, which may affect the effectiveness of the local search step of MCS, starting directly with high penalty parameters increases the risk of finding local optima or even infeasible solutions, especially if the starting point is infeasible. Our preliminary computational experiments have shown that for some classes of problems with a narrow feasible region, it is not only necessary to start with a low μ_0 but also increase it very slowly (i.e., adopting $\tau = 1.15\text{--}1.5$).

The block-flow diagram of the overall constrained MCS algorithm is shown in Figure 1. In addition to the quadratic penalty approach, the key peculiarities of the algorithm are the use of a look-up table to save evaluations of the black-box function and a starting point search procedure to select the best starting point (among the ones previously sampled) for each run depending on the value of the penalty parameter μ_k .

Since the simulation of adsorption cyclic processes is computationally intensive, it is essential to save black box

function evaluations. Accordingly, we introduced a look-up table where the values of objective functions, constraint violations, and the corresponding set of decision variables are saved for all function evaluations. When MCS needs to sample a point, it first searches in the look-up table, and if the desired point has not been yet simulated, it executes the adsorption cycle simulation code. The adoption of look-up tables within MCS is particularly advantageous because:

(1) If the increase of μ from iteration k to the next, $k + 1$, is not excessive, the global minimum of the penalized problem will likely remain in the same region.

(2) If the same initialization list of splitting points is passed to MCS, in the first Sweep iteration it will proceed by dividing the feasible region in the same way, sampling the same base points as in the previous ($k - 1$) optimization run. This occurs because MCS is a deterministic algorithm without random components.

Along with look-up tables, a procedure was designed in order to select the best starting point for the subsequent optimization run ($k + 1$) with an increased penalty parameter. When an instance of problem (P2) is solved with fixed penalty μ_k , the algorithm may return an intermediate suboptimal or even infeasible solution (i.e., which does not meet all the constraints g_i because the penalty parameter is too low). When the penalty parameter is updated at step $k + 1$ (μ_{k+1}), the best feasible point in the look-up table is selected as the starting point for the $k + 1$ iteration.

4.3. Extension of MCS to Multiobjective Optimization and Parallelization Strategy.

A widely used technique for generating a Pareto set in multiobjective optimization is the weighted sums (WS) approach. The basic idea is to convert the multiobjective problem into a sequence of single objective problems, where a convenient convex combination of the two objective functions is minimized.⁴⁶ However, the WS method has two main drawbacks: (i) it cannot find nonconvex Pareto frontiers, and (ii) obtaining an even distribution of solutions on the Pareto frontier is difficult.⁴⁷ Ko and Moon¹² proposed an extension of the WS method, which is able to return well-spaced Pareto points, but at the cost of several function evaluations, with therefore scarce applicability to computationally intensive problems. Another technique for generating the Pareto set is the normal-boundary intersection (NBI) method.⁴⁸ On the one hand, the NBI method can achieve fairly uniform distribution of Pareto-optimal solutions and can treat problems with nonconvex Pareto fronts by imposing equality constraints along equally spaced lines. On the other hand, including equality constraints makes the optimization problems more challenging, and the NBI approach may return dominated solutions (not-Pareto opti-

mal).⁴⁹ These two issues have been addressed by the normalized normal constraint (NNC) method.⁵⁰ The NNC method converts the multiobjective problem into a sequence of single objective nonlinear constrained problems (NLP) subject to normalized normal inequality constraints which limit the feasible objective space. As shown in Figure 2B, the j th Pareto point is obtained by minimizing one of the two objectives subject to the j th normalized normal constraint. Thus, a set of equally spaced Pareto points is obtained by performing several single objective optimizations with shifted normalized normal constraints. Compared to other approaches, NNC can deal with nonconvex Pareto fronts and return equally spaced Pareto solutions without the risk of finding dominated solutions.⁵⁰

Owing to its above-mentioned advantages, we selected the NNC approach to convert the constrained biobjective optimization problem (eq 5) into a set of single objective constrained problems which can be efficiently tackled with the MCS algorithm. Moreover, we devised an implementation capable of exploiting multiple processors (workers) in parallel computing while maximizing the benefit of the look-up table.

Given the constrained nonlinear biobjective in eq 5:

$$\begin{aligned} &\text{Constrained nonlinear biobjective problem (P4)} \\ &\min \{f_1(\underline{x}), f_2(\underline{x})\} \\ &\text{s.t.:} \\ &g_i(\underline{x}) \geq 0 \quad i = 1: N_C \\ &\underline{\text{lb}} \leq \underline{x} \leq \underline{\text{ub}} \\ &\underline{x} \in R^n \end{aligned} \quad (5)$$

Our implementation of the NNC method consists of the following steps:

1. **Anchor points calculation.** The extreme points of the Pareto front (anchor points) are found solving two single objective instances (A_1 and A_2) of problem P4, minimizing respectively one of the two objectives while neglecting the other.

$$\begin{aligned} &\text{Problem (P5): Minimization of } f_1 \\ &\min f_1(\underline{x}) \\ &\text{s.t.:} \\ &g_i(\underline{x}) \geq 0 \quad i = 1: N_C \\ &\underline{\text{lb}} \leq \underline{x} \leq \underline{\text{ub}} \\ &\underline{x} \in R^n \end{aligned} \quad (6)$$

$$\begin{aligned} &\text{Problem (P6): Minimization of } f_2 \\ &\min f_2(\underline{x}) \\ &\text{s.t.:} \\ &g_i(\underline{x}) \geq 0 \quad i = 1: N_C \\ &\underline{\text{lb}} \leq \underline{x} \leq \underline{\text{ub}} \\ &\underline{x} \in R^n \end{aligned} \quad (7)$$

Let us define $f_{A1}^* = [f_1(\underline{x}_{A1}^*), f_2(\underline{x}_{A1}^*)]$ and $f_{A2}^* = [f_1(\underline{x}_{A2}^*), f_2(\underline{x}_{A2}^*)]$ the corresponding optimal solutions obtained. The line joining them is called "utopia line" (see Figure 2A).

2. **Normalization.** The values of the objective functions are normalized according to the utopia points:

$$l_1 = f_1(\underline{x}_{A2}^*) - f_1(\underline{x}_{A1}^*) \quad (8)$$

$$l_2 = f_2(\underline{x}_{A1}^*) - f_2(\underline{x}_{A2}^*) \quad (9)$$

$$\bar{f}(\underline{x}) = \left(\frac{f_1(\underline{x}) - f_1(\underline{x}_{A1}^*)}{l_1}, \frac{f_2(\underline{x}) - f_2(\underline{x}_{A2}^*)}{l_2} \right) \quad (10)$$

3. **Utopia line vector definition.** The Utopia line vector \underline{N}_1 is a vector connecting the two extreme points of the Pareto front, defined in eq 11.

$$\underline{N}_1 = \underline{f}_{A2}^* - \underline{f}_{A1}^* \quad (11)$$

4. **Definition of the number of Pareto points.** From this step on, it is necessary to set the desired number of points on the optimal Pareto front, in addition to the two extreme points. According to the proposed approach, every Pareto point is the solution of a single objective optimization problem, and each problem is assigned to a specific worker. In order to maximize the benefit of parallel computing, we recommend the selection of a number of internal Pareto points that is a multiple of the number of workers of the machine (N_W). In this way, the calculation of the same number of internal Pareto points (here called "number of worker turns," N_{WT}) is assigned to each worker. Then, the total number of Pareto points (N_{PP} ; including the two extreme points) and the normalized increment δ can be obtained as shown in eqs 12 and 13.

$$N_{PP} = (N_W \cdot N_{WT}) + 2 \quad (12)$$

$$\delta = \frac{1}{N_{PP} - 1} \quad (13)$$

5. **Utopia line points generation.** Utopia line points (\underline{X}_j) are a series of equally spaced points along the utopia line.

$$\underline{x}_j = \alpha_{1,j} \cdot \underline{f}_1^* + \alpha_{2,j} \cdot \underline{f}_2^* \quad (14)$$

$$0 \leq \alpha_{1,j} \leq 1 \quad (15)$$

$$\alpha_{1,0} = 0 \quad (16)$$

$$\alpha_{1,j+1} = \alpha_{1,j} + \delta \quad \forall j = 0: N_{PP} - 1 \quad (17)$$

$$\alpha_{2,j} = 1 - \alpha_{1,j} \quad \forall j = 0: N_{PP} \quad (18)$$

We note that $\alpha_{1,j}$ is incremented by δ between 0 and 1. Values $\alpha_{1,0} = 0$ and $\alpha_{1,N_{PP}} = 1$ are discarded, since they would lead to the same Pareto points found in step 1.

6. **Generation of the Pareto points.** Using the set of evenly distributed Utopia line points, a corresponding set of Pareto points is generated by solving a sequence of optimization problems. The j -th optimization run yields a Pareto optimal point and is associated with utopia line point \underline{X}_j by the addition to the original problem P4 of a constraint normal to the utopia vector and passing through the j -th utopia point. In our approach, each optimization run is assigned to a specific worker, and two different types of NNCs (h^1 and h^2) are considered simultaneously, as shown in Figure 2C. Each NNC type starts from an opposite end of the Pareto front and cuts off opposite sides of the objective feasible space. Every optimization run is subject to either h^1 or h^2 NNC. Since each optimization run is carried out by one and only one worker, we can distinguish between

• $(N_{PP}/2) - 1$ optimizations runs, which are associated with the $(N_{PP}/2) - 1$ utopia line points closer to A_1 and are subject to

the h^2 constraint while minimizing \bar{f}_2 , represented by problem P7 (eq 19)

- The other $(N_{pp}/2) - 1$ points, closer to A_2 which will be subject to the h^1 constraint while minimizing \bar{f}_1 , represented by problem P8 (eq 20).

