UNIVERSITY COLLEGE LONDON

Efficient Targeted Optimisation for the Design of Pressure Swing Adsorption Systems for CO2 Capture in Power Plants

by

Joakim Henrik Beck

A Thesis Submitted for the Degree of Doctor of Philosophy
at the University College London.

Department of Chemical Engineering
University College London (UCL)
Torrington Place, London WC1E 7JE, United Kingdom

Declaration of Authorship

I, Joakim Henrik Beck confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.

Signed

Date

Abstract

Pressure swing adsorption (PSA) is a cyclic adsorption process for gas separation and purification, and can be used in a variety of industrial applications, for example, hydrogen purification and dehydration. PSA is, due to its low operational cost and its ability to efficiently separate CO_2 from flue gas, a promising candidate for post-combustion carbon capture in power plants, which is an important link in the Carbon Capture and Storage technology chain. PSA offers many design possibilities, but to optimise the performance of a PSA system over a wide range of design choices, by experimental means, is typically too costly, in terms of time and resources required. To address this challenge, computer experiments are used to emulate the real system and to predict the performance. The system of PDAEs that describes the PSA process behaviour is however typically computationally expensive to simulate, especially as the cyclic steady state condition has to be met. Over the past decade, significant progress has been made in computational strategies for PSA design, but more efficient optimisation procedures are needed. One popular class of optimisation methods are the Evolutionary algorithms (EAs). EAs are however less efficient for computationally expensive models. The use of surrogate models in optimisation is an exciting research direction that allows the strengths of EAs to be used for expensive models. A surrogate based optimisation (SBO) procedure is here developed for the design of PSA systems. The procedure is applicable for constrained and multi-objective optimisation. This SBO procedure relies on Kriging, a popular surrogate model, and is used with EAs.

The main application of this work is the design of PSA systems for CO_2 capture. A 2-bed/6-step PSA system for CO_2 separation is used as an example. The cycle configuration used is sufficiently complex to provide a challenging, multi-criteria example.

Acknowledgements

A special thanks goes to Prof. Eric S. Fraga, for his guidance, patience, and encouragement over these years of dissertation work. I also thank Dr. Serge Guillas for his valuable advice in the work related to statistical science.

I wish to acknowledge the Engineering and Physical Sciences Research Council for providing financial support over the duration of this project (grants EP/F034520/1, Innovative Gas Separations for Carbon Capture, IGSCC). I thank Prof. Stefano Brandani and Dr. Daniel Friedrich for our collaborative work during this project, as well as for providing computer code used in this work for simulating pressure swing adsorption processes. I also thank Dr. Audrius Varoneckas for insightful discussions on high-dimensional visualisation for process design. I also wish to thank all my collaborators for making research a lot of fun; especially Dr. Lorenzo Tamellini for friendship and for our enjoyable research collaboration.

My parents, whose love, never-ending support, and encouragement, have been a source of strength for me over these years, I cannot thank you enough.

I would like to thank my loving family. My children, whom continue to remind me to keep everything in perspective, as well as inspire me. I must acknowledge my wife and best friend, Amina, more than anyone, without whose love this would not have been possible. You support me, inspire me, and put a smile on my face when I need one. My love, this thesis is dedicated to you and our children.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Pressure swing adsorption (PSA) is a gas separation technique that has been studied for several decades, and can be seen as a cost-effective alternative to more traditional gas separation techniques, such as absorption. PSA has more recently emerged as a promising technique for separating out CO_2 from flue gas, and therefore of interest to Carbon Capture and Storage (CCS). PSA processes offer much flexibility in terms of process design, and have been shown to be promising for a wide range of applications. PSA units have already been installed worldwide, as home medical oxygen generators and air brake drying systems, but also for industrial hydrogen purification and air fractionation. Because of the complex nature of PSA processes, and the wide range of design possibilities, computer experiments are typically used to emulate PSA processes. To design PSA systems has been shown to be a challenging problem, mainly because the computer simulations are computationally expensive to run. There is a need for more efficient optimisation routines that can handle the design problem, especially for large-scale industrial applications. The application of interest in this study is carbon capture in power plants. In this chapter we provide the motivation for our work, and the scope. The final section will present the outline of this thesis.

1.1 Carbon Capture and Storage

There has been an increased interest in understanding the impacts of climate change and global warming, which have been observed over the past century. The scientific consensus is that global warming is due to the greenhouse effect, which is an effect intensified by the ever increasing global CO_2 emissions. Fossil-fuel power plants are accountable for approximitely one third of the total CO_2 emitted from man-made sources [AT05]. To achieve a satisfactory level of CO_2 emissions a near-term goal is to substantially reduce the CO_2 emissions to the atmosphere from the power plants. A possible mid-term goal could be to make better use of fuel in power generation and transportation, whereas the long-term challenge is to instead use alternative energy sources such as nuclear, solar, and wind power [AT05]. Fossil-fuel based power plants are responsible for 80 % of the global energy supply, with the coal-fired ones supplying as much as 42 % [MIT07, Adm10]. Coal-fired power plants are expected to continue to cover a large portion of the global energy supply for decades to come.

An initiative for achieving the near and mid-term goals is “Carbon Capture and Storage” (CCS), a strategy by the internationally joint Intergovernmental Panel on Climate Change (IPCC). The IPCC has been awarded the Nobel Peace Prize for 2007, a recognition “for their efforts to build up and disseminate greater knowledge about man made climate change, and to lay the foundations for the measures that are needed to counteract such change.” CCS is a process that, as a first step, uses gas separation to extract compressed CO_2 of high purity from an industrial source, followed by transportation of the captured CO_2 to a geologic storage reservoir (such as saline formations and oil or gas fields) or an ocean storage, for instance, and finally sequestration for long-term storage and isolation [IPC05]. The Climate Change Act 2008 (c.27) from the Parliament of the United Kingdom, includes a mandate to achieve a reduction in CO_2 emissions by at least 80 % for 2050. The baseline for comparison is the emission level recorded in 1990. In 2009 a UK research consortium was formed, under the name Innovative Gas Separations for Carbon Capture (IGSCC) and funded by the British Research Council EPSRC. The IGSCC aims to develop novel materials and processes for gas separation based on absorption, adsorption and membrane processes for CCS and investigate their capability based on technical performance, energy consumption, economical cost and sustainability. Our work is a part of the IGSCC effort. There are a few CCS technologies that are considered feasible from a scientific perspective. The CCS problem is however not only a scientific problem, it is also an economical one, since one of the critical objectives is to find a CCS technology that is economically viable. The capture system is required to meet certain economical limitations. According to U.S. Department of Energy, a feasible CCS technology should capture 90 % of the CO_2 emissions while keeping the resulting increase in electricity cost below 10 to 20 % (depending on CCS type) [oFEoS99]. For CCS the costs associated with the transportation, storage, verification and monitoring operations have been estimated to represent only one fourth of the total cost, whereas the remaining cost is related to the separation process [FFP+08, ER09].

1.2 CO$_2$ capture

The main challenges for CO$_2$ separation techniques are the low concentration of CO$_2$ in flue gas, and the occurrence of contaminant gases (oxygen, water vapour, sulphur oxides, and nitrogen) which makes CO$_2$ separation more difficult and might also lead to degradation of material and corrosion of equipment. A CO$_2$ capture system can be used for post-combustion, pre-combustion, or oxy-combustion.

Post-combustion

In post-combustion capture the CO$_2$ is separated from flue gas exiting from the combustion chamber. The flue gas composition is typically about 74 % N$_2$, 15 % CO$_2$, 7 % H$_2$O, 3 % O$_2$, and with a small fraction (less than 1%) of other particulates (e.g., SO$_x$, NO$_x$). The exhaust flue gas from the combustion is normally at ambient temperature and atmospheric pressure. In the combustion process, with air as the oxygen source, the following chemical reaction occurs

$$C_xH_y + \left(x + \frac{y}{4}\right) O_2 \rightarrow \text{Heat} + xCO_2 + \left(\frac{y}{2}\right) H_2O,$$

where stoichiometric coefficients x and y depend on the type of fossil fuel.

Pre-combustion

In pre-combustion capture the carbon dioxide is fully or partially separated from the hydrocarbon fuel gas, e.g. methane or gasified coal, to produce hydrogen-rich fuel gas for hydrogen combustion. Pre-combustion is a gasification process of coal (by partial oxidation reactors or coke oven gas units) that produces syngas, which consists of monoxide (CO), hydrogen (H$_2$), carbon dioxide (CO$_2$), and water (H$_2$O), along with small amounts of CH$_4$ and H$_2$S, occasionally. A CO shift converter is here installed to generate a CO$_2$/H$_2$ gas mixture out of a significant portion of the CO and H$_2$O involved. Much attention has in recent years been towards pre-combustion capture because of its low capital cost when compared to post-combustion capture [IPC05]. Also, the pre-combustion technology can be used in modern Integrated Gasification Combined Cycle (IGCC) power plants with gasification. For IGCC plants that burn methane the following chemical reactions occur before the CO$_2$ separation unit:

$$2CH_4 + \frac{1}{2}O_2 + H_2O \leftrightarrow 5H_2 + 2CO \quad \text{(auto thermal reforming reaction), and}$$

$$CO + H_2O \leftrightarrow CO_2 + H_2. \quad \text{(shift reaction)}$$

The resulting syngas is highly concentrated in CO_2 and at a high pressure.

Oxy-combustion

The fossil fuel is burned in an oxygen stream that contains little or no N_2. Oxy-combustion capture can be applied to existing PC power plants. A major drawback is that the air separation step before combustion is energy demanding. Oxy-combustion is outside the scope of this work, but we refer the interested reader to [FFP$^+$08, ER09].

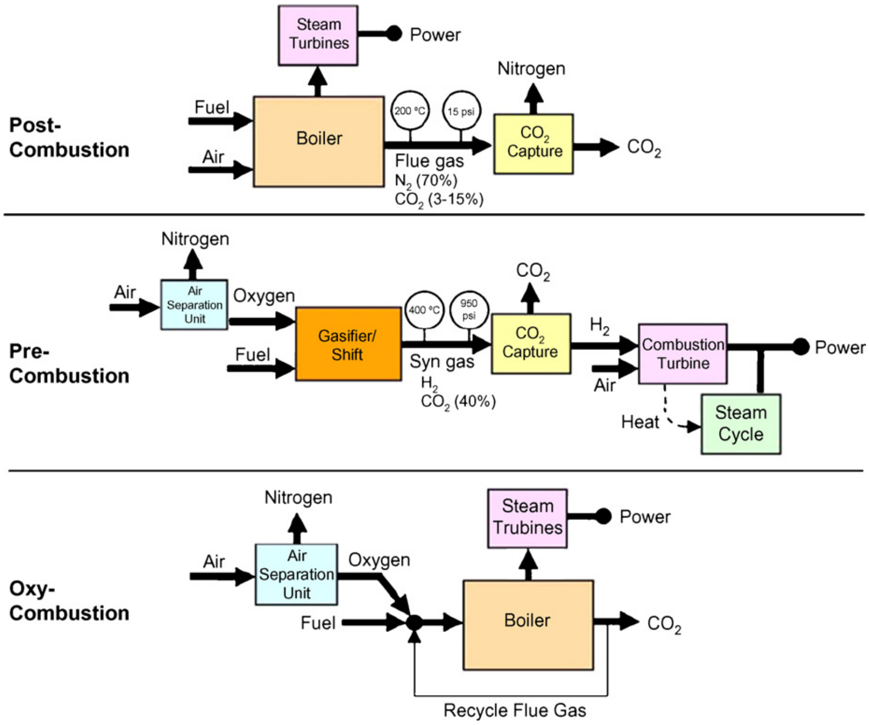


Figure 1.1: The diagram, originally published in [FFP$^+$08], illustrates three types of carbon capture systems: post-combustion, pre-combustion, and oxy-combustion.

