Available online at www.sciencedirect.com
SCIENCE DIRECT®
Computers and Chemical Engineering 29 (2005) 393–399
Computers & Chemical Engineering
www.elsevier.com/locate/compchemeng

Short communication

Design and optimization of pressure swing adsorption systems with parallel implementation

Ling Jiang, Lorenz T. Biegler*, V. Grant Fox

Department of Chemical Engineering, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, USA

Received 13 April 2004; received in revised form 9 August 2004; accepted 17 August 2004
Available online 3 October 2004

Abstract

Over the past three decades, pressure swing adsorption (PSA) processes have gained increasing commercial acceptance as an energy efficient separation technique. These processes are distributed in nature, with spatial and temporal variations and are mathematically represented by partial differential equations (PDEs). After a start-up time, the system reaches cyclic steady state (CSS), at which the conditions in each bed at the start and end of each cycle are identical, revealing normal production. We implement a Newton-based approach with accurate sensitivities to directly determine cyclic steady states with design constraints. We also design optimal PSA processes by means of state-of-the-art SQP-based optimization algorithms. The simultaneous tailored approach can incorporate large-scale and detailed adsorption models and is more robust and efficient than competing optimization methodologies. In order to improve the computational efficiency, we parallelize sensitivity calculation and achieve a close-to-linear speed up rate. Applications of several non-isothermal industrial O_2 VSA and H_2 PSA processes are presented.

PACS classification: 02.60.Pn; 02.60.Lj; 82.65.Yh

Keywords: Pressure swing adsorption; Optimization; Parallel computing; Sensitivity calculations

1. Introduction

With extensive industry applications of pressure swing adsorption (PSA), there is significant interest for efficient modeling, simulation and optimization strategies. However, the design and optimization of PSA systems still largely remain an experimental effort (Sircar, 2002). This is mainly because most practical PSA processes are fairly complex and are usually expensive and time-consuming to solve with the accuracy and reliability needed for industrial design. For example, the traditional way to determine a cyclic steady state (CSS) is to simulate a series of complete cycles until the bed conditions repeat periodically. This *successive substitution* method mimics the true operation of a real plant but usually takes hundreds or even thousands of cycles to converge. To design and optimize PSA, a common practice is to develop a simplified model for one specific process and fine-tune the model using experiments and pilot plant data. Although such models are often useful, the case-by-case studies are hard to transfer among different PSA systems. Recently, more sophisticated optimization strategies have been applied to PSA systems with significant improvements in cycle performance. A review of these approaches can be found in Biegler, Jiang and Fox (in press). Here we develop a flexible and reliable optimization strategy that incorporates general process models and rigorous solution procedures within a parallel computing framework. This paper is organized as follows. The next section outlines the solution strategies, including PDE discretization, CSS convergence acceleration, sensitivity evaluation and optimization. Section 3 discusses the parallelization algorithm. Section 4 presents four PSA processes as case studies and computational results are shown. Section 5 states the conclusions and future work.

* Corresponding author. Tel.: +1 412 268 2232; fax: +1 412 268 7139.
E-mail address: lb01@andrew.cmu.edu (L.T. Biegler).

0098-1354/$ – see front matter © 2004 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compchemeng.2004.08.014

2. Solution strategy

This section provides a concise overview of the numerical tools we develop in this research. More details can be found in Jiang, Biegler and Fox (2003) and Jiang, Fox and Biegler (in press).

2.1. PDE discretization

As described in Jiang et al. (2003), bed models consist of mass and energy balances and constitutive equations that are represented by hyperbolic partial differential equations (PDEs) that lead to sharp adsorption fronts. To solve these PDEs, we apply the *method of lines* (MOL). The PDEs are first discretized in space which results in a system of ordinary differential equations (ODEs) or differential algebraic equations (DAEs), which are then integrated over time by standard routines. The advantage of MOL is that since space and time discretizations are decoupled, high order accuracy can be achieved in each dimension. In Jiang et al. (2003), we apply a finite volume method for spatial discretization, in order to preserve the mass and energy conservation laws in the spatial direction. To resolve the sharp adsorption fronts that arise from rapid gas–solid mass transfer, we adopt the second-order Van Leer flux limiter to mitigate numerical error and avoid physically unrealistic oscillations near the adsorption fronts. Modifications are made to force the flux limiter to have continuous first derivatives everywhere.