Problem (P7): $\forall j: 1 \leq j \leq N_{pp}/2$

$$\min \bar{f}_2(\underline{x})$$

s.t.:

$$g_i(\underline{x}) \geq 0 \quad i = 1: N_C$$

$$h_j^2(\underline{x}) = -N_1 \cdot (\bar{f}(\underline{x}) - \underline{x}_j)^T \geq 0$$

$$\underline{lb} \leq \underline{x} \leq \underline{ub} \quad (19)$$

Problem (P8): $\forall j: N_{pp}/2 < j \leq N_{pp} - 1$

s.t.:

$$g_i(\underline{x}) \geq 0 \quad i = 1: N_C$$

$$h_j^1(\underline{x}) = N_1 \cdot (\bar{f}(\underline{x}) - \underline{x}_j)^T \geq 0$$

$$\underline{lb} \leq \underline{x} \leq \underline{ub} \quad (20)$$

As for the additional NNC constraint added in each optimization run, MO-MCS adopts the same methodology used for the black box constraints and outlined in section 4.2.

Clearly, this overall approach makes full use of the algorithm parallelization: half of the workers are assigned problems of type P7, and the other half, problems of type P8. As illustrated in Figure 3, the optimization runs tackled in the first worker turn

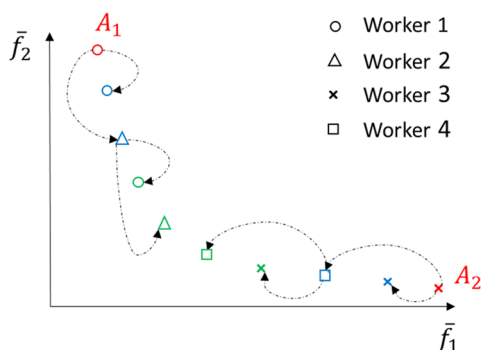


Figure 3. Example of the generation sequence of the Pareto front for a case with four workers and two worker sweeps. Each shape corresponds to a specific worker. The anchor points are derived in parallel by two different workers and marked in red. The points derived in parallel during the first worker sweep are marked in blue, and the ones derived in parallel during the second worker sweep are marked in green. Dashed lines link the starting point used to find each Pareto point. Points originated from the same anchor point are subject to a NNC constraint with the same direction.

lead to the Pareto points closer to the anchor points. Anchor point A_1 is used as a starting point for problems of type P7, while anchor point A_2 is the starting point for problems of type P8. We observe that this technique guarantees the feasibility of the starting point for every optimization run, for problems of both types P7 and P8. If more worker turns are needed, the starting points for both classes of problems are updated, choosing the last Pareto points derived in the previous worker turn. In this way, the Pareto front is built symmetrically from the ends toward the

center, as represented in Figure 3. The complete structure of the algorithm is represented in Figure 4. A look-up table is included, where information about already performed black box function evaluation is stored. Whenever a new function evaluation is needed, the algorithm first scans the look-up table to check if the point has already been explored by a worker in the same worker turn or in preceding worker turns. If so, results are retrieved from the look-up table, and the computational time for a black box function evaluation is saved. The last worker turns operate in a more beneficial state, since a higher number of points has been explored and is available in the look-up table.

As far as the algorithm parameters of MO-MCS are concerned, good results were obtained with those reported in Table 1 (and adopted for all the test cases of this work). It is worth noting that the number of worker turns is not critical for the quality of the results, and it is set depending on the desired number of Pareto points on the front. It is also worth stressing that a large number of additional Pareto points is found during the execution of the algorithm, which can be retrieved from the look-up tables without any additional computational expense. However, no guarantee on the feasibility, number, and spacing of such additional points exists. This means that it is likely, but not guaranteed, to obtain a finer approximation of the Pareto curve than the number of points set as input.

It is worth noticing that the search for extreme Pareto points involves one less constraint (the NNC) and hence is less difficult, so the initial value of the penalty factor can be higher than that used for the calculation of the internal Pareto points (as suggested in Table 1). When possible, normalization of the constraints is recommended so as to balance their weights.

If the user adopts a poor normalization of the constraints or too high/low initial values of the penalty parameters, one may obtain: (i) a perfectly spaced but sub-optimal Pareto curve (if the weight of the NNC constraint is too large compared to the weight of the other constraints of the problem), or (ii) a poorly spaced Pareto curve (if the weight of the NNC constraint is too small compared to the one of the other constraints of the problem).

5. MO-MCS BENCHMARKING

In this section, the performances of MO-MCS are compared against other well-known, derivative-free commercial algorithms. A similar benchmarking procedure, yet for an earlier version of MO-MCS, is reported in Joss et al.⁵¹ Because of the novel features, which improve considerably the computational performances of the algorithm, an updated benchmarking is reported here. Moreover, in this work, we have modified (e.g., enabling parallel computing) and tuned all considered algorithms to the best of our capabilities.

The test problem is a five-step TSA cycle, as considered in Joss et al.,⁵¹ where constraints on the purity and recovery of the separated CO_2 are considered. The objective functions are the productivity of the cycle (to be maximized) and the energy consumption (to be minimized). The decision variables are the duration of four out of five time steps (two time steps require equal duration). As far as the benchmarking procedure is concerned, the process details of the test problem are not relevant, and the reader is referred to ref 51 for further details.

The algorithms selected for the benchmarking are

1. Nonsorting Genetic Algorithm II (NSGA II), a genetic algorithm presented by Deb et al. in ref 16, which can handle multiobjective constrained problems with a fixed penalty approach.

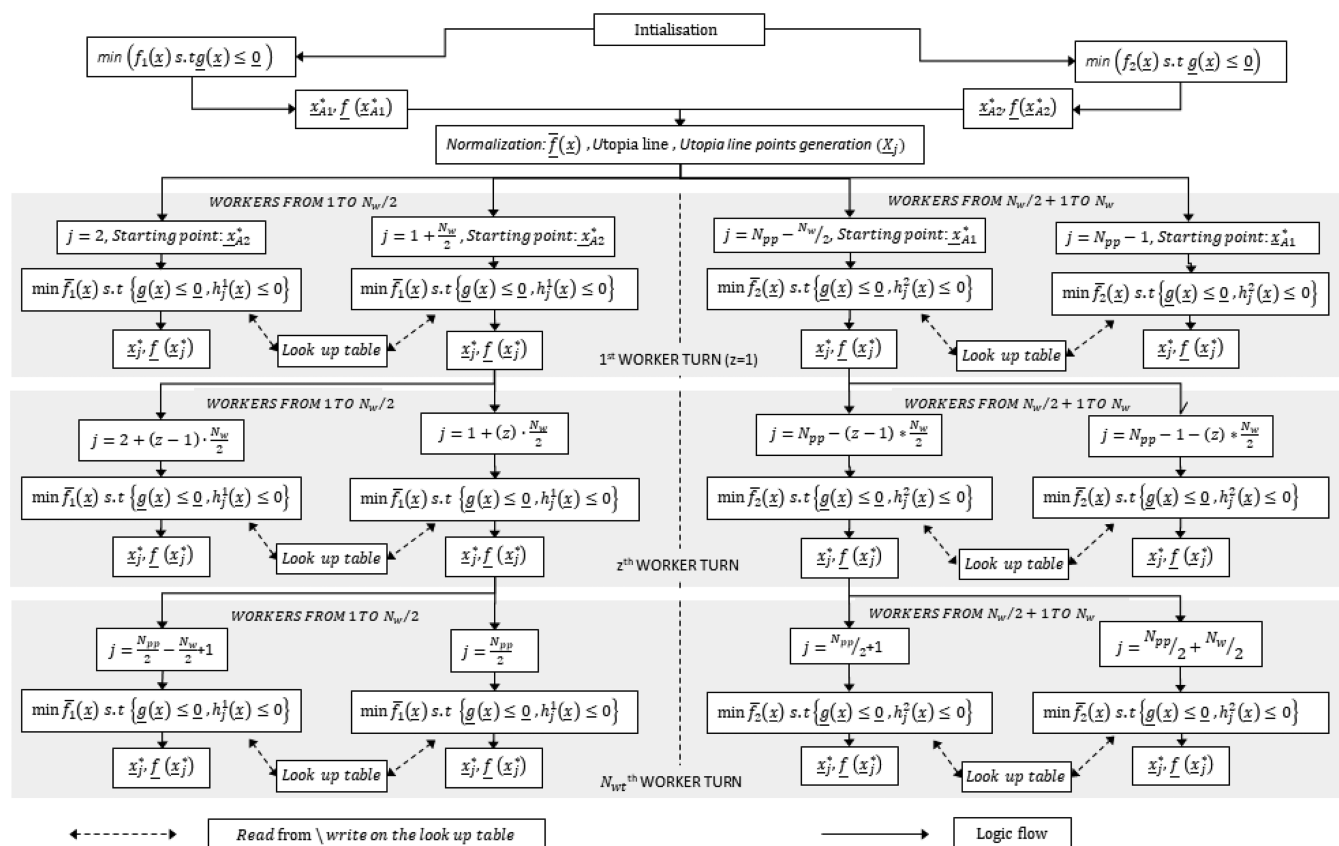


Figure 4. General scheme for the parallelization of MO-MCS. After the initialization step, the two extreme Pareto points are derived in parallel. The problem is normalized, and the utopia line points are generated. The starting points for the first worker turn are assigned. Index j spans the instances of the problem to be solved, each assigned to a specific worker and associated with a specific NNC. In each worker turn (z), the first $N_W/2$ instances tackled are subject to h^1 while minimizing f_1 , while the other $N_W/2$ problems are subject to h^2 while minimizing f_2 . After each worker turn, the starting points are updated, and a new worker turn is performed, until $z = N_{WT}$. During the search for each Pareto point (marked with *), each worker can read information from the look-up table (which is common to all workers) and write the results of new sampled points. In this way, each worker can access information on the black box function evaluations performed by other workers in the same work turn, as well as the ones performed in previous worker turns.

Table 1. Suggested Parameter Tuning for MO-MCS

| parameter | suggested value |
|---|-----------------|
| initial value of the penalty parameter (μ_0) for extreme Pareto points search: | 5 |
| initial value of the penalty parameter (μ_0) for internal Pareto points search: | 0.1 |
| initial value of the penalty parameter (μ_{0NNC}) for NNC constraint: | 0.1 |
| penalty parameter increase rate (μ_{k+1}/μ_k) | 1.5 |

2. Multi-Objective Particle Swarm Optimization (MOPSO), proposed by Coello et al. in ref 52. The original algorithm presents no built-in parallel computing approach. For the sake of this article, the MOPSO algorithm was updated to include such skill. In particular, the evaluation of each particle in the current population is parallelized, for both the initialization step and the subsequent repository updates.