Discussion

Combustion is a high-temperature process operating in the regime of atmospheric pressure. A barrier to the implementation of post-combustion carbon capture is that the CO_2 partial pressure is low in flue gas and much energy is therefore required by the separation process. For pre-combustion, on the other hand, the capture of CO_2 from fuel gas leads to a significantly higher CO_2 partial pressure and hence lower operational cost. The UK government has announced that they will allocate 1 billion GBP to support a CCS demonstration project of building a fossil-fuel power plant by 2014/15 [fBISB07]. The project plans to build a coal-fired power plant with a post-combustion technology. However, one prerequisite was that the CCS technology adopted should be viable to retrofit to pulverised coal-fired power plants. Post-combustion systems can be retrofitted to coal-fired power plants, whereas pre-combustion systems cannot. On the other hand pre-combustion is suitable for use in the more recent IGCC plants. In conclusion the post-combustion approach is favored over pre-combustion as a near-term solution, whereas the pre-combustion approach has in recent years been investigated as a long-term option.

1.3 CO_2 separation techniques

The choice of separation technique is very much problem-specific. A CO_2 separation technique should be capable to deliver high-purity streams of CO_2 at an acceptable economic cost. There exists a wide range of applicable commercial technologies for the separation of CO_2 from gas-mixtures: absorption into a liquid, adsorption on a solid, membranes, cryogenic liquefaction, or hybrid systems of some of the aforementioned. Even though the focal point of this thesis is pressure swing adsorption (PSA), we will here also present an overview of some alternative techniques, namely absorption and membranes. After that, we will justify the choice of PSA as a feasible option to achieve cost-effective separation of CO_2 for post-combustion carbon capture in power plants.

Absorption separation

Separating CO_2 from a gas mixture stream (e.g., flue or fuel gas) through absorption is a process that in principle scrubs CO_2 from the gas. A necessary property of the solvent is to favourably dissolve CO_2 above the other components in the gas mixture [AT05]. The other components of the gas mixture are referred to as carrier components. In the succeeding regeneration step the CO_2-rich solution is feed into a regeneration column, where the CO_2 is removed from the solution and the solvent is regenerated so that it can be reused for a new load of gas mixture. Absorption is best at low temperature and high pressure, whereas regeneration is best under the inverted conditions. The typical pressure level in an absorption process is atmospheric pressure, but could be designed differently to achieve a better performance.

An absorption process can be classified as either “chemical absorption” or “physical absorption.” A chemical absorption process reacts chemically with the gas mixture and is suitable for low pressure applications such as post-combustion, whereas a physical absorption process only reacts physically with the gas and is suitable for high pressure applications given the high partial pressure [ER09]. Physical absorption processes also suffer from fewer issues related to the solvent, such as degradation, compared to chemical absorption. A common chemical liquid solvent for CO_2 separation is monoethanolamine (MEA), which is regarded as an efficient option in terms of performance versus economical cost. MEA absorption is among the most promising [AT05]. Although, other options such as adsorption and membrane systems have potential of surpassing the performance of MEA absorption, if more efficient and commerciable materials would be developed [AT05]. Other examples of chemical solvents for CO_2 separation are diethanolamine (DEA), dissopropanolamine (DIPA), methydiethanolamine (MDEA), diglycolamine (DGA), alkaline liquid, and ammonia. Whereas for physical absorption, no efficient sorbent has been commercialised [FFP$^+$08]. Two examples of sorbents for physical absorption are chilled methanol and glycol ethers [ER09].

Absorption techniques are the most well-established in the industry, in particular for natural gas sweetening and H_2 recovery from syngas in the petrochemical industry [AT05, ER09]. In addition, absorption is also established for separating CO_2; in chemical process industry, absorption based separation is used to purify products via the removal of CO_2 to enforce process and final product requirements. There exists a range of solvents with good regeneration attributes, especially MEA absorption displays a high selectivity to CO_2 along with the capacity to generate a product stream with CO_2 concentration above 95% [AT05, LKL$^+$09]. Whereas on the other hand the main disadvantages are solvent degradation [LKL$^+$09];equipment corrosion due to the high level of oxygen concentration; secondary evaporation of amines due to high vapor pressure as amines react with CO_2 [LKL$^+$09, ER09], and the energy intensive regeneration [AT05]. A remark, 70-80 % of the operational expense is due to the large temperature change required in transition between the adsorption and regeneration steps [AT05, LKL$^+$09].

Membrane separation

Membrane separation processes are simple to implement, but the development and choice of appropriate membrane are on the other hand challenging. Permeable membranes (see, e.g., [LKL$^+$09]) follow the principle of selective permeation. Permeability is a measure of at what rate a gas component will propagate, and can be interpeted as the reciprocal resistance against mass transfer by the medium, in this case the membrane. The permeability depends on the membrane structure, the nature of the permeant species (size, shape, and polarity), and the interaction between the permeant species and the membrane.

In the permeation process a gas mixture stream is feed onto the membrane surface and gas components with higher permeation rate will pass faster through the membrane than components with lower permeation rate. The permeation rate depends on the partial pressure difference between each side of the membrane. In most applications the purpose is to allow passage

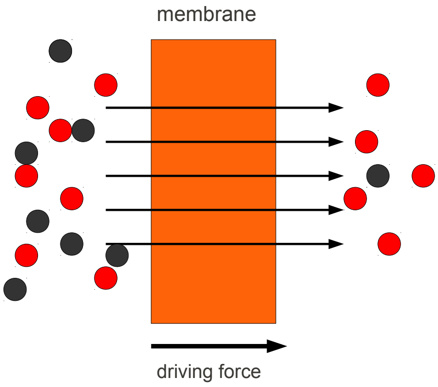


Figure 1.2: Membrane separation.

for a single component through the membrane, while rejecting passage of all other components involved in the gas mixture. There are three types of membranes for gas separation: polymeric, inorganic and biological. Only the first two will be discussed here. Polymeric can be divided into two categories: glassy and rubbery; similarly, inorganic membranes can be classified as either porous or dense.

Membranes are selected that exhibit a high permeation rate for CO_2, but a low rate for the other components involved. Some popular CO_2-permeant polymeric membranes for post-combustion are Polaris^{TM} and PIMs (see, e.g., [BM10, MLWB10]). In practice, polymeric membranes have been the most prominent in industry, for instance applied for purposes such as food and beverage processing, liquid desalination, and gas separation. Inorganic membranes have not been as extensively studied for CO_2 separation as polymetric membranes, but still have had some progress [YXF$^+$08]. Examples of inorganic materials for membranes are alumina, carbon, glass, metal, silica, zeolite and zirconia. As discussed in [AT05], the recent interest towards inorganic membranes is a product of the increased use of inorganic membranes for fuel cell and membrane reactor applications. What is specially attractive with inorganic membranes is their thermal and chemical stability [LKL$^+$09]. It could be argued that the best CO_2-selective inorganic membrane candidate to date is the zeolite membrane [LKL$^+$09].

Membrane techniques have the advantages of being simple (no moving parts, compact, easy to operate, control and scale-up), suitable for retrofitting in existing power plants, requiring little maintenance, environmentally friendly, and have a low energy demand since they are based on organic and biological systems and therefore have no phase change [FFP$^+$08]. The challenges are however the limited development of durable membrane supports of large surface area, “membrane wetting,” the fact that impurities in the gas stream can congest the membranes.

Adsorption separation

Adsorption is a process that attempts to force selected gas molecules in a gas mixture to be trapped close to a solid surface. When a gas molecule is near a solid surface, the molecules in the solid will apply an attractive force on the gas molecule that will reduce its potential energy. As a result the gas molecule density will be greater closer to the surface in the free-gas phase (see Figure 1.3).

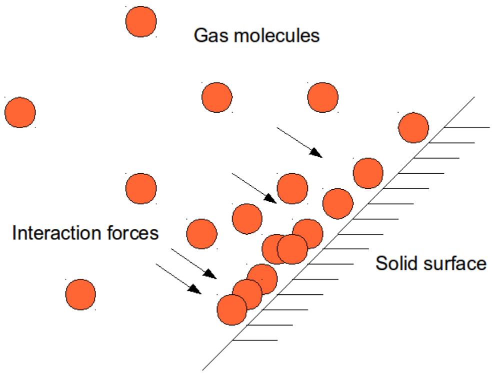


Figure 1.3: A simplified illustration of the principles of adsorption, when gas molecules are in close proximity to the solid surface with selectivity towards the molecules.

“Physical adsorption” is when the forces in the interaction between the adsorbing molecule and the surface are weak (van der Waals forces) and the adsorbate and the adsorbent are kept distinct; in contrast to what is known as “chemisorption,” which is when the interaction forces are strong enough for transfer or sharing of electrons between the adsorbate and the adsorbent. In practise most adsorption separation processes rely on physical adsorption [RFK93]. Examples of physical adsorption separation processes are pressure swing adsorption (PSA), vacuum swing adsorption (VSA), and thermal swing adsorption (TSA). Chemisorption on the other hand is regarded as an economically unattractive option because of its observed low capacity in applications.

Pressure swing adsorption

The pressure swing adsorption (PSA) process [RFK93] is a periodic adsorption process and is considered a viable energy and cost-efficient option which can be used for CO_2 separation from flue as well as fuel gas (see, e.g., [AT05, Sir06, HAW08, LKL$^+$09]). PSA processes have been applied for small, medium and large-scale purification and separation applications. As reported in 2006, at least a few hundred thousand PSA units have been installed world-wide [Sir06], mainly at the small-scale, for example as home medical oxygen generators and air-brake drying systems, but also implemented at medium- and large-scale for H_2 purification and air fractionation.

The main characteristic of a PSA process is to "swing" between adsorption and desorption process steps by increasing and decreasing the pressure in the system, respectively. PSA has attracted a lot of attention, as it is considered to be a cost-effective option for medium-scale operations [FFB09b]. PSA systems also have the ability to perform regeneration of the adsorbent bed without the need to request an interruption of the process. The vacuum swing adsorption (VSA) process, a type of PSA, has attracted much attention in the industry because of its simple design, low cost, and minimal corrosion and contamination [RFK93]. VSA is operating under mild conditions compared to other types of PSA systems. PSA processes are traditionally operated with a pressurised feed, whereas VSA processes are by definition carried out at near-ambient pressure with the gas driven by vacuum control. Another adsorption technique with similar characteristics as VSA/PSA is temperature swing adsorption (TSA). TSA varies the bed temperature in order to swing between the process of adsorption and desorption. The regeneration step from low to high temperature is however more energy demanding and slower compared to the one based on pressure swing. TSA is thus not as attractive as PSA from an economical and productivity perspective.

Adsorption processes allow regeneration to be executed on the fly, the systems can therefore consist of a series of process steps that either release (desorbe) or capture (adsorbe) the selective components. It is possible to use multiple "beds", with "interconnections" which allow gas flow between the beds. The wide range of design possibilities of PSA systems has made the design of PSA systems a matter of study [BJF05, ABZ10b].

There are two types of adsorption surfaces: solid adsorbents and porous adsorbents. With the solid adsorbents there are many challenges that need to be addressed for large-scale operations: it requires large volume of adsorbents; it exhibits a rapid decline in adsorption capacity, and there is a need for a regeneration process operated under large temperature swing conditions [LKL+09]. For porous adsorbents, 13X zeolite is currently commercially utilised for removal of CO_2 in an air separation process. A drawback with 13X zeolite is that it suffers from co-adsorption with water, i.e., 13X zeolite in the presence of water will lead to reduced adsorption capacity [LKL+09]. In contrast to lab-scale experiments, it is hard to rapidly change pressure and temperatures in large-scale adsorbent beds, and as pointed out in [LKL+09], it is important to develop and identify regeneration methods with strong suitability for large-scale operation. Also, more studies have to address the impact of gas impurities for the capacity, selectivity, and stability of the adsorption process [LKL+09].

PSA systems are considered to be promising for separation of CO_2 in large-scale operations. Recently there have been studies investigating PSA for both pre- and post-combustion [ABZ10b, ABZ10a].

Discussion

All the techniques presented for separating CO_2 from gas streams in a power plant setting have some issues that need to be addressed, or investigated. For example, liquid absorption with MEA may be viewed as the most promising CO_2 separation technique in a large-scale power plant setting [AT05], but this perception is typically based purely on the separation performance, and not in the light of the economic and environmental issues associated with it, which are important factors to consider. The key argument against absorption as a separation option for CO_2 is its high energy demand. Also, there has been opposition against amine-based absorption, as it may lead to environmental issues if the amine compounds are not properly contained. Membranes based methods need to prove themselves before being considered for large-scale applications. As seen in this section, there exist several alternative techniques for separation of CO_2 from flue gas and syngas.