2.2. DAE solver and sensitivity evaluation

After converting PDEs to DAEs/ODEs, we use DASPK 3.0 (Li & Petzold, 1999; Li et al., 2000) to integrate the system over time. DASPK solves initial value problems of DAEs/ODEs using backward differentiation formulae (BDF) and is well suited for stiff systems. For a general DAE system, $F(t, y, y', p) = 0$, $y(0) = y_0(p)$, where $y(t)$ are the differential–algebraic state variables and t the independent variable, time, DASPK solves this by a modified version of Newton's method. DASPK also automates a sensitivity analysis. The original DAE/ODEs are differentiated with respect to the sensitivity parameters, p, yielding additional $N_p \times N_y$ sensitivity equations. The latter are integrated together with the original DAEs:

$$G = \begin{cases} F(t, y, y', p) = 0, & y(0) = y_0(p), \\ \dfrac{\partial F}{\partial y} s_i + \dfrac{\partial F}{\partial y'} s_i' + \dfrac{\partial F}{\partial p_i} = 0, & s_i(0) = \dfrac{\partial y_0}{\partial p_i},\ i = 1, \ldots, N_p, \end{cases}$$

$$\text{where } s_i = \frac{\partial y}{\partial p_i} \tag{1}$$

Defining the variables $Y^{\mathrm{T}} = [y^{\mathrm{T}} s_1^{\mathrm{T}} s_2^{\mathrm{T}} \cdots s_{Np}^{\mathrm{T}}]$, the enhanced system can be rewritten as $G(t, Y, Y', p)=0$ and is solved at each time step by a Newton iteration method, $Y^{k+1} = Y^k - J_G^{-1} G(Y^k)$, where J_G represents the Jacobian of G. The sensitivities s_i are used in design and optimization. This is called the direct sensitivity approach. If J_G and the sensitivity equations are evaluated using automatic differentiation (Bischof, Carle, Khademi, & Mauer, 1992), the DAE integration is more robust and produces more accurate sensitivities. However, the disadvantage of the direct sensitivity approach is that the computational cost increases with the number of parameters N_p and can be quite expensive when N_p is large. In Section 3, we parallelize the sensitivity evaluation, thus enhancing the efficiency for design and optimization.

2.3. CSS convergence acceleration

To accelerate the convergence of CSS for an N-step PSA process, Croft and LeVan (1994) and Smith and Westerberg (1992) propose the direct determination approach. Here we define the vector of parameters as $p^{\mathrm{T}} = [y_0^{\mathrm{T}} q^{\mathrm{T}}]$ and write the CSS condition as a boundary value problem:

$$\begin{aligned} &F_1(y_1, y_1', q, t) = 0, \quad y_1(0) = y_0, \quad 0 \le t \le t_1, \\ &F_i(y_i, y_1', q, t) = 0, \quad y_i(t_{i-1}) = y_{i-1}(t_{i-1}), \quad t_{i-1} \le t \le t_i, \\ &i = 2, \ldots, N, \qquad C_k = y_0 - y_N(t_N) = 0 \end{aligned} \tag{2}$$