3. Direct Multi Search (DMS), presented by Custodio et al. in ref 53.

Both MOPSO and DMS algorithms have no standard routine to deal with constrained problems, so the same fixed penalty strategy available in NSGA II is extended to the other benchmarking algorithms. The test problem is first solved with the MO-MCS algorithm; then, the number of function evaluations required by the MO-MCS (reported in Table 2) is

Table 2. Benchmarking Optimization Parameters

| | MO-MCS | NSGAI | MOPSO | DMS |
|---|-------------------------|----------------|--------------|-----|
| workstation used to perform optimization runs | Intel Xeon CPU 2.79 GHz | | | |
| maximum number of function evaluations allowed (stopping criterion) | 1950 | | | |
| number of workers | 4 | | | |
| number of well-spaced Pareto points | 10 | | | |
| population size | | 50 | 50 | 100 |
| parallel computing | YES | YES, available | YES, adapted | NO |

set as a stopping criterion for the other algorithms. Such an upper limit is obtained by setting broad stopping criteria for the MO-MCS (i.e., max 50 function evaluations for each fixed penalty run and max 150 function evaluations for each Pareto point) and observing the algorithm convergence within these limits. The fixed penalty approach adopted to make NSGA II, MOPSO, and DMS suitable for constrained problems calls for a fine-tuning of the penalty parameter: too low a value may lead to unfeasible results, while too high a value may ill-condition the problem and lead to suboptimal results. In order to perform a fair comparison, several values of penalty factor have to be tested, since it is not possible to determine a priori an effective value. The values considered for the penalty parameter are 0, 2, 5, 20, and 250. As

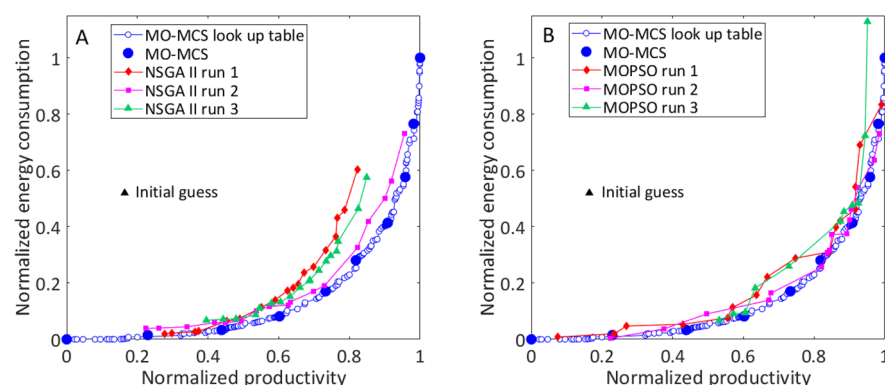


Figure 5. Normalized Pareto fronts for benchmarked algorithms (A, NSGA II runs; B, MOPSO runs) compared to the normalized Pareto front obtained with MO-MCS. Both the output from MO-MCS (filled dots) and the additional Pareto points retrieved from the lookup tables (empty dots) are considered. In addition, the initial guess adopted for all runs is reported. The shape of the Pareto front is because higher productivities are associated with higher energy consumptions, as expected.

expected, lower values of the penalty parameters perform better, but runs with a penalty factor lower than 5 are discarded, because they lead to unfeasible results. We report here only the best results obtained. Notably, MO-MCS features an autoadaptive strategy for the penalty factor to solve such a tuning issue that may perform well enough without tuning. While we do not report here the time for tuning, as it is very much dependent on the user, it is useful to keep this in mind when tackling a similar optimization problem.

Since the NSGA II and MOPSO are stochastic meta-heuristic algorithms, five runs of each algorithm were performed, and the best three runs are reported in Figure 5A and B. On the other hand, multiple runs are not necessary for DMS, as it is a deterministic algorithm. The run with the best penalty parameter is reported in Figure 6. All the results are plotted along with MO-MCS outcome. All the Pareto fronts are plotted in dimensionless form with the normalized productivity on the *x*-axis and the normalized energy consumption on the *y*-axis.

The results reported in Figure 5A show that MO-MCS outperforms NSGAII, since with a single run it can provide a Pareto front which is quite well spaced and wider, and it

dominates all the Pareto fronts obtained with five runs of NSGAII; i.e., for a fixed productivity, MO-MCS finds points with lower (or equal) energy consumption. The computational time required for a single run of NSGAII is 24% shorter than the time required from MO-MCS (see Table 3), thanks to the more

Table 3. Benchmarking Optimization Results^a

| | index M1, % | index M2 | index M3 | average computational time for a single run, h |
|---|----------------|-------------|-------------|---|
| MO-MCS | 0.19% | 183 | 1.41 | 1.81 |
| NSGA II (best envelope on 5 runs) | 2.59% | 28 | 1.18 | 1.37 |
| MOPSO (best envelope on 5 runs) | 1.21% | 24 | 1.26 | 1.33 |
| DMS | 1.87% | 174 | 1.33 | 3.02 |

^aIndex M1 measures the distance of the obtained Pareto front from the overall best envelope considering all Pareto fronts. Index M2 expresses the number of well-spaced particles in the Pareto fronts, while index M3 measures the extension of the Pareto fronts.

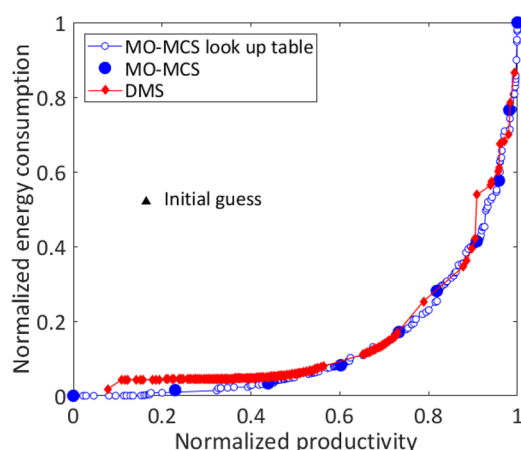


Figure 6. Normalized Pareto fronts for DMS algorithm compared to the normalized Pareto front obtained with MO-MCS. Both the output from MO-MCS (filled dots) and the additional Pareto points retrieved from the lookup tables (empty dots) are considered. In addition, the initial guess adopted for all runs is reported. The shape of the Pareto front is because higher productivities are associated with higher energy consumptions, as expected.

efficient use of parallel computing: in fact, MO-MCS includes a first step in which just two workers are exploited. On the other hand, it is important to remark that NSGAII required multiple runs to obtain a good approximation of the Pareto front, thus its total required computational time is considerably higher than a single run of MO-MCS.

In Figure 5B, the same Pareto front obtained from MO-MCS is compared to the results obtained from MOPSO. Three different runs are reported for MOPSO, and in this case, the best runs are obtained adopting the same penalty factor. Even if MOPSO is able to find some points laying on the optimal Pareto front, MO-MCS provides a better distributed Pareto front with a larger extension and number of Pareto points. The computational time required by a single run of MOPSO is slightly shorter than that of NSGA II, and indeed shorter than that for MO-MCS, as shown in Table 3. However, due to the need for making multiple runs of MOPSO, MO-MCS is much more efficient from a computational point of view.

In Figure 6, the Pareto fronts obtained from MO-MCS and DMS are compared. DMS can find some points laying on the optimal Pareto front identified by MO-MCS, although its Pareto front is not as well spaced and extended. From the computational time perspective, DMS is the slower among all the considered

optimizers because the code used for the tests does not exploit parallel computing (although Custódio et al.⁵³ suggest the possibility of parallelizing the poll steps, this option was not available in the most updated version (0.3) of the code). When parallelizing the poll step, even if the computation time is drastically reduced, we found that the resulting Pareto front is completely dominated by the one obtained with sequential computing. This is due to the fact that in the sequential computing approach it is possible to determine the non-dominated Pareto set after every single function evaluation (i.e., discard the dominated particles), while in the parallel computing approach such dominance relationships can be established only after each polling step, reducing the amount of information available during the polling step.

The performances of the four algorithms are quantified according to Zitzler's indices⁵⁴ and are reported in Table 3. Index M1 expresses the average distance from the Pareto optimal set, which is approximated with the envelope of all the Pareto fronts derived in the benchmarking procedure. Index M3 uses the maximum extent in each dimension to measure the spreading of the Pareto front. Index M2 aims at measuring the number of well spaced points in a Pareto front. This index includes a parameter σ , which is the minimum spacing in order to consider a Pareto to be well spaced. In this paper, we assume such spacing according to eq 21; i.e., the minimum satisfactory spacing is the one between $N_{\text{particles}}$ equidistributed along a line connecting Pareto extreme points in the normalized plane.

$$\sigma = \frac{\sqrt{2}}{(N_{\text{particles}} - 1)} \quad (21)$$

Since for NSGA-II and MOPSO the performance of the Pareto front will vary for the different runs, we decided to evaluate the performance indices based on the outer envelope of all Pareto points considering the three different optimization runs.

Even if the competing algorithms have an advantage because of the larger number of runs executed for each instance, MO-MCS performs better in terms of Pareto optimality (M1), spacing (M2), and width (M3). MOPSO ranks second in optimality of the returned Pareto curve (second lowest M1), but its spacing and spread are poorer than the ones achieved with MO-MCS. DMS trails MO-MCS according to M3 and M2, thanks to the large number of nondominated points; however, it shows longer computational time and worse M1 than MOPSO. NSGA-II returns Pareto curves with the lowest extension and hence ranks last according to index M3. MO-MCS also outperforms NSGA-II according to M1, since also the distance from the best Pareto envelope is penalized by the reduced extension of the Pareto curve.