However, only a handful are feasible for CCS, because most do not reach an adequate separation performance, or do not satisfy the requirements for acceptable economic conditions. The use of membranes is an option because of its simple use and low economic cost, but the gas streams in power plants have a very large volumetric flowrate, and there are no commerciable membrane materials to date that can withstand such enormous loading. A lot of attention has recently been devoted towards adsorption processes, in particular to PSA. PSA is a very promising technique for CCS, as it is considered to be a cost-efficient option with a technical performance comparable to absorption processes. Even though membrane based separation methods also have received a lot of attention in recent years, PSA is the more likely out of the two to be viable in the near future.

1.4 The operation of PSA cycles

In the PSA process, gas enters the adsorbent bed at high pressure (adsorption), and the feed-gas components most preferred by the adsorbent material become adsorbed. The remaining gas mixture not adsorbed passes through the bed and becomes the raffinate (light product). Eventually regeneration needs to be carried out in the bed by lowering the bed pressure so that the heavy gas components, which have been adsorbed, can be removed. This process is called desorption. The gas mixture released from the bed is the heavy product, which can either be removed or used as purge. Bed regeneration requires no interruption or restart of the process. PSA processes are thus continuous and cyclic. For a given system, all cycles are identically configurated and specified in terms of process steps executed in a sequential manner. PSA processes are operating at cyclic steady state (CSS). CSS is the state when the physical conditions at the end of a cycle are identical to those at the beginning of the cycle. To reach CSS from start-up could take hundreds or even thousands of cycles [THW$^+$01, BJF05]. PSA systems typically involve multiple adsorption beds with connecting streams. Common practise is to use the same sequence of steps for all beds but in shifted phase. In a multi-bed system a purge stream can be introduced during the desorption process to enhance regeneration.

1.4.1 Process steps

There are four basic PSA step types: pressurisation, depressurisation, adsorption, and desorption. A pressurisation step is characterised by high-pressure gas entering the bed while not permitting any gas to leave. A depressurisation step is characterised by no gas entering the bed while allowing gas to effuce at one bed end. An adsorption step is characterised by a high-pressure bed suitable for adsorption, with both ingoing and outgoing gas streams. The desorption steps operate in the same fashion as the adsorption step, but at low pressure so that desorption will take place.

A 2-bed/4-step PSA system is shown in Figure 2.1. This exact PSA system was designed by Charles W. Skarstrom in 1960, after which it was named the Skarstrom cycle, and is the earliest account of a PSA system [Ska60]. In the first step of Bed 2 the bed is pressurised with a gas mixture feed at high pressure. The second step is adsorption with a high pressure feed in which the gas mixture enriched in weakly-adsorbed components passes through the bed (light product). A fraction of the light product is channeled into a purge stream for the light reflux step of Bed 1. Bed 2 then performs countercurrent depressurisation in which the pressure within the

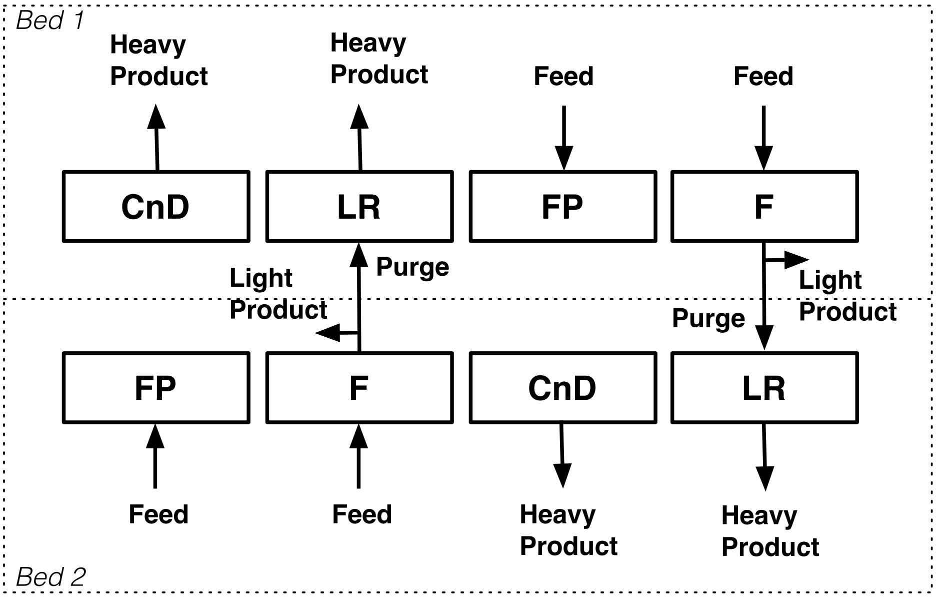


Figure 1.4: Schematic of 2-bed/4-step PSA system with a first-generation Skarstrom cycle. The steps included are feed pressurisation (FP), feed (F), countercurrent depressurisation (CnD), and light reflux (LR).

bed decreases as the gas mixture in the bed effuse (heavy product). The fourth and final step of the cycle in Bed 2 is desorption with a purge stream from the adsorption step of Bed 1, in which the heavy product is removed. The cycle is then repeated. As both beds have identical cycle configurations, the same process as in Bed 2 also occurs in Bed 1, but in shifted phase.

Here follows a list of process steps covering some of the most used (see, e.g., [RMER08, ABZ10b]): countercurrent depressurisation (CnD); cocurrent depressurisation (CoD); feed pressurisation (FP); feed or adsorption (F); feed plus recycle (F+R); heavy product pressurisation (HPP); heavy reflux (HR); light end equalisation (LEE); light product pressurisation (LPP); light reflux (LR); null, delay or idle (N); recycle of the heavy product (R), and recovery (REC). LPP represents the case when the light product from a highly pressurised bed, typically before depressurisation, is utilised to partially pressurise another bed. CnD is depressurisation with the same flow direction as the adsorption flow, whereas CoD is in counter-flow to the adsorption. Pressure equalisation steps are used to conserve energy [CSMM03]. They reuse gas that already is at high pressure in one bed to pressurise low-pressure beds in the system. A remark, the maximal number of equalisation steps that could fit inside a PSA system is equal to the number of beds. The use of pressure equalisation first appeared in [Ber66], using a modified Skarstrom cycle with a pressure equalisation step, and can significant reduce the power consumption of the system. The null/idle step, which represents the bed in an idle mode (that is, a state awaiting interaction with other beds), is sometimes necessary to include so that the beds are synchronised [MER11]. More sophisticated cycles could be designed with more elaborate bed interconnections. As an example, a 5-bed/11-step PSA system was designed in [JFB04], which included six pressure equalisation steps and one idle step, for an industrial application of PSA for H_2 bulk separation with H_2-N_2-CO_2-CO-CH_4 feed gas mixture.

Here follows a few examples of different types of PSA cycles:

- *Stripping reflux:* A stripping reflux cycle produces an enriched light product and a heavy waste product.
- *heavy reflux:* In contrast to stripping reflux, a PSA cycle classified as a heavy reflux [Wil82] is characterised by a light waste product of the gas mixture, and an enriched heavy product.
- *Duplex reflux* [DGH95]: A compromise between a stripping and heavy reflux cycle. It relies on a 2-bed system setup, for which one bed operating at high pressure whilst the other operating at low pressure. This leads to an enriched product at an intermediate pressure for the product streams at both bed ends. The feed stream can also enter at an intermediate position of the bed.

1.5 Challenges with PSA design

There are few important research directions that are persuited for PSA design for carbon capture [BJF05, ABZ10b, DVB12], for example, synthesizing novel adsorbent materials, improving the accuracy of the mathematical models describing PSA processes, developing computationally-efficient simulation and optimisation methods, and putting forward more systematic ways of choosing cycle schedule for specific tasks. The challenges related to synthesizing adsorbents, and improving the mathematical model for PSA processes, are outside the scope of our work. When designing a PSA system we have some design parameters that we are allowed to manipulated, such as valve constants, bed pressures, cycle times, bed dimensions, among others. The optimisation procedure will search through the design space of possible design configurations until we find the design that produces the "optimal" PSA system. The performance is often assessed based on some design criterion, for example, the purity of the product, or the system's power consumption.

Here follows an account of strategies proposed in the literature for the design of complex

PSA cycles.

1.5.1 Design of complex PSA cycles

A PSA cycle is described by the sequence of steps that the beds undergo. It is often assumed that the sequence of process steps to be performed in the PSA cycle is given, but this is not necessarily the case because each step is characterised by the connections between the steps (if open or closed), the feed, and the light and heavy product tanks.

Given the sequence of process steps, the graphical approach by Ritter et al. [MER10, MER11] can be used for complex PSA cycle scheduling. This approach, based on [Chi88], can easily generate and identify all PSA cycle schedules for the given sequence of process steps, under some constraints. This graphical framework divides the total cycle time into a set of unit cells such that the duration of any process step is required to be a multiple of the duration of a single unit cell, that is, a process step occupies one or several unit cells. The approach can generate all possible multi-bed systems with a given sequence of steps, even delay steps will be enforced appropriately to syncronise the beds.

PSA processes are very flexible, and the choice of process steps can be included as part of the cycle scheduling, as in [ABZ10b, ABZ10a], where a two-bed superstructure is proposed for pre- and post-combustion for CO_2 capture. In this design framework only two beds are considered, but that may not be a problem, as we later will see with the unibed approach [KFH$^+$94]. Many PSA schedules can be represented by a 2-bed structure. The 2-bed superstructure does not represent all possible bed interconnections, but many promising designs are likely to be explored. A bed is only allowed to interact with one other bed, no more, and all process steps can be classified as either adsorbing or desorbing. An advantage with this 2-bed superstructure is that the bed process behaviour is variable controlled, which means that only continuous decision variables are present, which avoids the need for mixed-integer programming. Here follows two more strategies worth mentioning. Fiandaca et al. [FFB09a] for multi-bed PSA cycles for CO_2 capture, where the cycle connections, and the associated valve constants, are generated at random between the process steps, the feed, and the light and heavy product tanks. In contrast, Nikolic et al. [NKG09] proposed a "state transition network" (STN) graph that describes all possible state transitions in a PSA system. In the STN, "states" are represented by process steps (e.g., found in [MER11]) and "state transitions" are decisions when a change of state should happen. In a STN, an initial state has to be specified, the authors suggest a co- or counter-current pressurisation or pressure equalisation step as the best candidate, in general. The STN procedure is as follows: begin in the initial state, a state transition is then decided by the optimiser in accordance with feasible transitions in the STN graph and under which conditions a transition occurs under the current state. For example, a pressurisation step can only be followed by the adsorption step, whereas the adsorption step can be followed by depressurisation, desorption, pressure equalisation, and others. However, the proposed STN graph does not include all of the most interesting state transitions [Aga10]. All of these strategies for PSA cycle scheduling make assumptions that restrict the search space. They provide a systematic way that can be used to identify viable PSA cycle schedules. The choice of the framework should be driven by the specifics of the problem at hand. Fiandaca et al. [FFB09a] provided the most general framework, but it can turn out to be an inefficient approach if the search space is not restricted enough to avoid excessive search in less interesting parts of the parameter space. Actually, the search space should not be too large, because of the computational expense associated with the simulation of PSA processes. This calls for efficient optimisation methods.

1.6 Problem statement

The focus of this thesis is PSA for post-combustion carbon capture, which is of great interest for the CCS process. We address the computational aspects that currently prohibit large-scale optimisation for PSA design. Although there are optimisation procedures that are considered adaquate for optimisation of PSA cycles, there are more difficult design problems for large-scale industrial applications that need to be addressed. In particular, this work is driven by the realisation that the PSA system is the key component for carbon capture in power plants, and should be assessed computationally as a carbon capture module in a power plant model. The performance will be evaluated online and should be designed while interacting dynamically with the other components of the power plant model. The PSA model is likely the most computationally expensive component of the carbon capture power plant model, and an efficient optimisation procedure should be able to take that into consideration.