This is solved using a shooting method with the DAE bed models solved implicitly and Newton-based methods used to directly determine the initial bed conditions y_0. When design constraints (W) such as purity and pressure are imposed, we define at iteration k an augmented error vector and augmented Jacobian as:

$$e_k = \begin{bmatrix} C_k \\ W_k \end{bmatrix} = \begin{bmatrix} y_N(t_N) - y_{0,k} \\ W_k \end{bmatrix} = 0,$$

$$J_k = \frac{\partial e_k}{\partial (y_{0,k},\ q_k)} \tag{3}$$

The vector q represents the manipulated variables for the design constraints. At each new iteration, the new variables (y_{0k+1}, q_{k+1}) are determined by $\begin{bmatrix} (y_{0,k+1})^{\mathrm{T}} & (q_{k+1})^{\mathrm{T}} \end{bmatrix} = \begin{bmatrix} (y_{0,k})^{\mathrm{T}} & (q_k)^{\mathrm{T}} \end{bmatrix} - (J_k)^{-1} e_k$. This process repeats until the CSS condition is satisfied. Newton methods can achieve a quadratic convergence rate near the solution. The Jacobian in (3) comes from the sensitivity Eq. (2). The high cost of obtaining the Jacobians has led Ding and LeVan (2001), Kvamsdal and Hertzberg (1997) and Smith and Westerberg (1992) to consider using quasi-Newton updates to substitute for new Jacobians. In addition, the high nonlinearity and ill-conditioning in realistic PSA systems can often lead to the failure of Newton and Broyden solvers. To ensure robust convergence, we use a trust region method with scaling to achieve good convergence for several industrial O_2 VSA and H_2 PSA processes (Jiang et al., 2003, in press). The trust region method combines Newton (or Broyden) and Cauchy steps. The search direction is controlled by the size of the trust region, which is determined by monitoring the convergence progress. With proper scaling, trust region methods are

well suited for ill-conditioned systems of equations (Biegler, Grossmann, & Westerberg, 1997).

2.4. PSA optimization

Optimization problems for design of PSA systems can be expressed as:

$$\text{Min}\,\phi(y, y_0, q) \quad \text{s.t.} \quad F(y, y', q, t) = 0,\ W(y(t, y_0, q)) \le 0,$$
$$C(y_0) = y_0 - y_N(t_N) = 0,\ \text{LB} \le (y_0, q) \le \text{UB} \qquad (4)$$

Here $F(y, y', q, t)^{\mathrm{T}} = [F_1(y_1, y_1', q, t)^{\mathrm{T}}\ F_2(y_2, y_2', q, t)^{\mathrm{T}} \cdots F_N(y_N, y_N', q, t)^{\mathrm{T}}]$ is the collection of bed models, discretized in space, $y^{\mathrm{T}} = [y_1^{\mathrm{T}} y_2^{\mathrm{T}} \cdots y_N^{\mathrm{T}}]$ are the state variables, y_0 initial conditions for the state variables, q are decision variables and are subject to the lower bounds (LB) and upper bounds (UB), W are design constraints which can include purity, pressure or production rate requirements and C are the CSS conditions. Candidates for the decision variables q can be geometric parameters such as bed length, diameter and adsorbent packing or process parameters such as flow rates, step times and operating pressures. Also, ϕ is the objective function which, for example, can maximize overall recovery or minimize operating cost at desired purity. As described in Biegler et al. (in press), we apply an efficient simultaneous tailored framework to solve (4). Here, convergence of CSS is incorporated as a constraint in the optimization problem while the DAEs are solved in an inner loop. The detailed bed model is solved at every optimization iteration, in order to evaluate objective and constraint functions and their sensitivities. The CSS is not converged until the optimal solution is reached, thus the time-consuming CSS direct substitution loop is eliminated. The optimization algorithm in the simultaneous tailored approach is reduced space Successive Quadratic Programming (rSQP). rSQP exploits this problem structure and is well suited to optimize large nonlinear programming systems with relatively few decision variables. More details on the rSQP algorithm can be found in Ternet and Biegler (1998). Extensions of rSQP to deal with ill-conditioning are described in Jiang et al. (2003).