6. CASE STUDIES

In the following, we apply MO-MCS to more challenging case studies, where fixed bed cycles are designed to fit the purpose of CCS. The considered processes are (i) a TSA cycle for the separation of CO₂ from ambient pressure fluegas and (ii) a PSA cycle for the separation of H₂ and CO₂ from a pressurized syngas. Not only do these two processes present key features that call for rigorous yet challenging optimization but they are also of great interest as alternative separation technologies to the more consolidated liquid scrubbing.^{3,32,55–57} Notably, the two case studies allow the testing of MO-MCS with two processes that, while sharing features of adsorption-based technologies, significantly differ in terms of driving force and typical time

length of the cycle steps. Indeed, a TSA cycle is controlled by heat exchange, which may take several minutes to bring the reactor to the desired conditions, while a PSA cycle is controlled by pressure change, which is notably fast. This also affects the resolution of the cycle simulation, as the TSA is inherently longer, while the sharp change of fronts in PSA makes the resolution more stiff.

6.1. TSA Cycle. The considered TSA cycle was presented in Joss et al.³² (referred to as *Cycle D*) as an advanced cycle for the separation of CO₂ starting from a binary CO₂–N₂ mixture at ambient pressure. The cycle is shown in Figure 7. Compared to

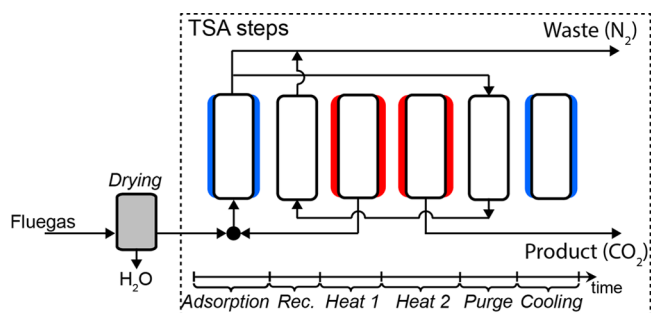


Figure 7. TSA cycle for production of CO₂ from N₂–CO₂ fluegas, which is considered to be dried beforehand.

the simplest TSA cycle, which comprises an adsorption, a heating, and a cooling step, this cycle features the following additional steps: (i) a rinse step, which is conducted after the adsorption by recycling a CO₂-rich gas, (ii) a preliminary heating step following the rinse, where the effluent stream is recycled to the adsorption step, and (iii) a purge step, which is carried out after the heating by purging the column with N₂, and which provides the gas to the rinse. Joss et al. showed that this cycle provides excellent separation performance with energy and productivity similar to amine scrubbing.³² It is worth stressing that with this work we do not aim at investigating the TSA process comprehensively (e.g., no drying or scheduling has been considered) but rather at explaining the physical rationale behind the optimal points identified by MO-MCS.

The fluegas entering the column has the following molar composition: N₂, 88%; CO₂, 12%.

The considered sorbent material is as in Joss et al.,³² i.e., a commercial zeolite 13X. Heating and cooling are carried out with an external thermal fluid so as to preserve the CO₂ purity. Eight decision variables are considered in the optimization: the time steps of adsorption, heating1, heating2, rinse—which is also the purge step—and cooling, the temperature of the heating fluid, the mass flow rate of inlet fluegas per column, and the mass flow rate of nitrogen used as a purge gas. The list of variables along with the corresponding range of values is reported in Table 4.

Problem Formulation. The optimization problem for the TSA test case is formulated in eq 22. The objective functions considered are the exergy equivalent (ex^{TSA}) of the cycle thermal power demand \dot{Q}_{TSA} (specific to the kilograms of CO₂ separated), as defined in eq 23, and the productivity. ex^{TSA} is selected to take into account the different availability contents of heat at different temperatures.

In terms of separation performance, the TSA optimization is constrained to achieve a CO₂ purity ($\phi_{\text{CO}_2\text{min}}^{\text{TSA}}$) of 96% and a CO₂ recovery ($r_{\text{CO}_2\text{min}}^{\text{TSA}}$) of 90%.

Table 4. List of the Decision Variables for the Optimization of the TSA Cycle and Their Respective Bounds

| decision variables | upper – lower bounds values | | initial guess |
|---|-----------------------------------|-----------------------------------|------------------------|
| time steps for: | | | |
| 1. adsorption, t_{ads} [s] | 1. 10 | 1. 4000 | 1. 300 |
| 2. recycle (and purge, as they are equal), t_{rec} [s] | 2. 5 | 2. 1000 | 2. 25 |
| 3. heating 1, t_{H1} [s] | 3. 1 | 3. 2000 | 3. 150 |
| 4. heating 2, t_{H2} [s] | 4. 20 | 4. 4000 | 4. 1200 |
| 5. cooling, t_{C} [s] | 5. 20 | 5. 4000 | 5. 850 |
| 6. temperature of the heating fluid, T_{H} [K] | 6. 380 | 6. 450 | 6. 420 |
| 7. flow rate of inlet fluegas, m_{FG} [m ³ /s] | $7.3.5 \times 10^{-4} \times 0.7$ | $7.3.5 \times 10^{-4} \times 1.3$ | $7.3.5 \times 10^{-4}$ |
| 8. flow rate of the N ₂ recycled to purge, m_{N2} [m ³ /s] | $8.3.5 \times 10^{-4} \times 1.3$ | $8.3.5 \times 10^{-4} \times 1.3$ | $8.3.5 \times 10^{-4}$ |

$$\begin{cases} \min ex^{\text{TSA}}(\underline{x}) \\ \max Pr^{\text{TSA}}(\underline{x}) \end{cases}$$

subject to:

$$r_{\text{CO}_2} \geq r_{\text{CO}_2\text{min}}^{\text{TSA}}$$

$$\phi_{\text{CO}_2} \geq \phi_{\text{CO}_2\text{min}}^{\text{TSA}}$$

$$\underline{\text{lb}} \leq \underline{x} \leq \underline{\text{ub}}$$

$$\underline{x} \in R^n$$

(22)

$$ex^{\text{TSA}} = \dot{Q}_{\text{TSA}} \times \left(1 - \frac{T_0}{T_{\text{H}}}\right)$$

(23)

The parameters adopted to run the MO-MCS algorithm for the TSA case study are the same as suggested in section 5, except for the ones reported in Table 5: due to the higher complexity of

Table 5. Mo-MCS Parameters for the TSA Study Case

| parameter | adopted value |
|--|---------------|
| number of workers | 8 |
| number of worker turn | 2 |
| number of Pareto points | 18 |
| maximum number of function evaluation for a fixed penalty parameter (N_{max_k}) | 150 |
| maximum number of function evaluation for a single Pareto point search ($N_{\text{max}_\text{tot}}$) | 800 |

the case study compared to the benchmarking problem, the maximum number of function evaluations is increased both for the fixed penalty runs and overall. In addition, a finer mesh for the Pareto front is specified, increasing the number of optimal Pareto points. The optimization run is carried out with the same hardware used for the benchmarking of MO-MCS (see Table 5).

Results. The overall computational time required to run the TSA optimization is 5.4 h. The total number of function evaluations required is 10 914. On average, each Pareto point search requires 606 function evaluations. The optimization run leading to the minimum exergy cost configuration is the most challenging and requires 1067 function evaluations, which correspond to six fixed penalty iterations, meaning it is more difficult to find feasible solutions close to the minimum of exergy

cost for the selected initial guess. When considering all the Pareto points, approximately 12% of the objective values are retrieved from the look-up tables. However, no clear trend can be established between the number of function evaluations, the percentage of values retrieved from the look-up tables, and the position of the point along the Pareto curve. For the vast majority of the Pareto points, the resulting number of function evaluations is considerably lower than the maximum allowed, meaning that MO-MCS is able to find feasible optimal solutions within the stopping criterion adopted.

In Figure 8, the Pareto curve resulting from the optimization is represented. Remarkably, all points satisfy purity and recovery

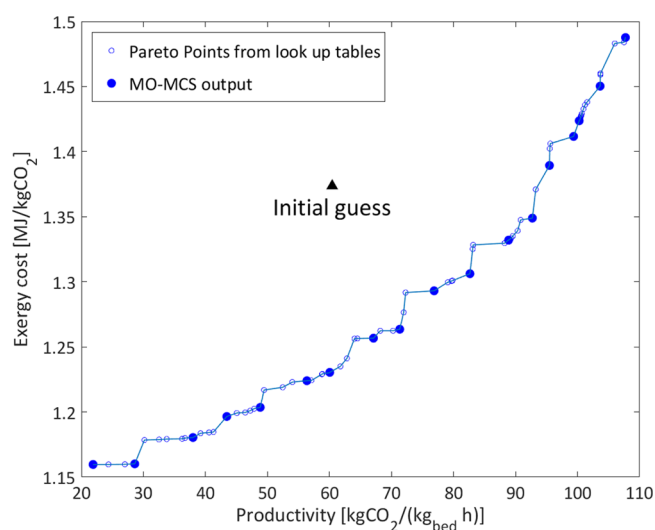


Figure 8. Pareto curve resulting from the MO-MCS run. The filled dots are the points returned directly from the algorithm, while the empty circles are the nondominated points found in the look-up tables after MO-MCS is executed.

constraints. The obtained Pareto curve is smooth. The spacing of the Pareto points is very satisfactory, and a clear trade-off between exergy cost and productivity can be easily identified. The productivity ranges from 23.1 to 106.9 kg_{CO2}/kg_{bed}·h_{cycle}, while the energy cost varies between 1.18 and 1.49 MJ/kg_{CO2}. The objective value is reported also for the selected starting point, which is suboptimal and not close to any Pareto point.