To adress this, we have developed an optimisation procedure that is robust, and less time-consuming than preceding optimisation methods applied to the PSA optimisation problem. The proposed method relies on the surrogate model known as Kriging [Jon01]. Kriging is used as an approximation model for the simulation outputs (e.g. purity, recovery, and power consumption) over the entire search space. We have proposed a Surrogate-based Optimisation (SBO) procedure, where the Kriging has been utilised in various ways to guide the optimiser more efficiently. Several modifications have been suggested to make the SBO procedure highly robust and useful for constrained optimisation and for multi-objective optimisation. The use of a Kriging surrogate model should be able to signficantly reduce the computational time to run the PSA model, especially if a surrogate model is used to approximate the PSA component in the full power plant model with carbon capture.

The Kriging has been used in the work by Faruque Hasan et al. [HKF$^+$11] for a simple PSA case example for CO_2 capture, but the implementation is not robust. An alternative approximation-based optimisation method is given in [ABZ09], using reduced-order modelling (ROM) for hydrogen purification. This method is limited to tight search domains, and the robustness is questionable due to the low-resolution approximation involved. We have also developed a strategy for fast global sensitivity analysis, based on Kriging and polynomial chaos expansion [BBMF13], that can be used for screening design variables before the optimisation.

This new efficient PSA optimisation using surrogate models allows a much larger number of design configurations and adsorbent materials to be investigated, over a wide range of operating conditions, which would help in the CCS effort. For demonstration, we have selected two relevant, but also challenging, PSA examples, one is a dual-piston PSA and the other is a 2-bed/6-step PSA. which will show the efficiency of our proposed SBO procedure. This procedure can be used with most, if not all, computer simulation strategies currently used for PSA.

Furthermore, the proposed method is also capable of efficiently performing multi-criteria optimisation. Few studies have addressed multi-criteria design for PSA, which for some applications can be a very important design problem.

An efficient implementation can make it possible to run a model plant online and the costs in operation of a power plant with and w/o carbon capture can be compared in an efficient manner. A single model run of a surrogate model typically takes a matter of seconds, or less, whereas a single model run of a rigorous PSA model can take hours, or even days. A model run means the computer simulation of the PSA model for a single design configuration. The use of surrogate models can lead to huge computational savings, which will be explored in this work.

1.7 Objectives of the project and outline of the thesis

In this thesis the main goal is to develop efficient optimisation procedures for challenging PSA design problem formulations, and demonstrate them for PSA design, in the context of CO_2 separation from a feed with composition similar to the typical composition of flue gas (relevant to the CCS process). Our work has been presented in seven chapters, and the outline of the chapters are as follows:

In Chapter 2, the principles of adsorption are presented, which include adsorption equilibrium and mass transfer, followed by the mathematical models describing dynamic behaviour of a PSA process. The computer simulation strategies are described, as well as the discretisation of the mathematical models that set the scene for PSA simulation. In Chapter 3 we present the most used optimisation strategies for PSA. The final section presents the main concept of evolutionary algorithms which will be used throughout this work. Chapter 4 introduces surrogate modelling, and formally develop a surrogate-based optimisation (SBO) procedure using Kriging, given as a step-by-step strategy. Each step of the procedure is discussed in detail, and the goal is to achieve a computationally-efficient optimisation method. Because the PSA simulations are computationally expensive, the SBO method attempts to find the global optimum while using as few PSA simulator runs as possible. The Kriging surrogate model is presented, along with some adjustments that improve the numerical stability and predictive quality. Two SBO methods have been proposed, surrogate-based GA and the popular EGO (efficent global optimisation). The SBO methods are carried out for a dual-piston PSA design study with six design variables varied during the optimisation. The SBO methods and a real-coded GA algorithm are compared against each other. The dual-piston PSA is an interesting system used for testing and characterising adsorbent materials. This is the first design study for a dual-piston PSA system. In Chapter 5 we extend the SBO procedure presented earlier to cases where we have multiple objectives, and expensive design constraints. The procedure is extended to evolutionary multi-objective optimisation (EMO), and to multi-objective efficient global optimisation (MOEGO). The EMO of choice in our work is the popular NSGA-II. In addition, we have taken a technique from active learning, and applied it to sequentially improve the global accuracy of the Kriging model, and in turn the robustness of the SBO because it will promote exploration. A novel modification is also made to the Kriging surrogate model to ensure the physical constraints of the product purity and recovery (since they are ratios) are enforced for the Kriging predictions made. This is, in fact, particularly important for multi-objective optimisation, as illustrated with a few examples. The failed simulation runs are utilised to improve robustness by allowing the SBO methods to learn where the PSA simulator may fail. Chapter 6 illustrates the SBO procedure for multi-criteria design on a challenging 2-bed/4-step PSA Skarstrom system. For the multi-objective problems, the SBO methods are compared to NSGA-II. The SBO procedure is put to the test over a variety of design problem formulations. This chapter includes the first PSA optimisation study where we have three objectives optimised simultaneously, that is, the product purity, recovery, and power consumption. Finally, we summarise the contributions of this thesis, and discuss directions for future work in Chapter 7.

Chapter 2

Modelling and Simulation of PSA Processes

Pressure swing adsorption (PSA) is a cylic adsorption process suitable for gas separation and purification. In this process, some feed-gas mixture is passed through a bed packed with some adsorbent material. The adsorbent bed is packed with some adsorbent that should preferentially adsorb one or more components of the feed gas, which share similar characteristics, while the remaining gas passes through the bed. This is a complex process where the choice of adsorbent material is a critical factor. Hence, in order to understand the modelling and simulation of adsorbent beds, and in the end the design of such systems, it is important to first have an adequate understanding of the adsorption principles. An introduction to the adsorption principles that underpin the adsorption bed process is given in the following section. Then we present mathematical models that are used to describe the process dynamics involved. The main equations involved are the ones that describe the conservation of mass and energy, pressure profiles, and adsorption kinetics in the adsorbent bed. These models tend to be rigorous, and consist of coupled non-linear partial differential equations (PDEs) in space and time, as well as non-linear algebraic equations representing non-isothermal heat effects and adsorption isotherms. Finally, we provide an overview of computational simulation strategies for the solution of PSA models. The computer simulations performed rely on the spatial and temporal discretisation applied to the mathematical system.

2.1 Principles of adsorption

The adsorption forces that may arise between the surface of the adsorbent material and the surrounding gas molecules depend on the interplay between them. In this section the principles of adsorption are explained in order to show the importance of the choice of adsorbent material for PSA. First the three most influential factors for adsorption are described.

Selectivity. An adsorbent is desired to have a high affinity between the adsorbing molecule and the adsorbent surface. This corresponds to a high selectivity. An adsorbent has either equilibrium or kinetic selectivity. The equilibrium selectivity is when the separation depends on the difference in the equilibrium affinities. For equilibrium-selective adsorbents the impact of the mass transfer resistance is significant. Kinetic selectivity on the other hand depends on the difference in adsorption rates. For kinetic-selective adsorbents the kinetic factors are the predominant ones.

Capacity. The larger the surface area of the adsorbent, the higher the adsorption capacity. An adsorbent with low adsorption capacity tends to require a larger, more costly bed.

Physical strength. The degradation of the sorbent causes the adsorbent particles and its capacity to deteriorate over time. The strength mainly depends on the manufacturing of the adsorbents, pretreatment, and material properties.

The pore size influences the characteristics of the adsorbent, for example, the adsorption and desorption rates. The pore size distribution determines the level of resistance an adsorbed molecule needs to overcome to escape the attractive force from the solid surface. In a micropore the adsorbed molecule cannot leave the solid surface, even when positioned at the center of the pore. Whereas in mesopores and macropores, adsorbed molecules at the center of the pore are considered “free” as they experience little or no attractive force. Macropores have a considerably smaller surface area than the full pore volume, which allows the adsorbed molecules to penetrate at the center of the pore into the interior. Pores are often classified according to their size [Bur76]: micropores with diameters < 2 nm; mescopores with between 2 and 50 nm, and macropores > 50 nm. For homogeneous adsorbents, the pore structure is the same throughout the entire particle, and therefore the pore size distribution is unimodal. For composite adsorbents, the structure is non-homogeneous, and a clustering of microporous particles often tends to bind together. The non-homogeneous structure yields a bimodal pore size distribution where micropores within the particles connected through the macropores within the pellet. In equilibrium-controlled adsorption a wide pore size distribution could be acceptable; as opposed to kinetic-controlled adsorption which uses a narrow distribution of pore size as the selectivity mainly depends on steric hindrance and less on material properties. Several techniques to achieve control over the pore size are discussed in the book [RFK93], e.g., enforcing ion exhange and silanation. The layer of adsorbed molecules at the solid surface may be seen as a single thermodynamic phase.

2.1.1 Adsorption equilibrium

The state of equilibrium between the gas phase and the adsorbed layer phase at the surface and the convergence towards equilibrium are described by the laws of thermodynamics. Physical adsorption is an exothermic process; thus adsorption is favorable at low temperature, in contrast to desorption which is favorable at high temperature. At low concentration the adsorption equilibrium relationship could be described in a linear form by Henry's Law:

$$q = Kc \text{ or } q = K'p. \tag{2.1}$$

Here K (and K') is the Henry constant; q is the amount adsorbed; p is the pressure, and c is the concentration of the sorbate in the fluid phase. The temperature-dependent relationship between the Henry constant and adsorption equilibrium could be described by the van't Hoff equation:

$$K' = K'_0 \exp\left\{-\frac{\Delta H}{R_g T}\right\}, \text{ and } K = K_0 \exp\left\{-\frac{\Delta U}{R_g T}\right\}, \tag{2.2}$$

where T is the temperature; $\Delta H = \Delta U - R_g T$ is the total energy change on adsorption, and R_g is the universal gas constant. At high concentration levels the adsorption equilibrium relationship exhibits nonlinearity, making the linear Henry's law not ideal.

An adsorption isotherm describes the equilibrium of the sorption at the solid surface at constant temperature. Isotherm models depend on the type of underlying sorption. Brunauer's classification [Rut84] divides isotherms into five classes of isotherm curve shapes. See below for a selection of popular adsorption isotherm models:

- Langmuir isotherm:

$$\frac{q^*}{q_s} = \frac{b}{c^{-1} + b} \tag{2.3}$$

- Freundlich isotherm:

$$q^* = bc^{\alpha^{-1}} \tag{2.4}$$

- Langmuir-Freundlich isotherm:

$$\frac{q^*}{q_s} = \frac{b}{c^{-\alpha^{-1}} + b} \tag{2.5}$$

- The extended Langmuir isotherm (for multicomponent sorption):

$$\frac{q_i^*}{q_s} = \frac{b_i p_i}{1 + \sum_{j=1}^{N_\mathrm{c}} b_j p_j} \tag{2.6}$$

- The loading ratio correlation (LRC) isotherm:

$$\frac{q_i^*}{q_s} = \frac{b_i p_i^{\alpha_i^{-1}}}{1 + \sum_{j=1}^{N_c} b_j p_j^{\alpha_j^{-1}}} \tag{2.7}$$

  where $\alpha_i, \alpha_j > 0$ are correction factors, N_c the number of components, and i and j are the component numbers.

Here q^* is the saturated limit to the amount adsorbed q, and p_j is the partial pressure for component j. To account for the effects of non-isothermality, the equilibrium constants are typically temperature dependent. For example, the Langmuir isotherm constant, b, can depend on the temperature as follows:

$$b = b_0 \exp\left\{-\frac{\Delta H}{R_g T}\right\} \tag{2.8}$$

The Langmuir model has shown to be a good fit for isotherms for many adsorption related applications [RFK93]. Moreover, the Langmuir model is consistent with Henry's law in the low-concentration regime. Langmuir-Freundlich isotherm is a good fit in the high-concentration regime, but becomes less reliable in the so-called "Henry's law" regime. Freundlich-based isotherms have correction parameters that provide more degrees of freedom in their parametrization.

2.1.2 Mass transfer and diffusion in adsorbents

The resistance to mass transfer is different in homogeneous and composite adsorbents because of their differences in pore structure. In a composite adsorpent there are three kinds of resistance to mass transfer: external film resistance, macropore diffusion, and micropore diffusion.

External fluid film resistance. The external fluid film encircle the adsorption pellet and causes resistance to mass transfer. Although the external film resistance is recognised as a contributer to the overall resistance, it is often negligible when compared to the internal diffusional resistances.