3. Parallelization with message passing interface (MPI)

3.1. Algorithms

For our optimization algorithm, the sensitivity calculation is the most time-consuming step and remains a bottleneck for design and optimization. However since the sensitivity calculation with respect to each parameter is independent, parallelization is straightforward. Zhu and Petzold (1999) compare several parallel sensitivity analysis schemes for DAEs and find the distributed parameter only (DPO) approach is the most efficient. Here, the sensitivity parameters are divided into different sets and are distributed to different processors, with each processor running a copy of the state equations and computing a subset of the sensitivity variables. Although the computation of DAE models is repeated at each processor, this overhead work is rather small compared to the effort of calculating sensitivities. We implement a master–slave paradigm. The master processor has the maximum control over the process and deals with the optimizer directly. When the master processor decides to do sensitivity calculation, it broadcasts all the necessary information to the slave processors and gathers the sensitivities from slaves upon their completion. The slaves are mostly working independently, and have no control over the computation process. Because the communication and synchronization costs among slave processors are very low, nearly linear speedup is possible to achieve.

3.2. Message passing model

The message-passing model posits a set of processes that have only local memory but are able to communicate with other processes by sending and receiving messages. The data transfer, from local memory of one process to the local memory of another, requires operations to be performed by both processes. The message passing model has the advantages of universality, expressivity, ease of debugging and high performance (Gropp, Lusk, & Skjellum, 1999). The message passing interface (MPI) addresses the message-passing model with a collection of processes communicating with messages. The structural diagram with MPI implementation is shown in Fig. 1.

3.3. Computing facility

The parallel computing work is performed on the Beowulf computer cluster, a “Beowulf” class distributed parallel computer built and maintained by the Department of Chemical Engineering at Carnegie Mellon University (http://beowulf.cheme.cmu.edu/). Beowulf has 3 servers and 41 computing nodes. Most nodes have dual 1 GHz Pentium III processors with between 0.5 and 2 GB RAM. Nodes are arranged in 3 rack cabinets, with 100 Mbps interconnects and 1 Gbps uplinks to the file server.

4. Case studies

Applications of several O_2 VSA and H_2 PSA industrial cycles are employed for illustration. More details can be found in Jiang et al. (2003, in press).

4.1. System 1

As seen in Fig. 2, this is a single-bed 3-step non-isothermal O_2 VSA cycle consisting of Make Product, Evacuation and Repressurization steps. This process separates oxygen from air using active zeolite. For design purposes, we choose the