It is worth noting that the black-box model optimization approach allows the evaluation of how the decision variables affect the performance. Indeed, this can be done by considering the value of the decision variables for the Pareto optimal points. Such results are shown in Figure 9. First, the variable trends show that there are multiple, not trivial, combinations of the decision variables that are optimal for the system. These combinations would be hardly identified without a proper optimizer. While some variables show a clear trend moving along the Pareto (from minimum to maximum productivity), others do not. In the latter cases, it might be possible to identify a range of optimal values. As shown in Figure 9B, the duration of the heating2 step is a maximum in the low exergy consumption region, while it decreases to achieve higher productivity. Indeed, longer heating times allow the bed to reach a higher temperature, hence a larger amount of desorbed CO₂. In parallel, the hot level temperature (Figure 9F) is a maximum in the high productivity region and decreases for low exergy consumptions. This because a higher temperature level implies a higher exergy cost of the thermal

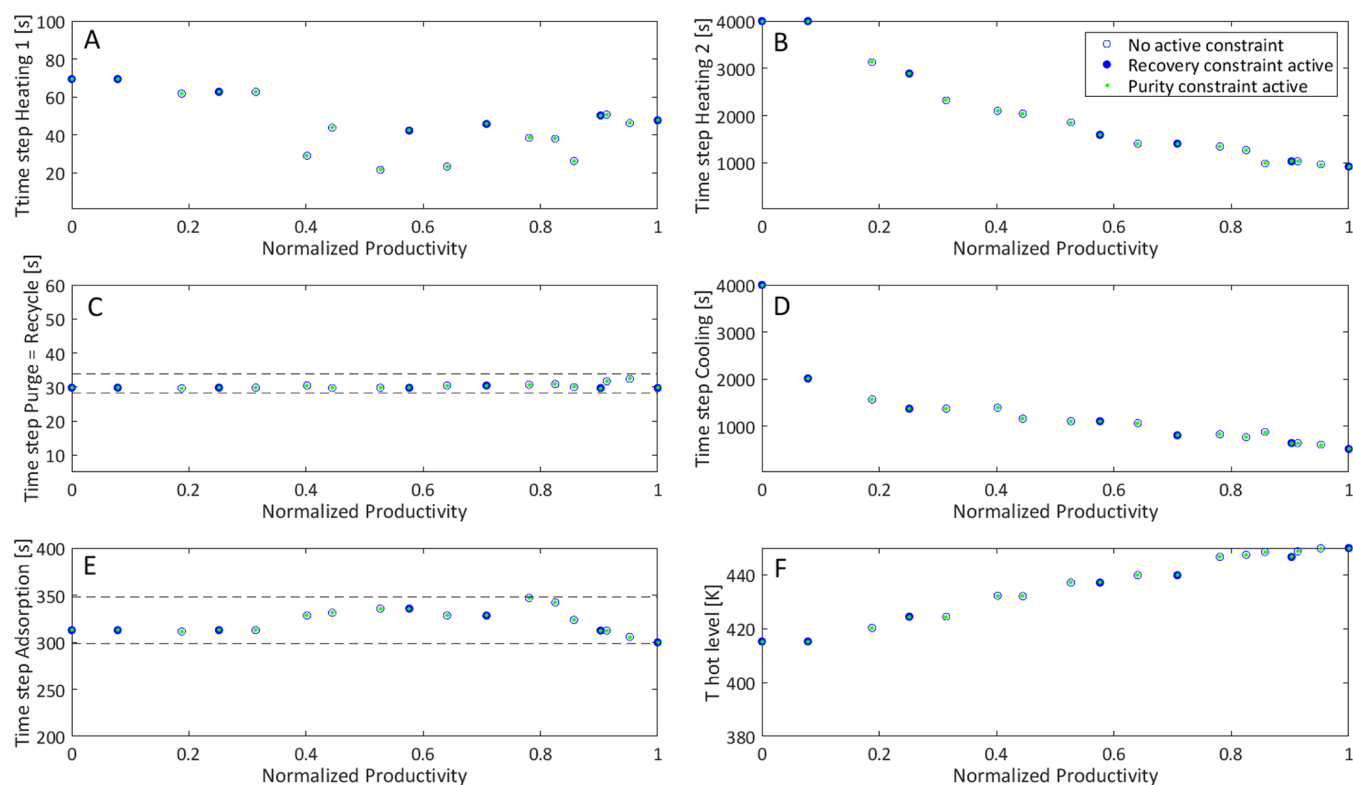


Figure 9. Trends of the decision variables for the TSA case study. The trend of each decision variable is plotted as a function of the normalized productivity. A: Time step heating 1. B: Time step heating. C: Time step purge, equal to time step recycle. D: Time step cooling. E: Time step adsorption. F: Hot temperature level. The empty circles represent optimal solutions where no constraint is active, while filled circles represent points where the constraint on CO₂ recovery is active. Green smaller circles represent points where the constraint on CO₂ purity is active.

power required during heating steps while leading to faster heating. Although no clear trend emerges for the duration of the adsorption and the recycle (purge) step (Figure 9E), the optimizer identifies a narrow optimal range (indicated by dashed lines in Figure 10). The trends for the feed and N₂ purge flow rate are not reported in Figure 9 since they are constant for all optimal points, hitting the lower and upper bounds of the considered optimization range, respectively. Additional information about the influence of the constraints on the Pareto curve are reported with a color code in Figure 9. The constraint on the purity of the CO₂ stream produced is active along the whole Pareto curve, meaning that it is limiting the performance of the considered adsorption cycle. On the other hand, the constraint on the CO₂ recovery is activated without any clear trend. It is worth noting that the optimizer requires more function evaluations for those points where the constraint on the CO₂ recovery is active.

Scaling of Parallel Computing Performance. In order to understand the effect of the number of workers used to run MO-MCS on the performance of the algorithm itself, a sensitivity analysis is performed: the same TSA case study is solved employing 2, 4, 8, and 16 workers. The number of Pareto points is kept fixed and equal to 18, so the number of worker turns is adjusted consequently. The starting point and other algorithm parameters are not varied with respect to the previously presented TSA case study. The main results of the sensitivity analysis are reported in Table 6. While the overall number of function evaluation is almost constant, it is possible to identify a clear trend for the computational time, which decreases when increasing the number of workers. The case where all the internal Pareto points are calculated simultaneously (case using 16 workers) leads to the minimum computational time. However,

Table 6. Sensitivity Analysis to the Number of Workers Scale-Up

| number of parallel workers | 2 | 4 | 8 | 16 |
|---|--------|--------|--------|--------|
| number of Pareto points | 18 | 18 | 18 | 18 |
| number of worker turns | 8 | 4 | 2 | 1 |
| total number of points sampled | 11 064 | 11 194 | 10 914 | 11 064 |
| fraction of points retrieved from the look-up table | 19% | 21% | 12% | 17% |
| computation time, h | 13.64 | 9.03 | 5.41 | 4.92 |
| computational time reduction | | −34% | −58% | −64% |

the relative advantage of increasing the number of workers is progressively reduced according to the well-known Amdahl's Law.⁵⁸ The reasons limiting the speedup are (i) the calculation of the two anchor points can exploit only two workers (thus its computational time is constant when increasing the number of workers), (ii) the serial code sections cannot be parallelized, (iii) there are communication overheads between processors, and (iv) there is a load imbalance between processors. If more than 16 workers are available, no speedup is expected, and it is advantageous to increase the number of calculated Pareto points.

Concerning the use of a look-up table, between 12 and 21% of the function evaluations are retrieved with a considerable savings of computational time. The minimum use of the look-up table occurs in the run with 8 workers because of the following trade-off:

- For a larger number of workers (e.g., 16), several workers start from the same starting point and hence tend to follow the same search path, at least in the initial phase of the optimization. When the same points are sampled by

different workers (not simultaneously), the look-up table is efficiently used.

- For a smaller number of workers (e.g., 2, 4), the successive worker turns benefit from the points sampled in the previous worker turns.

6.2. PSA Cycle. The PSA cycle was synthesized specifically for this study, with the aim of providing both the light and heavy product (i.e., H₂ and CO₂, respectively) at the purity of interest. Accordingly, the complexity of the cycle increases: following the adsorption step and three pressure equalizations, which allow an increase in the recovery, two blowdown steps (BD) are adopted to regenerate the sorbent. The first blowdown provides a waste product that contains most of the impurities and a remarkable amount of CO₂. The second blowdown provides the CO₂ product. The back-pressure regulator is fixed at 1.5 bar for the first blowdown step and 1 bar for the second blowdown step. The actual pressure reached inside the reactors may be higher and is determined by the duration of the respective blowdown step and is hence an output of the model. The resulting cycle is shown in Figure 10. The increased complexity of the adsorption process,

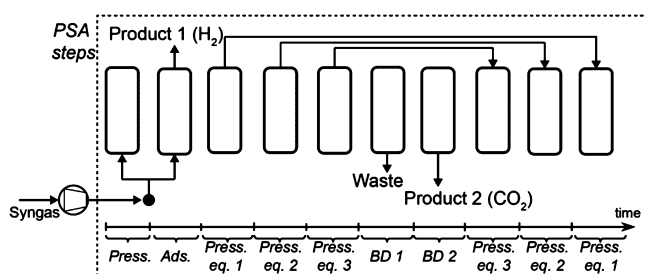


Figure 10. Schematic representation of the PSA cycle considered in this work.

along with the rapid dynamics of pressure swing cycles, make the optimization particularly challenging, yet also particularly suitable for testing the algorithm. In this case, the decision variables, which are reported in Table 7, are the time steps of

Table 7. Decision Variables for Optimization of the TSA Cycle

| decision variables | upper – lower bounds values, s | | initial guess |
|---|-----------------------------------|------------|------------------|
| time steps for: | | | |
| 1. adsorption, t_{ads} [s] | 1. 5 | 1. 100 | 1. 30 |
| 2. blowdown 1, t_{BD1} [s] | 2. 1 | 2. 30 | 2. 5 |
| 3. blowdown 2, t_{BD2} [s] | 3. 1 | 3. 30 | 3. 20 |
| 4. pressure of the inlet syngas, p_{in} [bar] | 4. 34.1 | 4. 50 | 4. 34.1 |
| 5. flow rate of inlet syngas, m_{SG} [m ³ /s] | 5. 1.4e-05 | 5. 2.6e-05 | 5. 2e-05 |

adsorption, BD1 and BD2, the pressure levels of the cycle, i.e., the inlet syngas pressure and the final pressure after BD2, and the flow rate of the feed gas.