Macropore diffusion. The mass transfer resistance appearing as a result of macropore diffusion depends on the ratio between the diameter of the pores and the mean free path. When the pores have a diameter considerably larger than the mean free path ($1/d_{\text{pore}} < 0.1$ [PG97]), the diffusion in the macropores becomes similar to molecular diffusion. The molecular diffusivity is characterised by

$$D_m \propto \frac{T^{1.7}}{p\sqrt{M}}, \tag{2.9}$$

where D_m is the molecular diffusivity, and M is the molecular weight. When on the other hand pores are small (i.e., $\frac{l}{d_{\text{pore}}} > 10$ [PG97]) and at low pressure, the diffusion accounts for the resistance created by collisions between the diffusing molecules and the pore wall, known as Knudsen diffusion. The Knudsen diffusivity [PG97] can be described as

$$D_K = 48.5 d_{\text{pore}} \left(\frac{T}{M}\right)^{\frac{1}{2}}. \tag{2.10}$$

Here d_{pore} is the diameter of the pore. In the transition region between molecular diffusion and the Knudsen diffusion, where both are considered important, an effective diffusivity can be calculated by

$$\frac{1}{D_{\text{eff}}} = \frac{1}{D_K} + \frac{1}{D_m}, \tag{2.11}$$

where $D_K, D_M > 0$.

Micropore diffusion. The micropore diffusion is the diffusion in pores with a diameter size of the same magnitude as the one for the diffusing molecule. In micropore diffusion the diffusing molecule is unable to leave the force field of the pore wall. All molecules are in the adsorbed phase, becase in the “small” pores there is no distinction between adsorbed molecules and gaseous molecules in the central region of the pore. This is a big difference to macropore adsorbents. Moreover, the micropore diffusivity is found to vary with the temperature as described by the Arrhenius equation

$$D_0 = D_\infty \exp\left\{-\frac{E}{R_g T}\right\}, \tag{2.12}$$

where D_0 is the corrected diffusivity, D_∞ is the diffusivity in the limit $T \to \infty$, and E is the activation energy for the process.

2.2 Mathematical modelling of adsorption beds

Mathematical models for describing adsorption and desorption processes are well-established and are considered to be in good agreement with experiments [RFK93, LHR99]. The full set of mathematical equations is typically a large, coupled, nonlinear system of equations. See the following list of mathematical equations required to describe processes in fixed-adsorption beds:

- Infra-particle mass transfer model for describing the adsorbate uptake rate by pellets;
- Adsorption equilibrium model;
- A material balance for each component;
- An energy balance for the gas, solid and the adsorbed phase;
- The relation between the pressure drop and the gas velocity (momentum balance).

Together these equations form a set of coupled partial differential and algebraic equations (PDAEs). We also need to specify initial conditions for the state, as well as the periodic boundary conditions. Besides the governing equations, many models can be differentiated based on the following model choices [RFK93]:

- plug flow or axially dispersed plug flow pattern;
- isothermal or non-isothermal (if accounting for heat effects);
- constant or linear change in fluid velocity along the bed, and
- the pressure drop along the bed (if assuming linear drop).

The remainder of this section will describe the details of the mathematical modelling of PSA processes in fixed-adsorption beds. For more details, we recommend [WL03] and references therein.

2.2.1 Intra-particle mass transfer model

Here follows a brief account of three commonly used mass transfer models, which describe the resistance to mass transfer occuring in a particle while the gas and adsorbate phase are in local thermal equilibrium (see, e.g.,[AB05, CMM06]): Fickian diffusion model, dusty gas (DG) model, and linear driving force (LDF) model. The models relate the mass transfer rate between the gas and the adsorbate phase to the fluxes in the pores and in the surface. The Fickian diffusion model and the DG model are closely related, for instance, both incorporate the molecular diffusion, Knudsen diffusion, and viscous flux.

Here follows the equations for the intra-particle mass balance in the Fickian model and the DG-model [CMM06]: the intra-particle mass balance for component i can be written as

$$\frac{\varepsilon_p}{R_g T}\frac{dp_i}{dt} + \rho_s \frac{dq_i}{dt} = -\frac{1}{r^s}\frac{d}{dt}\left(r^s[\epsilon_p N_i^p + N_i^s]\right), \tag{2.13}$$

where p_i is the partial pressure; q is the amount adsorbed; ε_p is particle porosity; T is the temperature; R_g is the gas constant; t represents the time dependency; r is the coordinate of the particle in a spherical coordinate system, and s is the geometry factor (0=slab, 1=cylinder, and 2=sphere). Here the terms N^p and N^s represent the fluxes in the pores and in the surface, respectively. The N^s term accounts for the surface diffusion flux and is given by

$$N_i^s = -D_i^s(T)\frac{q_i}{p_i}\rho_s\frac{dp_i}{dt}, \tag{2.14}$$

where D_i^s is the surface diffusion coefficient, and ρ_s is the particle density.

For the Fickian model and the DG model the N^p term is problem-specific [CMM06], accounting for the molecular diffusion and Knudsen diffusion. The two models are equivalent if every component except for the sorbent in the gaseous-mixture have a low concentration, or the Knudsen diffusion is dominant. Given that a linear adsorption isotherm model is deployed, and the viscous flux is negligible, both models can be expressed by the homogeneous diffusion equation:

$$\frac{dq_i}{dt} = \frac{1}{r^s}\frac{d}{dt}\left(r^s D_{M,i}^e \frac{\partial q_i}{\partial r}\right), \tag{2.15}$$

where $D_{M,i}^e$ are the effective homogeneous diffusion coefficients. The linear driving force (LDF) model, first introduced by Glueckauf and Coates [GC47], is a popular approximation of the homogeneous diffusion equation (2.15). The LDF approximation simplifies the expression for the mass transfer rate, and can be written as

$$\frac{d\bar{q}}{dt} = k(q^* - \bar{q}). \tag{2.16}$$

That is, the mass transfer rate is proportional to the difference between the surface concentration and the average concentration within the particle. The value assigned to k can be derived experimentally or estimated by

$$k = \Omega\frac{D}{r_p^2},$$

where D is the diffusivity constant; r_p the particle radius, and Ω the correction parameter. When the LDF model was introduced in [GC47], the parameter Ω was set to 15, regardless of the underlying process. For cyclic adsorption processes the value of Ω can be related to the cycle time. The choice of the lumped mass transfer coefficient k is a matter of study and much attention has been devoted to the following:

- the selection of k so that consistency with the diffusion model is achieved [AS92];
- describe the correlation between k and the cycle time [RHR86, AB05, TW06];
- the problem of lack of diffusivity data to estimate k [DA00], and
- the use of different values of k for adsorption and desorption operations [DA00].

The LDF model has been shown to be in good agreement with the original diffusion model at CSS. LDF approximations are used for determining the process behaviour at CSS, and not for predicting the full transient behaviour of the process, especially not accurate for transient behaviour in fast cycles [NS83, AB05].

2.2.2 Adsorption equilibrium model

Knowledge about the adsorption equilibria is necessary for achieving an accurate empirical prediction model. Unfortunately it is not a simple task to obtain reliable equilibrium data over specific process conditions, for example over the pressure and temperature ranges of interest. The most common adsorption equilibrium models have been described in Section 2.1.1. PSA involves both adsorption and desorption processes, which for a given system typically operate at the same temperature. The isotherm model should be close to linear to avoid the adsorption or desorption to become too low [RFK93].

Small deviations from linearity are acceptable in the isotherm. Linear least squares can be used to determine the deviation from linearity, but it is a non-trivial task to define what is an acceptable level of deviation when using this method for isotherm fitting. For example, how "small" should the sum of squared residuals be to ensure reversibility in the isotherm regime of interest in order to avoid residual concentration accumulation in the adsorbed phase.

2.2.3 Material balances

The flow pattern in a PSA bed is nothing else than the flow pattern in any fixed-adsorbent bed, which makes the axial dispersive plug flow pattern a suitable model. By assuming this flow pattern model, the material balances for the individual gas components can be described by

$$\frac{\partial}{\partial z}(uc_i) + \epsilon_b \frac{dc_i}{dt} + (1-\epsilon_b)\rho_s \frac{dq_i}{dt} = D_L \frac{\partial^2 c_i}{\partial z^2}, \forall z \in (0, L], \tag{2.17}$$

for $i = 1, \ldots, N_c$, where ϵ_b is the bed volume voidage; D_L is the axial dispersion coefficient; c is the cross sectional average concentration of the component in the fluid phase; u is the fluid velocity; q_i is the cross sectional average solid loading; N_c is the number of adsorbable components in the feed; L is the length of the adsorption bed, and z represents the spatial coordinate. The concentration and loading are summarised as cross sectional averages as no radial dependence is assumed. The first and second term on the left hand side in (2.17) represent the properties of the gaseous mixture, and the third remaining term accounts for the macropore material balance. The right-hand side, namely the axial dispersion term, represents axial mixing. If the axial dispersion term is omitted, we get the plug flow model. The plug flow approximation is mainly justified when the axial dispersion term is sufficiently small compared to the mass transfer resistance term. Also, when the Reynold number is large (> 100) the plug flow model could be a reasonable approximation [WL03].

A PSA system may be classified as a trace or a bulk separation system depending on the application. For a trace system the adsorbable components account for only a small fraction of the incoming feed with excess of inert carrier gases. For a bulk separation system the adsorbable components instead account for a large fraction of the feed. To consider the two types as different can be helpful because the dynamic behaviour within the system is expected to be very different. For example, one may assume constant velocity along the bed for trace systems when the pressure gradients are small, but not for bulk separation for which the velocity variation plays a more central role [RFK93]. For CO_2 separation from flue gas a bulk separation system should be used, whereas for typical hydrogen purification a trace system should be used.

2.2.4 Energy balances

The gas phase heat balance can be written as

$$
\begin{aligned}
C_g u \frac{\partial T}{\partial z} + C_g T \frac{\partial u}{\partial z} + \left(C_g + \frac{1-\epsilon_b}{\epsilon_b} C_s \right) \frac{dT}{dt} \\
+ \sum_{i}^{N_c} \Delta H_i \frac{1-\epsilon_b}{\epsilon_b} \frac{dq_i}{dt} + \frac{2h}{\epsilon_b r_b} (T - T_w) = K_L \frac{\partial^2 T}{\partial z^2}, \forall z \in (0, L], \quad (2.18)
\end{aligned}
$$

where r_b is the bed radius; h is the heat transfer coefficient; C_g is the heat capacity of the gas; C_s is the heat capacity of the solid; T_w is the temperature at the wall, and K_L is the thermal diffusivity. A gas phase heat balance equation describes the heat transfer. Here it is assumed that only the axial dispersion term contributes to the heat conduction in the system. The temperature at the bed wall is assumed to be equal to the temperature of the feed, but this assumption can be relaxed by introducing an energy balance for the bed wall. If the change in total energy is almost zero ($\Delta H \approx 0$) the isothermal conditions are met. In a multicomponent system, the gas density and heat capacity are taken as their averages with respect to the composition at each position in the bed. Studies have shown that isothermal PSA models often suffice as an approximation for kinetic-controlled separation since mass transfer rates are significantly slower than in equilibrium-controlled systems (see,e.g., [RFK93]). The temperature effects can be ignored if the trace component is weakly adsorbed [TW06].

2.2.5 Momentum balance

The steady-state momentum balance of gas flow relates the pressure drop to the gas velocity along the adsorbent bed. There are several ways of describing the steady-state momentum balance and the pressure transients, but perhaps the most popular are Darcy's law (see, e.g., [NP98, ABW99])

$$\frac{\partial p}{\partial z} = -\frac{180\mu u}{(2r_p)^2}\frac{(1-\epsilon_b^2)}{\epsilon_b^3}, \forall z \in (0, L], \tag{2.19}$$

and Ergun's equation (see, e.g., [ABZ09, JBF03])

$$\frac{\partial p}{\partial z} = -150\frac{\mu u}{(2r_p)^2}\frac{(1-\epsilon_b)^2}{\epsilon_b^3} + 1.75\frac{\rho M}{2r_p}\frac{(1-\epsilon_b)}{\epsilon_b^3}u|u|, \forall z \in (0, L]. \tag{2.20}$$

Here p is the pressure; μ is the gas viscosity, and M is the molecular weight. Other choices include the Blake-Kozeny equation [NGGK08], d'Arcy equation [CMM05], and adopting a constant pressure drop along the bed. The pressure drop effect becomes more important for short cycle times [SW88].