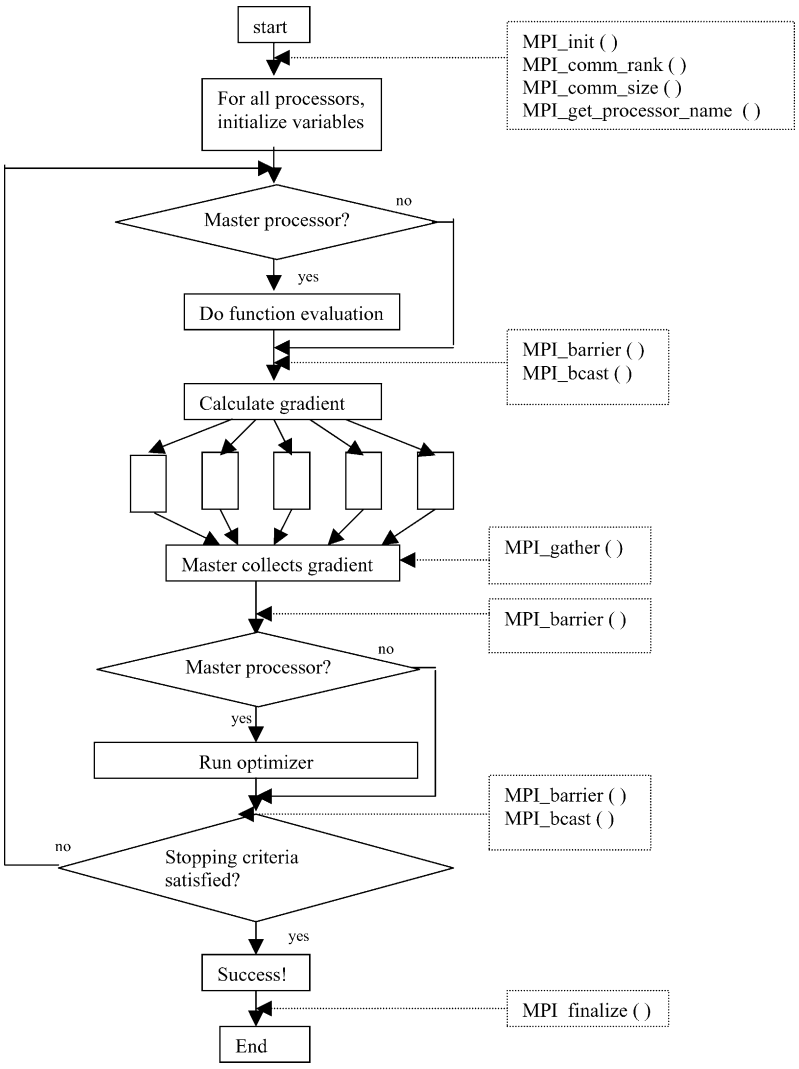

Fig. 1. MPI implementation diagram for optimization.

flow rates of feed, evacuation and purge (F_1, F_2, F_3) and the valve constant (CV) to be the manipulated variables. The design targets are end-of-step pressures P_1, P_2, P_3 and 35% O_2 product purity. For optimization, we maximize O_2 recovery at desired purity (35%) at cyclic steady state. We choose product tank pressure, valve constant, and step times (t_2, t_3) as decision variables. The optimal condition achieves 27% more recovery than design condition, by withdrawing product and repressurizing at a higher level.

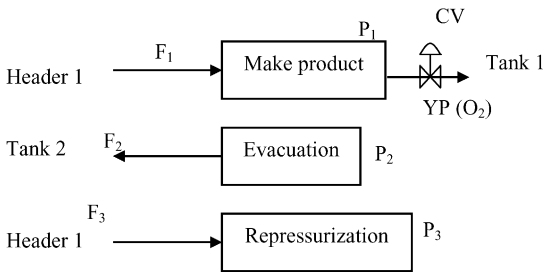

Fig. 2. Single-bed, three-step PSA cycle.

4.2. System 2

This is a single-bed 6-step industrial O_2 VSA process (Fig. 3). The adsorption bed is continuously packed with zeolite. The design targets are end-of-step pressures P_1, P_3, P_4, P_6 and 95% O_2 purity in the product tank. The manipulated

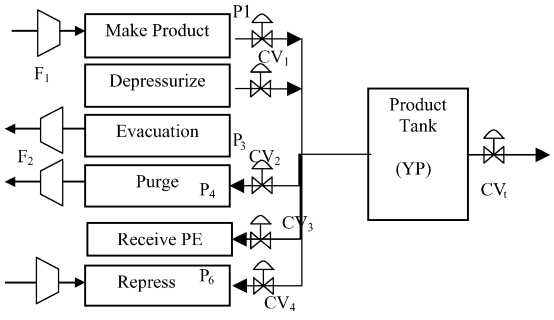

Fig. 3. Single-bed, six-step O2 VSA cycle.