For the PSA process, the sorbent is a commercial activated carbon, which is particularly suitable for H₂–CO₂ separation. The adsorption related parameters used are those reported in Marx et al.⁴¹

The feed syngas has the following molar composition: CO₂, 38.3%; H₂, 53.8%; N₂, 7.4%; Ar, 0.5%.

In order to determine the energy performance of the adsorption cycle, we consider that the feed syngas is available at 34.1 bar, in line with existing reforming processes, and that

only further compression represents an energetic cost for the process. The separated CO₂ stream is designated to storage, and therefore compressed to 120 bar. The energetic cost for CO₂ compression is included, since it is affected by the variation of the blowdown pressure and duration, which determine the CO₂ product. The energy consumption of the CO₂ compressor is calculated with commercial software Aspen Plus, considering a state of the art compressor configuration.

Problem Formulation. The objectives considered are the specific energy consumption and the productivity. The energy consumption, as electricity needed, is per mass of produced CO₂, as the H₂ separation could be carried out with no energy needs when no CO₂ recovery is considered (the syngas is pressurized before the PSA). In terms of separation performance, the PSA optimization is constrained to achieve a CO₂ purity ($\phi_{\text{CO}_2}^{\text{PSA}}$) of 85%, a CO₂ recovery ($r_{\text{CO}_2}^{\text{PSA}}$) of 80%, and a H₂ purity ($\phi_{\text{H}_2}^{\text{PSA}}$) of 95%. The resulting optimization problem is formulated as follows in eq 24:

$$\begin{aligned}
 & \begin{cases} \min en^{\text{PSA}}(\underline{x}) \\ \max Pr^{\text{PSA}}(\underline{x}) \end{cases} \\
 & \text{subject to:} \\
 & r_{\text{CO}_2} \geq r_{\text{CO}_2, \min}^{\text{PSA}} \\
 & \phi_{\text{CO}_2} \geq \phi_{\text{CO}_2, \min}^{\text{PSA}} \\
 & \phi_{\text{H}_2} \geq \phi_{\text{H}_2, \min}^{\text{PSA}} \\
 & \underline{lb} \leq \underline{x} \leq \underline{ub} \\
 & \underline{x} \in R^n
 \end{aligned} \quad (24)$$

Slightly different parameters are adopted with respect to the TSA case. As reported in Table 8, a lower number of overall

Table 8. MO-MCS Parameters for the PSA Study Case

| parameter | adopted value |
|--|---------------|
| number of workers | 8 |
| number of worker turns | 2 |
| number of Pareto points | 18 |
| maximum number of function evaluation for a fixed penalty parameter ($N_{\text{max},k}$) | 100 |
| maximum number of function evaluation for a single Pareto point search ($N_{\text{max},\text{tot}}$) | 500 |

function evaluations is allowed, because of the lower number of decision variables (4 vs 8 for the TSA case study), and in order to limit the computational time required to solve the optimization problem, since the convergence time for the PSA model is considerably longer than for the TSA. Moreover, the maximum number of function evaluations for a single Pareto point is also decreased, because a higher number of fixed penalty runs is suggested when dealing with a heavily constrained problem as in the case of the PSA case study, which features three process constraints.

Results. The overall computational time required to run the PSA optimization is 24.6 h. The overall number of function evaluations required is 7785. On average, each Pareto point search requires 432 function evaluations. The optimization runs leading to points in the high productivity region appear to be

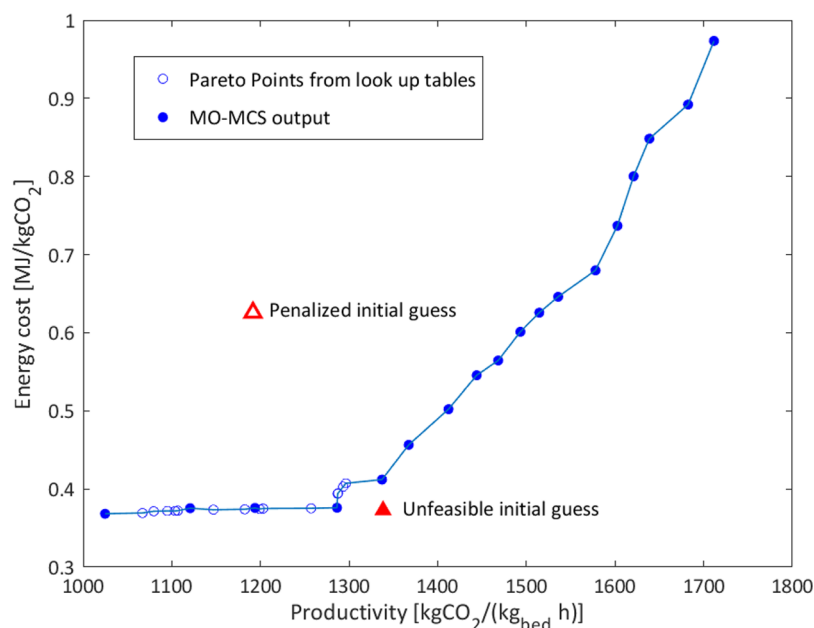


Figure 11. Pareto curve resulting from the MO-MCS run. The filled dots are the point returned directly from the algorithm, while the empty circles are the nondominated points found in the look-up tables after MO-MCS is executed. Note that the selected initial guess (filled triangle) does not meet the constraints of the problem. Hence, it is not a feasible solution, and the associated values of the decision variables are far from that of the Pareto optimal ones. The objective function values seen by the optimizer (with the first guess penalty factor) are shown with an empty triangle and are considerably distant from the Pareto curve.

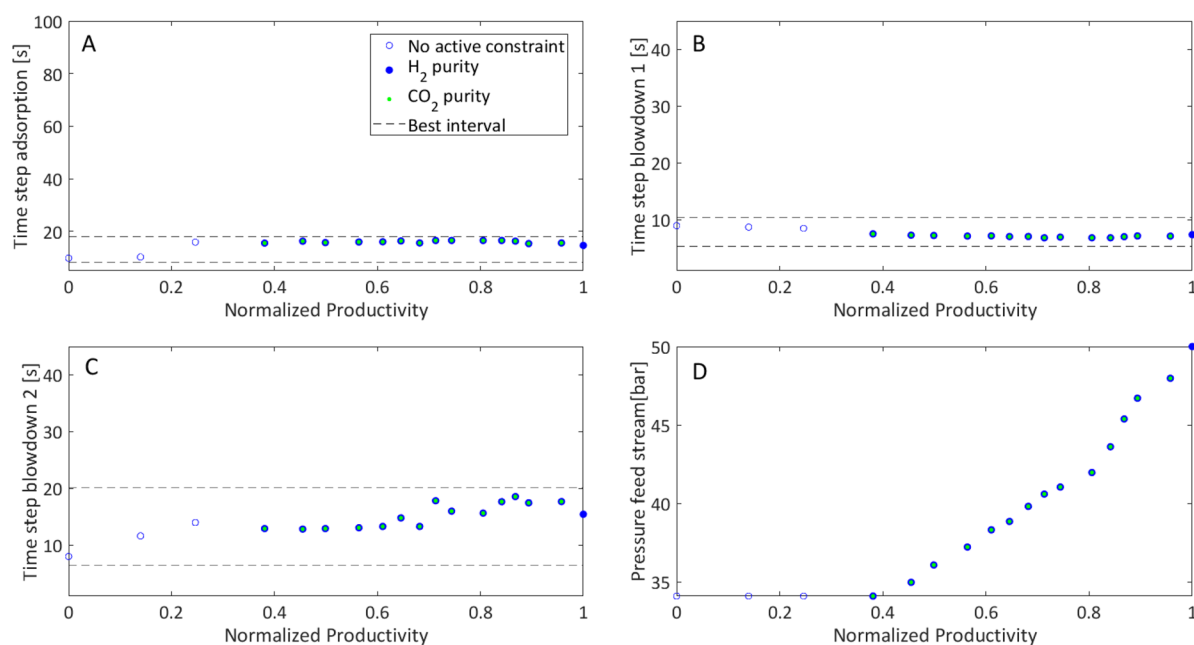


Figure 12. Trends of the decision variables for the PSA case study. For each decision variable, the trend is plotted against normalized productivity. A: Time step adsorption. B: Time step blow down 1. C: Time step blow down 2. D: pressure of the feed stream. The empty circles represent optimal solutions where no constraint is active, while filled circles represent points where the constraint on H_2 purity is active. Smaller green circles represent points where the constraint on CO_2 purity is active. The constraint on CO_2 productivity is never active.

more challenging, resulting in a higher number of function evaluations. This is confirmed by the lack of Pareto points retrieved from the look-up tables in the region close to the maximum productivity (see Figure 11), which might be explained by the additional active constraint in that region of the Pareto curve. The main contribution to the increased complexity of the PSA case study with respect to the TSA is given by the larger number of constraints and the fast change of the

adsorption fronts when pressure is increasing or decreasing. Moreover, the use of look-up tables is limited (7% of total function evaluations are retrieved from the look-up table, compared to the 12% for the TSA case study): this because the objective functions are more sensible toward the penalty factor (i.e., the shape of the objective functions changes more significantly when increasing the penalty factors), so the algorithm tends to sample different points in successive fixed

penalty runs. The sampled points which fail to converge are assigned a suitable suboptimal value, but they are not written down into the look-up tables.