2.3 Computer simulation of PSA processes

The mathematical equations describing the PSA process are represented by a set of coupled parabolic/hyperbolic differential algebraic equations (PDAEs) over the spatial and temporal domain, along with the necessary initial and boundary conditions for the problem at hand. To obtain an analytical solution to such a large-scale nonlinear system of equations is often out of the question, but the system can instead in most cases be solved numerically using computer simulations. For an actual PSA process the time to reach cyclic steady state (CSS) is typically short when compared to its full life time, so a performance assessment of the system makes most sense at CSS.

2.3.1 The CSS solution to the PSA model

To determine the CSS solution to the PSA model, we can formulate the following simulation problem:

$$
\begin{aligned}
&\boldsymbol{F}_1\left(\boldsymbol{q}_1, \boldsymbol{q}_1', \boldsymbol{x}, t\right) = \mathbf{0}, \quad \boldsymbol{q}_1(t_0) = \boldsymbol{q}_0, \quad t \in [t_0, t_1) \\
&\boldsymbol{F}_k\left(\boldsymbol{q}_k, \boldsymbol{q}_k', \boldsymbol{x}, t\right) = \mathbf{0}, \quad \boldsymbol{q}_k(t_{k-1}) = \boldsymbol{q}_{k-1}(t_{k-1}), \quad t \in [t_{k-1}, t_k),\ k = 2, 3, \ldots, N\text{-}1 \\
&\boldsymbol{F}_N\left(\boldsymbol{q}_N, \boldsymbol{q}_N', \boldsymbol{x}, t\right) = \mathbf{0}, \quad \boldsymbol{q}_N(t_{N-1}) = \boldsymbol{q}_{N-1}(t_{N-1}), \quad t \in [t_{N-1}, t_N] \\
&\boldsymbol{W}\left(\boldsymbol{q}_N(t), \boldsymbol{x}\right) = \mathbf{0}, \quad t \in [t_{N-1}, t_N] \\
&\varepsilon(\boldsymbol{q}_N) = \boldsymbol{q}_N(t_N) - \boldsymbol{q}_N(t_{N-1}) = \mathbf{0}
\end{aligned}
\tag{2.21}
$$

Here $\boldsymbol{F}$ are the PSA bed model equations; $\boldsymbol{q}$ are the state variables, and subscript $k > 0$ represents the cycle number; $\boldsymbol{q}_0$ are the initial conditions for the state variables; $\boldsymbol{W}$ are the design constraints, and t_0 to t_N is the cycle duration. The set of equations for $\varepsilon(\boldsymbol{q}_N)$ are the CSS constraints. As the problem is stated, the CSS equations need to be strictly satisfied, or at least up to machine precision in a computer sense. In practice some small tolerance $\epsilon > 0$ is used, that is, $\|\varepsilon(\boldsymbol{q}_N)\|_2 < \epsilon$, where $\|\cdot\|_2$ is the Euclidean norm. The overall simulation problem is challenging, and demands a lot of effort from a computational perspective:

Non-linear Parabolic/Hyperbolic PDEs. When the axial dispersion term of the material balance is non-zero, the mathematical model has a parabolic nature. The solution to parabolic PDEs is smooth, although singularities could be present. Whereas when the axial dispersion is omitted, the model instead adopts a hyperbolic shape. In this case, any discontinuities of the initial and boundary conditions will propagate into the solution of the system. It is problematic that even if the input data to nonlinear hyperbolic PDEs is free from discontinuities, the PDE solution may not be. Hyperbolic PDEs tend to generate solutions suffering sharp fronts in the gas concentration profile, and non-physical oscillations due to shock waves [LeV02].

Convergence to Cyclic Steady State (CSS). The standard approach to find CSS is to use the final conditions of one cycle as the initial conditions for the next, that is,

$$\boldsymbol{q}_k(t_{k-1}) = \boldsymbol{q}_{k-1}(t_{k-1}), k = 2, 3, \ldots, N, \tag{2.22}$$

where N is the number of cycles, and subscript k represents the cycle number. This substitution is repeated from the initial cycle until the system has converged to CSS. This approach is called Successive Substitution (SS), also known as Picard iteration. SS is easy to implement, robust, and captures the process conditions during the transition to CSS. SS is performed over the same number of cycles as expected by the real-life system, which typically is a large number. Simulations based on non-isothermal models are expected to run more cycles than their isothermal counterparts. The real-life system may need hundreds or thousands of cycles to reach CSS [TFMW03]. For reducing the simulation run time there exist techniques for CSS acceleration [BJF05]. A selection of these techniques are summarised later.

2.3.2 The computational model

The computational PSA model is the computer code implementation of the discrete-mathematics version of the mathematical model. A computational model representation is necessary because computers can only interpret mathematics in discrete form. Here is an example of a complete discretisation using finite difference for the inviscid Burgers' equation:

$$\frac{\partial u(x,t)}{\partial t}+u(x,t)\frac{\partial u(x,t)}{\partial x}=0\rightarrow\frac{u(x_i,t_j)-u(x_i,t_j)}{t_j-t_{j-1}}+u(x_i,t_j)\left(\frac{u(x_i,t_j)-u(x_i,t_j)}{x_i-x_{i-1}}\right)=0, \tag{2.23}$$

where $u(x,t)$ is partitioned over the spatial domain $[x_0, x_1, \ldots, x_i, \ldots, x_M]$ with M intervals, and the temporal domain $[t_0, t_1, \ldots, t_j, \ldots, t_N]$ with N intervals.

2.3.3 Simulation strategies

There are two strategies for constructing computational models from a set of non-linear PDAEs:

Complete Discretisation. The Complete Discretisation (CD) approach entails discretisation of the PDAEs on a fixed grid over both the spatial and temporal domain. Then, the mathematical problem is completely discretised into a large-scale system of non-linear algebraic equations, for which Newton's method can be employed. This brute force-like discretisation is hard to combine with adaptive time stepping, and for this reason the non-linear system could in many cases become too large, making the approach computationally intractable.

Method of Lines. Method of Lines (MoL) is a two-step approach (see [CS98]): first it produces a semi-discrete model, and then perform the simulation by solving the computational model of the PSA model on a fixed grid, or on a grid adaptively refined/coarsened, in the temporal domain with integration over time. A spatial discretisation scheme is used, such as the finite difference method, resulting in the semi-discrete problem which forms a large system of differential algebraic equations (DAEs). Then for the simulation to run, a time integration routine is applied.

Since MoL treats the spatial and temporal domain independently there are many ODE/ADE solvers available for time-adaptive integration, such as Runge Kutta methods. However, when applying the spatial discretisation to create a system of ODEs/DAEs, the resulting system can become numerically unstable unless extremely small time steps are applied [Dav13]. Such systems are known as stiff systems, and to circumvent this issue complete discretisation can be used, or you need to resort to solvers that are developed for handling stiff DAEs.

The discretisation schemes for the spatial derivatives of the continuous problem have to be judiciously chosen to avoid numerical issues, which may cause simulation failure or result in an inaccurate representation of the underlying mathematical model. Hyperbolic PDEs are particularly difficult to discretise. Spatial discretisation schemes should mitigate the non-physical traits that may arise, such as oscillatory behaviour near steep adsorption fronts. For fast PSA cycles we expect shock waves during the pressurisation [SW88]. The most popular methods for spatial discretisation of hyperbolic PDEs are finite volume methods [LeV02], which respect the conservation laws. Other chocies for PDEs are the finite difference methods, and the finite element methods [EEHJ96].

The simulation time can be reduced by tuning the internal parameters for the spatial scheme, such as tolerance and grid size. To avoid selecting a too large error tolerance, or grid size, some tampering is warranted to fit the computer model to our expectation based on our knowledge of the system and model structure, and in some cases empirical data if available. Adaptive strategies are often used to refine the spatial grid in regions of the spatial domain where we expect larger derivatives, and coarser in regions that show little spatial variability.

Here follows a list of some of the schemes tested for PSA simulation:

- Finite difference [KSB05, Fia10]
- Finite volume method [HBEF12] (Modified van Leer limiter [JBF03, JFB04, JBF05, ABZ09], QUICK [WH00], Adaptive QUICK [THW$^+$01], SMART [CSMM05], WENO scheme [HMN$^+$13])
- Orthogonal collocation on finite elements (Radau scheme [TE99, NGGK08])

Low-order finite difference methods, and even finite element methods, often introduce unrealistic numerical oscillations near the steep adsorption fronts. For this reason high resolution methods may be required. Spatial discretisation schemes for PSA models have been explored thoroughly and we draw the conclusion that the finite volume is the prevailing method.

In some cases it is possible to simplify the mathematical model, which can be convenient computationally. For example, when the material balance of the PSA model can be written as a series of continuous stirred tank reactors (CSTRs) [Fia10], the discretisation along the spatial direction is not required. Also, then the material balance of the fluid phase is no longer dominated by a convective term as in the plug flow reactor (PFR) model [CMM05]. Alternatively, there are DAEs that can be used to describe the mass and energy balances [SIW91].

For PSA simulation, when following the Method of Lines (MOL) strategy, we are interested in solving a system of DAEs of the following form:

$$\mathbf{0} = \mathbf{F}\left(\mathbf{u}(t), \frac{d\mathbf{u}(t)}{dt}, t\right), \quad t_0 \leq t \leq t_N, \tag{2.24}$$

with initial conditions for $\mathbf{u}(0)$ and $\frac{d\mathbf{u}(0)}{dt}$ when needed to satisfy the degrees of freedom of the system. DAEs are a generalisation of ordinary differential equations (ODEs)

$$\frac{d\mathbf{u}(t)}{dt} = \mathbf{F}\left(\mathbf{u}(t), t\right), \quad t_0 \leq t \leq t_N, \tag{2.25}$$

for which there is a much broader range of numerical solvers. DAEs solvers are used for large-scale nonlinear systems, often generated by discretisation of the time derivatives. Some popular temporal discretisation schemes are Euler methods, Runge-Kutta methods, and Backward Differentiation Formulas (BDFs) [Fau99]. BDF methods are high-order methods with the stiff decay property, making it suitable when solving challenging stiff DAE systems. The k-step BDF method with constant step size applied to (2.24) is given by

$$\mathbf{0} = \mathbf{F}\left(\mathbf{u}(t_i), \frac{1}{\beta_0(t_i - t_{i-1})} \sum_{j=0}^{k} \alpha_i \mathbf{u}(t_{i-j}), t_i\right), \tag{2.26}$$

where β_0 and α_j are some specific coefficients for the BDF method. DAE solvers can also feature some adaptive time-stepping procedure. An automatic time step control can be added, that is,

$$t_{i+1} = t_i + \left(\frac{e_{i-1}}{e_i}\right)^{k_p} \left(\frac{\mathrm{TOL}}{e_i}\right)^{k_I} \left(\frac{e_{i-1}^2}{e_i e_{i-2}}\right)^{k_D} (t_i - t_{i-1}), \tag{2.27}$$

$$e_i = \frac{\|\mathbf{u}(t_{i+1}) - \mathbf{u}(t_i)\|_2}{\|\mathbf{u}(t_{i+1})\|_2} \tag{2.28}$$

where e_n is the error indicator for time step i. Here TOL, k_p, k_I, and k_D are tunable parameters.

Some DAE solvers that have been applied to PSA models are, for example, DASSL (BDF; [BCP89, BFB^{+}12]), ADIFOR 2.0 (Automatic Differentiation), DASPK 3.0 (BDF with sensitivity analysis capability; [JFB04, JBF05]), DASPK 3.1 (BDF), DASOLV (BDF; [TE99]), IPOPT 3.4 (Orthogonal Radau collocation on finite elements; [ABZ10b]).

2.3.4 Numerical techniques for fast determination of CSS

The objective of the simulation problem is to determine some quantities of interest at CSS under given constraints. However, the use of the Successive Substitution (SS) method to CSS is typically very costly. One approach to speed up the SS method is to use Successive Node Refinement [THW^{+}01], which is linked to the spatial discretisation procedure. In [THW^{+}01], it has been reported that the overall CPU time for simulation can be reduced by half, or even more, when applying this technique for a 4-step VSA for oxygen enrichment. Similar results has been reported for a dual piston PSA system for CO_2 separation in [FFB13]. The importance of the resolution level of the spatial discretisation was also underlined in [JFB04], where the number of spatial discretisation nodes was reduced from 81 to 18.