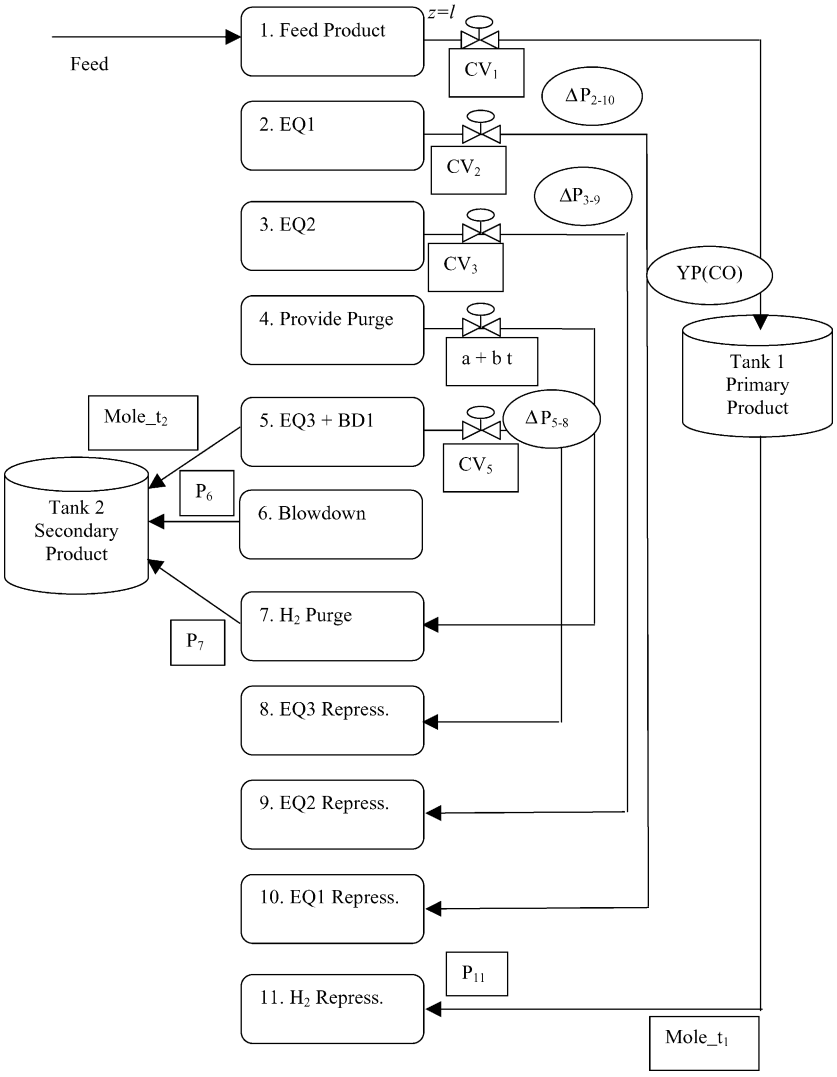

Fig. 4. Flow sheet of five-bed H_2 PSA system.

variables are feed flow rate (F_1), evacuation flow rate (F_2) and valve constants (CV_2, CV_4, CV_t). For optimization, we minimize the specific work usage at 95% O_2 purity at cyclic steady state. Additional constraints on the step pressures are $P_1 \geq 1.5$ atm and $P_4 = P_3 + 0.1$ atm. Feed flow rate (F_1), evacuation flow rate (F_2), step times t_1 and t_4, valve constants (CV_2, CV_4, CV_t) are decision variables. Compared with design conditions, we obtain an 8% energy saving. The energy saving is accomplished by lowering F_1, maintaining P_1 at its lower bound and reducing t_4 when product is recovered.

4.3. System 3

This system is very similar to system 2, except that the adsorption bed is packed with two layers of different adsorbents. The adsorbent in the first one-third of the bed is inert while the adsorbent in the next two-thirds is active zeolite. The inert adsorbent represents adsorbent poisoned by water and introduces an extreme nonlinearity in the temperature profile where the two adsorbents meet. The same design variables and constraints are used. In order to achieve the same

Table 1
Wall clock time comparison with single processor (in CPU hours)

	CSS convergence		Design, direct determination	Optimization	
	Successive substitution	Direct determination		Black Box	Simultaneous tailored
System 1	0.56	0.62	0.62	56	40.77
System 2	4.55	2.91	2.91	455	38.03
System 3	6.91	6.04	6.04	691	68.65
System 4	6.67	Not tested	Not tested	Failed to converge	380