The Pareto curve obtained for the PSA case study exhibits a clear trade-off between energy cost and productivity, as shown in Figure 11. All the points satisfy purity and recovery constraints. The productivity ranges from 1025 to 1712 kg_{CO₂}/kg_{bed}·h_{cycle}, while the energy cost varies between 0.368 and 0.973 MJ/kg_{CO₂}. It is worth noting that the energy consumption of the PSA system includes the energy cost of the compression of the CO₂ produced stream. The selected starting point is close to the Pareto curve. However, it does not meet the constraints of the problem; hence, it cannot be regarded as the optimal solution. The resulting spacing of Pareto points is satisfactory, and interestingly, a discontinuity in the slope of the Pareto curve is identified: in the low productivity region, the productivity can be enhanced with a reduced increase of energy consumption, while after the discontinuity the curve is steeper, meaning that further increasing the productivity becomes more energy expensive. This is because of the inlet stream compressor, whose degree of utilization is decided by the optimization: for low productivity and energy consumption, the syngas feed pressure is kept at 34.1 bar, with no consumptions for the PSA section but only for the CO₂ compression. For high productivity, the optimizer tends to increase the syngas pressure. In fact, this is equivalent to a superstructure-like approach: if the pressure of the syngas is kept to its lower bound, no syngas compressor is adopted, while it is when higher syngas pressures are optimal. Accordingly, the non-smoothness found for productivity close to 1300 kg_{CO₂}/kg_{bed}·h_{cycle} corresponds to switching on/off the syngas compressor. This is confirmed also in Figure 12D, where the trend of the feed stream pressure is outlined: low pressure can be found in the low productivity region, high pressure in the high productivity region.

The trends of the decision variables are represented in Figure 12. As mentioned before, the trend for the pressure of the feed stream (Figure 12D) retraces the behavior of the Pareto curve. On the other hand, it is difficult to appreciate clear trends for the other decision variables; however, MO-MCS identifies a narrow optimal range within the considered decision variables space and adjusts the other decision variables according to the predominant effect of the feed pressure. Such optimal ranges are highlighted with dashed lines in Figure 12A, B, and C. Moreover, we can observe that the constraint on both H₂ and CO₂ purity becomes active when the feed compressor is used, while it is not active in the low productivity region. It is worth noting that the constraint on CO₂ recovery is never active, meaning that it is dominated by other constraints (i.e., a solution which meets H₂ and CO₂ purity constraints also meets the specification on CO₂ recovery). The trend for the syngas flow rate is not reported in Figure 12, since it is constant at equal to the upper bound assumed for the optimization. Such an upper bound is not further extended in order to stay in the validated range of conditions for the adsorption model. The resulting H₂ recovery is a minimum for the minimum energy consumption configuration (61.3%), and it increases moving toward higher productivity design up to 80%. The maximum H₂ recovery design corresponds to the maximum productivity one. Overall, the adoption of MO-MCS has allowed achievement of remarkable performance with a rather simple adsorption cycle (in fact, the novel step is the blow down of the first CO₂ product).

7. CONCLUSIONS

This paper proposes a novel algorithm, named MO-MCS, that was specifically devised to tackle the optimization of adsorption processes with a black-box approach. The algorithm is based on MCS, which is a model-based derivative-free algorithm with known convergence results. MCS has been extended to deal with constrained biobjective optimization problems with the normalized normal constraints method, which enables the calculation of Pareto sets that are (i) well-spaced and (ii) nonsmooth and nonconvex. In order to save computationally expensive function evaluations, look-up tables have been used to retrieve the objective values of already sampled decision variable vectors, and the calculation of Pareto points exploits parallel computing on multiple workers.

MO-MCS is benchmarked against three other multiobjective optimization algorithms for a TSA test case. MO-MCS performs better than all such algorithms, returning a Pareto curve which is 6% more extended than the second best algorithm, closer to the best Pareto envelope (84% improvement with respect to MOPSO) and well-spaced Pareto points. The computational time required for a single run of MO-MCS is about 35% longer than the one required for NSGA-II and MOPSO because during the search of the extreme Pareto points MO-MCS does not fully exploit parallel computing. On the other hand, the stochastic nature of NSGAII and MOPSO calls for multiple runs of the problem, which makes MO-MCS more efficient also from a computational point of view.

The application of MO-MCS to relevant TSA and PSA case studies shows that the proposed optimization algorithm can be used as an effective and flexible tool to guide the design of adsorption-based processes.

In fact, MO-MCS can assess the overall potential of an adsorption process in terms of productivity and energy potential, and it is able to return information on the influence of each considered decision variable on the process, thus identifying possible trends as well as its optimal range.

■ ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge on the ACS Publications website at DOI: 10.1021/acs.iecr.8b00207.

The main equations describing the adsorption model adopted in this work (PDF)

■ AUTHOR INFORMATION

Corresponding Author

*E-mail: emanuele.martelli@polimi.it.

ORCID

Federico Capra: 0000-0002-7681-1593

Matteo Gazzani: 0000-0002-1352-4562

Lisa Joss: 0000-0002-1600-0203

Marco Mazzotti: 0000-0002-4948-6705

Emanuele Martelli: 0000-0002-8843-2400

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

Federico Capra acknowledges RSE (Ricerca Sistema Energetico) and LEAP (Laboratorio Energia and Ambiente Piacenza) for supporting his research activity.

NOMENCLATURE

b = Sips constant [$1/\text{bar}$] as a function of temperature
 c = fluid phase concentration, [mol/m^3]
 C_{ads} = heat capacity of the adsorbed phase, [$\text{J}/(\text{K kg})$]
 C_g = heat capacity of the gas, [$\text{J}/(\text{K m}^3)$]
 C_s = heat capacity of the solid, [$\text{J}/(\text{K kg})$]
 d_p = particle diameter for packing material [m]
 e = energy performance index
 en = specific energy cost [MJ/kg]
 ex = specific exergy cost [MJ/kg]
 f = objective function
 g = constraint function
 h = normalized normal constraint
 h_L = heat transfer coefficient (lumping column inside + wall), [$\text{J}/(\text{m}^2 \text{ s K})$]
 l = pareto curve extension
 \underline{l}_b = lower bound vector
 \underline{N} = utopia vector
 N = normal vector
 N_{max_k} = maximum number of function evaluation for a fixed penalty run
 $N_{\text{max}_\text{tot}}$ = maximum number of function evaluation for a single Pareto point search
 P = pressure
 Pr = productivity
 q = solid phase concentration [mol/kg]
 $q_{i,\text{ref}}^\infty$ = Sips isotherm parameter [mol/kg]
 r = recovery
 R = column radius (reactor model)
 \mathcal{R} = universal gas constant
 sp = exponent in Sips isotherm $[-]$
 SW = sweep
 T = temperature
 t = time coordinate (reactor model)
 T_{amb} = ambient temperature [K]
 u = superficial gas velocity, [m/s]
 \underline{u}_b = lower bound vector
 \underline{x} = decision variables vector
 \underline{X} = utopia line points
 y = mole fraction $[-]$
 z = spatial coordinate (reactor model)
 δ = normalized increment
 ΔH = isosteric heat of adsorption [J/mol]
 ε_b = bed void fraction
 ε_t = overall void fraction, $[-]$
 μ = penalty factor
 ρ_b = fluid phase density, [kg/m^3]
 ρ_p = particle density, [kg/m^3]
 σ = spacing constant
 τ = penalty update factor
 ϕ = purity

ACRONYMS

BDF = backward differentiation formula
 CSS = cyclic steady state
 DAE = differential algebraic equations
 DMS = direct multisearch
 LDF = linear driving force
 MCS = multilevel coordinate search
 MO-MCS = multiobjective multilevel coordinate search
 MOPSO = multiobjective particle swarm
 NBI = normal-boundary intersection

NG = natural gas
 NNC = normalized normal constraint
 NSGA II = nonsorting genetic algorithm II
 PDE = partial derivative equations
 PSA = pressure swing adsorption
 TSA = temperature swing adsorption
 VPSA = vacuum pressure swing
 WS = weighted sum

SUBSCRIPTS/SUPERSCRIPTS

A1 = Pareto extreme point 1
 A2 = Pareto extreme point 2
 b = Bulk phase
 C = Constraint
 PP = Pareto points
 s = solid phase
 W = worker
 WT = worker turns

REFERENCES

- (1) D'Alessandro, D. M.; Smit, B.; Long, J. R. Carbon Dioxide Capture: Prospects for New Materials. *Angew. Chem., Int. Ed.* **2010**, *49*, 6058.
- (2) Zanco, S. E.; Joss, L.; Hefti, M.; Gazzani, M.; Mazzotti, M. Addressing the Criticalities for the Deployment of Adsorption-based CO₂ Capture Processes. *Energy Procedia* **2017**, *114*, 2497.
- (3) Webley, P. A. Adsorption technology for CO₂ separation and capture: A perspective. *Adsorption* **2014**, *20*, 225.
- (4) IPCC Climate Change 2013; The Physical Science Basis, Cambridge University Press, 2014.
- (5) Fuss, S.; Canadell, J. G.; Peters, G. P.; Tavoni, M.; Andrew, R. M.; Ciais, P.; Jackson, R. B.; Jones, C. D.; Kraxner, F.; Nakicenovic, N.; Le Quéré, C.; Raupach, M. R.; Sharifi, A.; Smith, P.; Yamagata, Y. Betting on negative emissions. *Nat. Clim. Change* **2014**, *4*, 850.
- (6) Nilchan, S.; Pantelides, C. C. On the Optimisation of Periodic Adsorption Processes. *Adsorption* **1998**, *4*, 113.
- (7) Smith, O. J., IV; Westerberg, a. W. Acceleration of Cyclic Steady State Convergence for Pressure Swing Adsorption Models. *Ind. Eng. Chem. Res.* **1992**, *31*, 1569.
- (8) Ding, Y.; LeVan, M. D. Periodic states of adsorption cycles III. *Chem. Eng. Sci.* **2001**, *56*, 5217.
- (9) First, E. L.; Hasan, M. M. F.; Floudas, C. A. Discovery of novel zeolites for natural gas purification through combined material screening and process optimization. *AIChE J.* **2014**, *60*, 1767.
- (10) Webley, P. A.; He, J. Fast solution-adaptive finite volume method for PSA/VSA cycle simulation; 1 single step simulation. *Comput. Chem. Eng.* **2000**, *23*, 1701.
- (11) Dowling, A. W.; Vetukuri, S. R. R.; Biegler, L. T. Large-Scale Optimization Strategies for Pressure Swing Adsorption Cycle Synthesis. *AIChE J.* **2012**, *58*, 3777.
- (12) Ko, D.; Moon, I. Multiobjective Optimization of Cyclic Adsorption Processes. *Ind. Eng. Chem. Res.* **2002**, *41*, 93.
- (13) Jiang, L.; Biegler, L. T.; Fox, V. G. Simulation and optimization of pressure-swing adsorption systems for air separation. *AIChE J.* **2003**, *49*, 1140.
- (14) Agarwal, A.; Biegler, L. T.; Zitney, S. E. A superstructure-based optimal synthesis of PSA cycles for post-combustion CO₂ capture. *AIChE J.* **2010**, *56*, 1813.
- (15) Effendy, S.; Xu, C.; Farooq, S. Optimization of a Pressure Swing Adsorption Process for Nitrogen Rejection from Natural Gas. *Ind. Eng. Chem. Res.* **2017**, *56*, 5417.
- (16) Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **2002**, *6*, 182.
- (17) Kapoor, A.; Yang, R. T. Optimization of a pressure swing adsorption cycle. *Ind. Eng. Chem. Res.* **1988**, *27*, 204.
- (18) Haghpanah, R.; Majumder, A.; Nilam, R.; Rajendran, A.; Farooq, S.; Karimi, I. A.; Amanullah, M. Multiobjective Optimization of a Four-

Step Adsorption Process for Postcombustion CO₂ Capture Via Finite Volume Simulation. *Ind. Eng. Chem. Res.* **2013**, *52*, 4249.