Successive Node Refinement. Using MOL with SS, the steps of this procedure are:

Step 1. Select a coarse spatial discretisation, e.g., 8 nodes, for the computational model.

Step 2. Integrate the resulting DAEs over time with SS until CSS is achieved. The quantities of interest are then calculated, for example Purity and Recovery.

Step 3. Select a finer spatial discretisation level than what was used in Step 2.

Step 4. Finish if the difference between two successive runs is small enough in some error norm. Otherwise repeat from Step 2, but with the initial conditions for this finer discretisation set to values determined from the interpolation of the CSS solution for the previous node scheme.

The difference between two successive runs becomes smaller as the number of nodes increases. Eventually a good balance can be achieved between spatial resolution and computational efficiency.

Often the CSS-based simulation problem, (2.21), does not need to be solved fully, as it would be enough to identify the CSS cycle, $\boldsymbol{q}_N(t), t \in [t_{N-1}, t_N]$. With the SS method all cycles from start-up to the onset of CSS are simulated. To circumvent the need to simulate all cycles we can employ techniques for direct determination of CSS. The keyword “direct” refers to the determination of CSS being the goal, not to perform the full simulation. These methods assume that the CSS cycle is unique, and in turn insensitive to the initial conditions in place at the startup of the process. Direct determination of CSS can lead to a substantial reduction in the time spent to identify CSS, although at the expense of missing the full transient behaviour of the process.

Direct determination with quasi-Newton methods: Smith and Westerberg [SW92] tackled the nonlinear system of CSS equations directly:

$$\varepsilon(\boldsymbol{q}_0(z), \boldsymbol{x}) = \boldsymbol{q}_0(z) - \Psi\left[\boldsymbol{q}_0(z), \boldsymbol{x}\right] = \mathbf{0}, \tag{2.29}$$

where z represents the spatial coordinate along the bed (of length L). $\Psi\left[\boldsymbol{q}_0, \boldsymbol{x}\right]$ are the final conditions obtained from a simulation run of a single cycle with initial conditions $\boldsymbol{q}_0$, and design configuration $\boldsymbol{x}$. The CSS equations can be solved simultaneously with Newton's method. Newton's method requires its Jacobian matrix to have elements with analytical expressions, which can be difficult to obtain due to the nonlinearity of typical PSA models. Quasi-Newton methods therefore tend to be more practical, as they try to find the root to (2.29) by using successive estimates of the Jacobian. One example of such a method is Broyden's method [Bro65], which uses approximations of the Jacobian matrix and its inverse. Quasi-Newton methods for optimisation problems typically require the Hessian, but here we are interested to solve the root optimisation problem which only needs the Jacobian.

Smith and Westerberg [SW92] applied Broyden's method, with the first three cycles performed with SS for the estimation of the Jacobian matrix. The inverse of the Jacobian matrix is then updated periodically at successive iterations using the Broyden update formula. Without updating the Jacobian inverse the solution of (2.29) cannot be guaranteed. Kvamsdal and Hertzberg [Kva97] instead applied Broyden's method with Muller's update formula, which is only appropriate when the state variables are not conjugated too much. Broyden's method achieves superlinear convergence rates near the solution [Gri86]. To update the Jacobian matrix is costly due the size of the Jacobian and the non-sparsity of the mapping $\Psi\left[\cdot\right]$. Also, convergence is not guaranteed, not even for cases for which Newton's method is known to converge [SW92]. Given that the the fixed adsorbent beds are packed in a continuous way a modification can be made to Broyden's method to achieve a speed up of an order of magnitude in the calculation time of the Jacobian matrix [DL01]. This approach approximates some of the Jacobian matrix evaluations by interpolation with piecewise polynomials. The first Newton-step in Broyden's method can also be avoided by replacing it with a SS step.

Direct determination with Newton methods: Newton methods are difficult to overlook with their guaranteed quadratic convergence *near* the solution. So instead of employing quasi-Newton methods, Croft and LeVan [CL94] suggested the use of sensitivity analysis to preserve the quadratic convergence of Newton's method. Without loss of generality, let us assume that (2.29) is a system of DAEs. A remark, if the system consists of PDAEs, very often it can be semi-discretised with a spatial discretisation scheme into a system of DAEs [HNW91]. With sensitivity analysis, the analytical sensitivities for (2.29) can be calculated by

$$\frac{\partial \boldsymbol{\varepsilon}(\boldsymbol{q}_0(z), \boldsymbol{q}_0'(z), \boldsymbol{x})}{\partial \boldsymbol{x}} \frac{d\boldsymbol{x}}{d\boldsymbol{q}_0(z)} = -\left(\frac{\partial \boldsymbol{\varepsilon}(\boldsymbol{q}_0(z), \boldsymbol{q}_0'(z), \boldsymbol{x})}{\partial \boldsymbol{q}_0(z)} + \frac{\partial \boldsymbol{\varepsilon}(\boldsymbol{q}_0(z), \boldsymbol{q}_0'(z), \boldsymbol{x})}{\partial \boldsymbol{q}_0'(z)} \right) \tag{2.30}$$

where $\boldsymbol{\varepsilon}(\boldsymbol{q}_0(z), \boldsymbol{q}_0'(z), \boldsymbol{x}) = \boldsymbol{q}_0(z) - \Psi\left[\boldsymbol{q}_0(z), \boldsymbol{q}_0'(z), \boldsymbol{x}\right]$ for all $z \in [0, L]$. The sensitivities are first calculated and then inserted into the Jacobian matrix of the DAE system consisting of the CSS conditions (2.29), the sensitivity matrix (2.30), and the design/process constraints $\boldsymbol{W}$. The sensitivites can be computed with for instance the finite difference method [DS96] or direct sensitivity methods. The finite difference method is of low order accuracy, and is known to suffer from scaling issues. Nevertheless, for well-scaled problems it has been shown to be reasonable for sensitivity calculation in the PSA setting [JBF03]. Direct sensitivity methods have been shown to be quite accurate for PSA, they rely on integration of the analytical sensitivities, and the DAEs, simultaneously over a cycle $t \in [0, t_\mathrm{c}]$. At each iteration of Newton's method the new state variable conditions with sensitivities are calculated for the next step. Unfortunately, the direct sensitivity methods suffer from high computational complexity associated with the sensitivity matrix (2.30), which exhibits high non-sparsity. The computation cost also quickly increases with the number of state variables, for forward sensitivity methods, and with the number of design variables, for adjoint sensitivity methods [PLCS06]. The direct determination of CSS with Newton's method has been demonstrated for PSA model in the context of air separation [JBF03]. The calculation of the sensitivities can be performed in parallell [JBF05]. Another method employed for the computation of the partial derivatives in the sensitivity matrix (2.30) is automatic differentiation method [JBF03].

Direct determination with optimisation-based methods:

A more recent approach to solve the simulation problem is to minimise the sum of squares of the CSS residuals [LST08]:

$$\tfrac{1}{2}\boldsymbol{\varepsilon}(\boldsymbol{q}_0, \boldsymbol{x})^T \boldsymbol{\varepsilon}(\boldsymbol{q}_0, \boldsymbol{x}), \tag{2.31}$$

with respect to the state variables at the beginning of the cycle, $\boldsymbol{q}_0$, while satisfying the constraints $\boldsymbol{W}$ of the simulation problem (2.21). Here $\boldsymbol{\varepsilon}(\boldsymbol{q}_0, \boldsymbol{x})$ is defined by (2.29), and $\boldsymbol{\varepsilon}(\cdot)^T$ is the transpose of $\boldsymbol{\varepsilon}(\cdot)$. The solution (state variable vector) to this optimisation problem is equal to the initial state conditions of the CSS cycle. Latifi et al. [LST08] applied a gradient-based nonlinear programming method (SQP) to solve this optimisation problem for a case example for CO_2/N_2 separation with a 2-bed/4-step Skarstrom PSA system. The adjoint method may be used for calculating the sensitivities when the number of decision variables is large. This optimisation-based approach has been reported to cause a high number of convergence failures when the spatial discretisation is too fine (>40 finite volume elements) [ATL11].

2.3.5 The Unibed approach: multiple-bed model reduction

To simulate the dynamic process behaviour taking place in a single PSA bed is computationally demanding, but even more so when the PSA unit consists of multiple beds with complex interconnections between them. In a PSA system the sequence of process steps in all beds are identical but with a phase difference. Kumar et al. [KFH$^+$94] showed that for every multiple bed system there exists a single bed (quasi-)system, henceforth called the Unibed system, with the same performance at CSS. The Unibed system is a “quasi-system”, because it requires virtual storage to mimic the interconnections between the beds in the original system. The Unibed model is easy to derive. However, the Unibed system does not guarantee an accurate description of the transition to CSS.

The Unibed system of a multiple bed system does not dismiss any bed interconnections involved in the original system, it introduces so-called virtual data storage in the computer code implementation in order to store the outgoing gas stream values from one bed to another. So instead of the outgoing gas stream entering the other bed instantaneous, the outgoing gas stream will be stored virtually until the receiving process step occurs for the first time, which is in the next cycle. This will preserve the bed interconnections and their effects at CSS. When the process has reached CSS the performance attributes of the Unibed system and the original multiple bed system will be identical [KFH$^+$94].

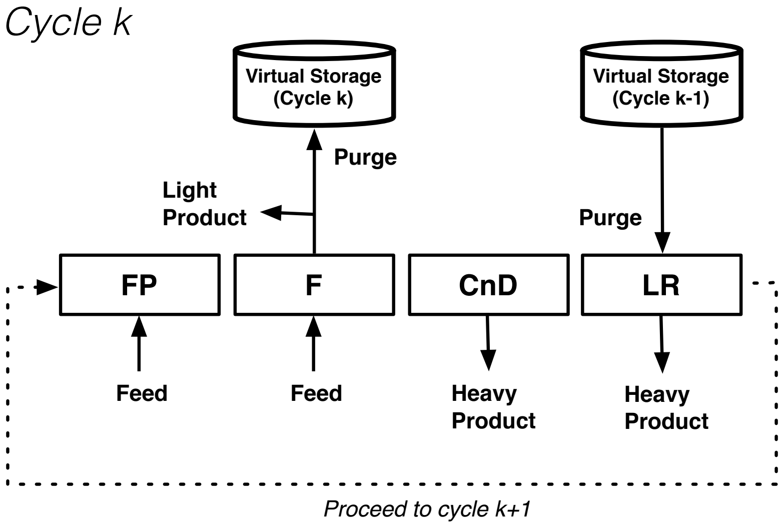


Figure 2.1: Unibed system of a 2-bed/4-step PSA system with classical Skarstrom cycle. The process steps included are feed pressurisation (FP), feed (F), countercurrent depressurisation (CnD), and light reflux (LR).

2.4 Summary

We started with a review of the principles of adsorption, we described the physics behind pressure swing adsorption processes that can be modelled mathematically using coupled PDEs with algebraic equations. The mathematical models are highly detailed, with well-established theory even at the microscopic scale, and thus it is possible to produce accurate models to describe the process behaviour of PSA processes. We also presented a range of computer simulation techniques to numerically solve the models. Because the performance of PSA processes are most desired at CSS, much development has been made for direct determination of the CSS, rather than simulating cycle-to-cycle until the CSS is reached. The latter is known as the method of successive substitution, and is the most robust of the simulation approaches, but also the most time-consuming. The solution of the hyperbolic PDAEs describing PSA processes are, due to high nonlinearities generated by the non-isothermal effects, as well as the non-linear isotherms, typically suffering from sharp fronts and non-physical oscillations, and thus provide a challenge for the simulation method. To address this, the use of finite volume methods tends to mitigate such undesired effects caused by the hyperbolic nature of the problem. Because the goal of this thesis is to produce efficient optimisation, the robustness is vital, hence the recommendation is to adopt successive substitution with finite volumes for simulation driven by backward differentiation formulas (BDFs).