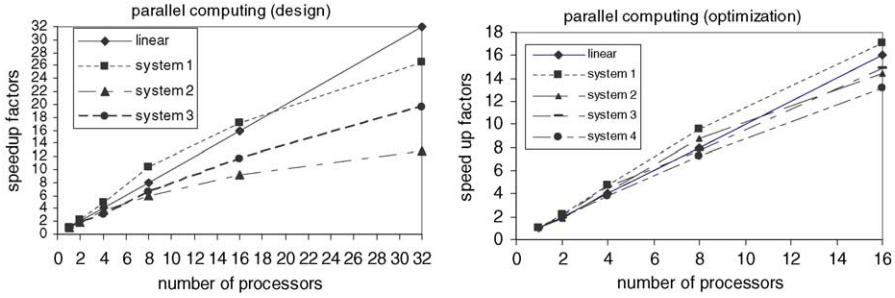

Fig. 5. Parallel computing results: (a) design; (b) optimization.

purity as system 2, we decrease F_1, increase F_2, and set the valve CVs smaller. To formulate an optimization problem, we substitute valve CV_1 for feed flow rate F_1. Here a 13.4% energy saving is gained over the design conditions.

4.4. System 4

This is a five-bed hydrocarbon separation process (Fig. 4). High-purity hydrogen is obtained from a gas mixture of H_2, N_2, CO_2, CO and CH_4. The adsorbent bed is packed with APHP carbon and UOP 5A zeolite. Each bed undergoes eleven steps, and an idle step synchronizes five bed operations. This system is solved with a multibed optimization strategy. We maximize H_2 recovery with a 10 ppm CO level at cyclic steady state. Pressure constraints are imposed at each step (P_1, P_2–P_{10}, P_3–P_9, P_5–P_8, P_6, P_7, P_{11}). The decision variables include step times (T_5, T_{cycle}), valve constants (CV_1, CV_2, CV_3, CV_5), molar flow rates (mole_t_1, mole_t_2, *a*, *b*) and bed diameter. In Jiang et al. (in press) we compare the H_2 recoveries with CO levels at 10, 100 and 1000 ppm and observe a trade-off. H_2 recovery increases as the H_2 purity is reduced. However, the penalty for producing higher purity H_2 is not large. An upgrade in the purity from 1000 to 10 ppm decreases recovery by only 2.8%. Longer production times lead to higher H_2 recovery but at the cost of a lower production rate.

5. Computational results

Table 1 compares the CPU time usage by different methods for design and optimization. When additional design constraints are included, the computational effort for direct determination method rarely increases, which is a real advantage over successive substitution. Compared to Black Box approach, the simultaneous tailored approach can significantly improve the optimization efficiency and robustness. Note that Black Box approach failed in system 4 because of inaccurate finite difference derivatives. Also, because Black Box approaches require successive substitution for CSS, full parallelization is not possible with this approach and parallel implementation was not attempted.

Table 2
Percentages of non-parallelized parts for design and optimization

	Design (%)	Optimization (%)
System 1	1.717	0.57
System 2	4.86	1.03
System 3	1.914	0.86
System 4	N/A	1.36

Fig. 5(a) and (b) shows the speed up factors with increasing number of processors for design and optimization, using the direct determination and simultaneous tailored approaches, respectively. The speedup factors for each system are obtained by dividing the single processor times in Table 1 by the actual wall time under multiple processors. For optimization, the speedup factors are calculated based on time per rSQP iteration. With parallel computing, each processor solves a smaller set of variables so the accuracy level is slightly different with varying number of processors. For instance, the number of time steps and the number of nonlinear iterations for integration are smaller and the average step size is larger. This is a reason for super-linear speed up in system 1. The different level of accuracy of the sensitivities also affects optimization and leads to different numbers of rSQP iterations. On the other hand, the non-parallelized parts, such as function evaluation and search direction determination, limit the potential for speed up as the number of processors increases. The time spent on these parts is a fixed cost and the percentage of these non-parallelized parts increases when the total wall time decreases. Table 2 lists the percentages of non-parallelized (serial) parts in design and optimization with a single CPU, which explains the different speedup behaviors. For each system, the speedups are typically larger for optimization than for design, due to the smaller percentages for optimization. All speedup factors are sub-linear when the number of processors is large as the serial calculations now limit the throughput.