(19) Krishnamurthy, S.; Haghpahan, R.; Rajendran, A.; Farooq, S. Simulation and Optimization of a Dual-Adsorbent, Two-Bed Vacuum Swing Adsorption Process for CO₂ Capture from Wet Flue Gas. *Ind. Eng. Chem. Res.* **2014**, *53*, 14462.

(20) Beck, J.; Friedrich, D.; Brandani, S.; Fraga, E. S. Multi-objective optimization using surrogate models for the design of VPSA systemsElsevier Ltd. *Comput. Chem. Eng.* **2015**, *82*, 318.

(21) Boukouvala, F.; Floudas, C. A. ARGONAUT: AlgoRithms for Global Optimization of coNstrAined grey-box compUTational problemsSpringer Berlin Heidelberg. *Optim. Lett.* **2017**, *11*, 895.

(22) Boukouvala, F.; Hasan, M. M. F.; Floudas, C. A. Global optimization of general constrained grey-box models: new method and its application to constrained PDEs for pressure swing adsorption-Springer US. *J. Glob. Optim.* **2017**, *67*, 3.

(23) Li, Z.; Floudas, C. A. Optimal scenario reduction framework based on distance of uncertainty distribution and output performance: II. Sequential reductionElsevier Ltd. *Comput. Chem. Eng.* **2016**, *84*, 599.

(24) Misener, R.; Floudas, C. A. ANTIGONE: Algorithms for coNTinuous/Integer Global Optimization of Nonlinear Equations. *J. Glob. Optim.* **2014**, *59*, 503.

(25) Misener, R.; Floudas, C. A. GloMIQO: Global mixed-integer quadratic optimizer. *J. Glob. Optim.* **2013**, *57*, 3.

(26) Jiang, L.; Fox, V. G.; Biegler, L. T. Simulation and optimal design of multiple-bed pressure swing adsorption systems. *AIChE J.* **2004**, *50*, 2904.

(27) Jiang, L.; Biegler, L. T.; Fox, V. G. Design and optimization of pressure swing adsorption systems with parallel implementation. *Comput. Chem. Eng.* **2005**, *29*, 393.

(28) Kim, S.; Ko, D.; Moon, I. Dynamic Optimization of a Dual Pressure Swing Adsorption Process for Natural Gas Purification and Carbon Capture. *Ind. Eng. Chem. Res.* **2016**, *55*, 12444.

(29) Ko, D.; Siriwardane, R.; Biegler, L. T. Optimization of Pressure Swing Adsorption and Fractionated Vacuum Pressure Swing Adsorption Processes for CO₂ Capture. *Ind. Eng. Chem. Res.* **2005**, *44*, 8084.

(30) Ko, D. Optimization of Vacuum Pressure Swing Adsorption Processes To Sequester Carbon Dioxide from Coalbed Methane. *Ind. Eng. Chem. Res.* **2016**, *55*, 8967.

(31) Agarwal, A.; Biegler, L. T.; Zitney, S. E. Simulation and Optimization of Pressure Swing Adsorption Systems Using Reduced-Order Modeling. *Ind. Eng. Chem. Res.* **2009**, *48*, 2327.

(32) Joss, L.; Gazzani, M.; Mazzotti, M. Rational design of temperature swing adsorption cycles for post-combustion CO₂ capture. *Chem. Eng. Sci.* **2017**, *158*, 381.

(33) Huyer, W.; Neumaier, A. Global Optimization by Multilevel Coordinate Search. *J. Glob. Optim.* **1999**, *14*, 331.

(34) Conn, A. R.; Scheinberg, K.; Vicente, L. N. *Introduction to Derivative-Free Optimization*; SIAM: Philadelphia, 2009.

(35) Rios, L. M.; Sahinidis, N. V. Derivative-free optimization: a review of algorithms and comparison of software implementations. *J. Glob. Optim.* **2013**, *56*, 1247.

(36) Todd, R. S.; He, J.; Webley, P. A.; Beh, C.; Wilson, S.; Lloyd, M. A. Fast Finite-Volume Method for PSA/VSA Cycle Simulation s Experimental Validation. *Ind. Eng. Chem. Res.* **2001**, *40*, 3217.

(37) Casas, N.; Schell, J.; Joss, L.; Mazzotti, M. A parametric study of a PSA process for pre-combustion CO₂ captureElsevier B.V. *Sep. Purif. Technol.* **2013**, *104*, 183.

(38) Casas, N.; Schell, J.; Pini, R.; Mazzotti, M. Fixed bed adsorption of CO₂/H₂ mixtures on activated carbon: Experiments and modeling. *Adsorption* **2012**, *18*, 143.

(39) Radhakrishnan, K.; Hindmarsh, A. C. *Description and use of LSODE, the Livemore Solver for Ordinary Differential Equations*; National Aeronautics and Space Administration: Livermore, CA, 1993.

(40) Schell, J.; Casas, N.; Marx, D.; Mazzotti, M. Precombustion CO₂ Capture by Pressure Swing Adsorption (PSA): Comparison of Laboratory PSA Experiments and Simulations. *Ind. Eng. Chem. Res.* **2013**, *52*, 8311.

(41) Marx, D.; Joss, L.; Hefti, M.; Gazzani, M.; Mazzotti, M. CO₂ Capture from a Binary CO₂ /N₂ and a Ternary CO₂ /N₂ /H₂ Mixture by PSA: Experiments and Predictions. *Ind. Eng. Chem. Res.* **2015**, *54*, 6035.

(42) Marx, D.; Joss, L.; Hefti, M.; Mazzotti, M. Temperature Swing Adsorption for Postcombustion CO₂ Capture: Single- and Multi-column Experiments and Simulations. *Ind. Eng. Chem. Res.* **2016**, *55*, 1401.

(43) Hefti, M.; Joss, L.; Bjelobrk, Z.; Mazzotti, M. On the potential of phase-change adsorbents for CO₂ capture by temperature swing adsorptionRoyal Society of Chemistry. *Faraday Discuss.* **2016**, *192*, 153.

(44) Casas, N.; Schell, J.; Blom, R.; Mazzotti, M. MOF and UiO-67/ MCM-41 adsorbents for pre-combustion CO₂ capture by PSA: Breakthrough experiments and process design. *Sep. Purif. Technol.* **2013**, *112*, 34.

(45) Nocedal, J.; Wright, S. J. *Numerical Optimization*; Springer: New York, 2006.

(46) Andersson, J. A *Survey of Multiobjective Optimization in Engineering Design*; Linköping University, 2000.

(47) Das, I.; Dennis, J. E. A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems. *Struct. Optim.* **1997**, *14*, 63.

(48) Das, I.; Dennis, J. E. Normal-Boundary Intersection: A New Method for Generating the Pareto Surface in Nonlinear Multicriteria Optimization Problems. *SIAM J. Optim.* **1998**, *8*, 631.

(49) Kim, I. Y.; de Weck, O. L. Adaptive weighted-sum method for bi-objective optimization: Pareto front generation. *Struct. Multidiscip. Optim.* **2005**, *29*, 149.

(50) Messac, A.; Ismail-Yahaya, A.; Mattson, C. a. The normalized normal constraint method for generating the Pareto frontier. *Struct. Multidiscip. Optim.* **2003**, *25*, 86.

(51) Joss, L.; Capra, F.; Gazzani, M.; Mazzotti, M.; Martelli, E. *Comput.-Aided Chem. Eng.* **2016**, *38*, 1467.

(52) Coello Coello, C. A.; Lechuga, M. S. *Proceedings of the 2002 Congress on Evolutionary Computation*; 2002 World Congress on Computational Intelligence - WCCI '02, Honolulu, HI, 2002; p 1051.

(53) Custódio, a. L.; Madeira, J. F. a.; Vaz, a. I. F.; Vicente, L. N. Direct Multisearch for Multiobjective Optimization. *SIAM J. Optim.* **2011**, *21*, 1109.

(54) Zitzler, E.; Deb, K.; Thiele, L. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evol. Comput.* **2000**, *8*, 173.

(55) Clause, M.; Merel, J.; Meunier, F. Numerical parametric study on CO₂ capture by indirect thermal swing adsorptionElsevier Ltd. *Int. J. Greenhouse Gas Control* **2011**, *5*, 1206.

(56) Ntiemoah, A.; Ling, J.; Xiao, P.; Webley, P. A.; Zhai, Y. CO₂ Capture by Temperature Swing Adsorption: Use of Hot CO₂ -Rich Gas for Regeneration. *Ind. Eng. Chem. Res.* **2016**, *55*, 703.

(57) Mérel, J.; Clause, M.; Meunier, F. Carbon dioxide capture by indirect thermal swing adsorption using 13X zeolite. *Environ. Prog.* **2006**, *25*, 327.

(58) Amdahl, G. *AFIPS Conference Proceedings*; ACM Press, 1967; Vol. 30, p 483.