6. Summary

In previous work by Jiang et al. (2003, in press), we employ a Newton-based approach to quickly converge

the cyclic steady state with design specification, and a simultaneous tailored approach and the state-of-art rSQP optimization strategy to design optimal PSA processes. We find the competitiveness of these approaches depends on an efficient and accurate sensitivity evaluation. In this work, we parallelize this sensitivity evaluation, thus accelerating the design and optimization processes. Several O_2 and H_2 industrial cycles have been solved for illustration. In the future, we plan to implement the adjoint sensitivity approach. Although harder to implement, adjoint approach is more efficient than the direct sensitivity approach, especially when the number of sensitivity parameters is large.

Acknowledgments

The authors gratefully acknowledge the financial support of the NSF/GOALI program and Air Products and Chemicals Inc. The authors also thank Prof. Steinar Hauan for the support on the Beowulf computing facility and Dr. Shengtai Li for the help on DASPK package.

References

Biegler, L. T., Grossmann, I. E., & Westerberg, A. W. (1997). *Systematic methods of chemical process design*. Englewood Cliffs, NJ: Prentice-Hall.

Biegler, L. T., Jiang, L., & Fox, V. (in press). Recent advances in optimal design of pressure swing adsorption systems. *Separation and Purification Reviews*.

Bischof, C., Carle, A., Khademi, P., & Mauer, A. (1992). *The ADIFOR 2.0 system for the automatic differentiation of Fortran 77 programs*. CRPC-TR94491.

Croft, D. T., & LeVan, M. D. (1994). Periodic states of adsorption cycles-I. Direct determination and stability. *Chemical Engineering Science*, *49*, 1821–1829.

Ding, Y., & LeVan, M. (2001). Periodic states of adsorption cycles III. Convergence acceleration for direct determination. *Chemical Engineering Science*, *56*(17), 5217–5230.

Gropp, W., Lusk, E., & Skjellum, A. (1999). *Using MPI: portable parallel programming with the message-passing interface*. Cambridge: MIT Press.

Jiang, L., Biegler, L. T., & Fox, V. G. (2003). Simulation and optimization of pressure swing adsorption systems for air separation. *American Institute of Chemical Engineering Journal*, *49*(5), 1140–1157.

Jiang, L., Fox, V. G., Biegler, & L. T. (in press). Simulation and optimal design of multi-bed pressure swing adsorption systems. *American Institute of Chemical Engineering Journal*.

Kvamsdal, H. M., & Hertzberg, T. (1997). Optimization of PSA systems-studies on cyclic steady state convergence. *Computers and Chemical Engineering*, *21*(8), 819–832.

Li, S., & Petzold, L. (1999). *Design of new DASPK for sensitivity analysis*. Santa Barbara, CA, USA: University of California.

Li, S., Petzold, L., & Zhu, W. (2000). Sensitivity analysis of differential–algebraic equations: a comparison of methods on a special problem. *Applied Numerical Mathematics*, *32*, 161–174.

Sircar, S. (2002). Pressure swing adsorption. *Industrial Engineering Chemistry Research*, *41*, 1389.

Smith, O. J., & Westerberg, A. W. (1992). Acceleration of cyclic steady state convergence for pressure swing adsorption models. *Industrial Engineering Chemistry Research*, *31*, 1569.

Ternet, D. J., & Biegler, L. T. (1998). Recent improvements to a multiplier-free reduced Hessian successive quadratic programming algorithm. *Computers and Chemical Engineering*, *22*, 963–978.

Zhu, W., & Petzold, L. (1999). Parallel sensitivity analysis for DAEs with many parameters. *Concurrency: Practice and Experience*, *11*(10), 571–585.