Chapter 3

Computer-aided Design of PSA Systems

This chapter begins by giving an overview of the strategies proposed in the literature for the optimisation of PSA cycles. We are also discussing the different choices of optimisation algorithms that can be used with these strategies. Finally, we have identified some of the problems facing engineers when asked to design a PSA system.

3.1 Optimisation strategies for the design of PSA cycles

The performance of a PSA system can vary much between different design choices, and to simulate the PSA process for each choice can be very computationally expensive. With predictive computer simulation tools as those presented in the previous chapter, the process conditions (such as flow rates, and valve constants) can be manipulated with the help of optimisation routines. This highlights the need for efficient optimisation methods that are able to identify the most promising designs using as few simulation runs as possible.

What makes the PSA design problem very challenging is that it usually involves multiple, often conflicting, design criteria. See the list below for the most common design objectives used in literature:

- Maximise purity and/or recovery w/o design constraints
- Maximise purity and/or recovery with design constraints
- Minimise power consumption at desired purity and recovery

The design constraints can depend on, for example, the desired level of product purity and recovery, bed pressure limits, and production rate requirements. The technical objectives such as the product purity and recovery are easy to calculate from knowing the bed conditions over the CSS cycle, whereas economic objectives such as power consumption are considerably harder to express without economic empirical data. Actual economic data are often not made available to the public by the industries. The purity and recovery are conflicting objectives, and can therefore be treated in a multi-criteria framework. Literature suggests that a significant enrichment of the product tends to add a small penalty on the recovery (see [WS00, JFB04]). The correlation between the power consumption and purity/recovery is less known, but it is reasonable that there is a positive correlation between the purity of the product and power consumption [FFB09b].

Utilizing the simulation problem formulation (2.21), the general PSA optimisation problem can be expressed as follows:

$$\begin{aligned}
&\arg\min_{\boldsymbol{x}} \quad \boldsymbol{\phi}(\boldsymbol{q},\boldsymbol{q}_0,\boldsymbol{x})\\
&\text{s.t.}\\
&\qquad \boldsymbol{F}_1\left(\boldsymbol{q}_1,\boldsymbol{q}_1',\boldsymbol{x},t\right)=\boldsymbol{0},\quad \boldsymbol{q}_1(t_0)=\boldsymbol{q}_0,\quad t\in[t_0,t_1)\\
&\qquad \boldsymbol{F}_k\left(\boldsymbol{q}_k,\boldsymbol{q}_k',\boldsymbol{x},t\right)=\boldsymbol{0},\quad \boldsymbol{q}_k(t_{k-1})=\boldsymbol{q}_{k-1}(t_{k-1}),\quad t\in[t_{k-1},t_k),\ k=2,3,\ldots,N\text{-}1\\
&\qquad \boldsymbol{F}_N\left(\boldsymbol{q}_N,\boldsymbol{q}_N',\boldsymbol{x},t\right)=\boldsymbol{0},\quad \boldsymbol{q}_N(t_{N-1})=\boldsymbol{q}_{N-1}(t_{N-1}),\quad t\in[t_{N-1},t_N]\\
&\qquad \boldsymbol{W}\left(\boldsymbol{q}(t),\boldsymbol{x}\right)\leq\boldsymbol{0},\quad t\in[t_{N-1},t_N]\\
&\qquad \boldsymbol{\varepsilon}(\boldsymbol{q}_N)=\boldsymbol{q}_N(t_N)-\boldsymbol{q}_N(t_{N-1})=\boldsymbol{0}\\
&\qquad LB\leq(\boldsymbol{q}_0,\boldsymbol{x})\leq UB.
\end{aligned}\tag{3.1}$$

Here $\boldsymbol{\phi}=(\phi_1,\phi_2,\ldots,\phi_s)^T$ is the goal function vector with s design objectives such as power consumption, purity, and recovery; $\boldsymbol{F}(\cdot)$ are the PSA bed model equations; $\boldsymbol{q}$ is the complete set of spatially discretised state variables throughout the simulation, and subscript k represents the cycle number; $\boldsymbol{q}_0$ are the initial conditions for the state variables; $\boldsymbol{W}(\cdot)$ are the inequality design constraints, t_0 to t_N is the cycle duration, and LB, UB are the lower and upper bounds, respectively, of the state and design variables. The set of equations for $\boldsymbol{\varepsilon}(\boldsymbol{q}_N)$ are the CSS constraints. The optimisation problem formulation is stated as a minimisation problem, but clearly some design criteria should be maximised, such as the recovery. This can easily be achieved by using $-\phi(\cdot)$ instead of $\phi(\cdot)$ for the objectives that should be maximised.

As shown the PSA process simulation for large-scale industrial applications of PSA can be very computationally expensive. The optimisation of complex PSA cycles by searching through a wide range of possible design configurations tends to be too demanding to solve in reasonable time. Biegler et al. [BJF05] provide an overview of different strategies for approaching the optimisation of PSA cycles: black-box optimisation, complete discretisation optimisation, simultaneous tailored optimisation, and simplified optimisation. We here briefly describe these approaches.

3.1.1 Black-box optimisation

The black-box optimisation problem can be written as:

$$\begin{aligned} \underset{\boldsymbol{x}}{\arg\min} \quad & \boldsymbol{\phi}(\boldsymbol{q}(\boldsymbol{x})) \\ \text{s.t.} \quad & \boldsymbol{W}\left(\boldsymbol{q}(\boldsymbol{x})\right) \leq \mathbf{0} \\ & LB \leq \boldsymbol{x} \leq UB \end{aligned} \tag{3.2}$$

In this approach the solution of the PSA bed model, given by the vector of spatially discretised state variables $\boldsymbol{q}(\boldsymbol{x})$, is computed from solving the simulation problem (2.21) embedded in the objective function $\boldsymbol{\phi}(\cdot)$. $\boldsymbol{W}$ are the inequality constraints for the design. This means the approach is non-intrusive. Some advantages with this approach are its easiness in implementation, and that it allows us to specify objective functions that can use the full range of $\boldsymbol{q}(\boldsymbol{x})$, and not only the data for the CSS cycle $\boldsymbol{q}_N(\boldsymbol{x})$. $\boldsymbol{\phi}(\cdot)$ can be calculated from the CSS solution of the simulation problem. More precisely, we are mostly interested in the bed conditions and gas concentration profiles for the ends of the bed during the CSS cycle.

Here, the optimiser calls $\boldsymbol{\phi}(\boldsymbol{q}(\boldsymbol{x}))$ for different design points $\boldsymbol{x}$, where $\boldsymbol{q}(\cdot)$ are the bed conditions along the bed, to assist the optimiser to find a new search direction. This is however a rather naïve approach making it more computationally expensive than may be required for the specific problem. The optimisation problem is stated in such a way that the computer simulation is performed through $\boldsymbol{q}(\boldsymbol{x})$ for a given design configuration $\boldsymbol{x}$. Most work on PSA optimisation has used this traditional approach, see, e.g., [CSMM03, FFB09b, HKF$^+$11].

3.1.2 Complete discretisation optimisation

In this approach complete discretisation (cf., Section 2.3.2) is applied on (3.1). The mathematical problem (3.1) is discretised into a nonlinear system for which Newton methods are applicable. Using Newton methods is highly desired due to their quadratic convergence rate. For periodic adsorption processes, this approach first appeared in [NP98] for air separation application of a RPSA system. In the presence of sharp, moving concentration fronts in the PDAE solution, a large number of discretisation points is typically required in the temporal domain [KSB03]. This is the major bottleneck for this approach, complete discretisation relies on a fixed grid and therefore unable to adjust the temporal grid adaptively, which is a common way procedure for solving large-scale systems. The computational complexity is strongly related to the resolution of the numerical scheme, which can make this approach prohibitively expensive in a computational sense.

The complete discretisation approach has been applied to some basic PSA systems [KM02, KSB03]: a RPSA for recovery of oxygen from air, and to a single-bed 4-step PSA system with Skarstrom cycle for CO_2 separation from a binary mixture CO_2/N_2. More recently, a two-bed superstructure has been proposed and implemented for pre- and post-combustion carbon capture [ABZ10a, ABZ10b], which utilised complete discretisation.

3.1.3 Simultaneous tailored optimisation

The simultaneous tailored optimisation problem can be formulated as follows.

$$\begin{aligned} \underset{\boldsymbol{q}_0,\boldsymbol{x}}{\arg\min} \quad & \boldsymbol{\phi}(\boldsymbol{q}(\boldsymbol{q}_0,\boldsymbol{x})) \\ \text{s.t.} \quad & \boldsymbol{W}\left(\boldsymbol{q}(t,\boldsymbol{q}_0,\boldsymbol{x})\right) \leq 0 \\ & \boldsymbol{\varepsilon}(\boldsymbol{q}_0,\boldsymbol{x}) = \boldsymbol{q}_0 - \Psi\left[\boldsymbol{q}_0,\boldsymbol{x}\right] = 0 \\ & \text{LB} \leq (\boldsymbol{q}_0,\boldsymbol{x}) \leq \text{UB} \end{aligned} \tag{3.3}$$

Here the CSS conditions are posed as constraints for the optimisation problem, unlike the black-box optimisation where they were solved simultaneously with the DAEs for the PSA model when simulating $\boldsymbol{q}(\cdot)$. Another simultaneously tailored approach is the one proposed in Ding et al. [DCL02] that is expressed the same as (3.3) but without inequality constraints $\boldsymbol{W}(\cdot)$. This approach is similar to the direct determination of CSS approach in simulation. The CSS conditions are only required to hold at the optimum. In fact, the first solution we find that satisfies the CSS condition is an optimal one [DCL02]. In other words, the CSS is not reached until the optimal solution is reached. This approach is considered to be faster than black-box optimisation but at the price of less accurate CSS conditions [KSB03]. The nonlinearity of CSS conditions are also introducing some extra computational complexity. This approach was first proposed for small-scale air separation unit [JBF03], and has later been adopted for CO_2 separation from a N_2/CO_2 gas mixture [KSB03].

3.1.4 Simplified optimisation

The simplified optimisation approach uses a simplified model as a less computationally expensive substitute for the original high-fidelity PSA model in the black-box optimisation framework:

$$\begin{aligned} \arg\min_{\boldsymbol{x}} \quad & \boldsymbol{\phi}(\hat{\boldsymbol{q}}(\boldsymbol{x})) \\ \text{s.t.} \quad & \boldsymbol{W}\left(\hat{\boldsymbol{q}}(\boldsymbol{x})\right) \leq \mathbf{0} \\ & LB \leq \boldsymbol{x} \leq UB \end{aligned} \tag{3.4}$$

Here $\boldsymbol{\phi}(\cdot)$ is based on spatially discretised design variables, $\hat{\boldsymbol{q}}$, provided by the simulation of a simplified PSA model. $\hat{\boldsymbol{q}}$ is an approximation of $\boldsymbol{q}$. This approach allows a different set of optimisation methods which normally are confined to problems that are fast-to-evaluate. This optimisation problem should be much faster to solve. However the resulting solution could be deceptive, since the solution of $\arg\min_{\boldsymbol{x}} \boldsymbol{\phi}(\hat{\boldsymbol{q}}(\boldsymbol{x}))$ and $\arg\min_{\boldsymbol{x}} \boldsymbol{\phi}(\boldsymbol{q}(\boldsymbol{x}))$ could be very different if the accuracy of the simpler model is low. Even more concerning is the fact that even if the simpler model matches the more rigorous one very well on average, the convergence cannot be guaranteed unless the simpler model and the detailed model have matching gradients at *all* points, which would mean that the simpler and the detailed model are exactly same. Biegler et al. [BGW85] find a simplified model to be an appropriate substitute if also the high-fidelity model recognises that the optimum satisfies the Karush-Kuhn-Tucker (KKT) conditions, or less strictly if they have matching gradients at the optimum. The latter may occur when some stop criterion is met before the optimiser has converged to a KKT point.

Smith and Westerberg [SIW91] simplified the PSA model by using time integrated mass and energy balances that basically only capture the qualitative physical behavior, this reduced the computational complexity enough to make simple optimal design affordable. Similarly, in [FFB09b, FFB09a] a small number of continuous stirred tank reactors (CSTRs) was employed for the mass balances, together with time integrated energy balances. Agarwal, Biegler and Zitney [ABZ09] recently proposed a reduced order model (ROM), more specifically a proper orthogonal decomposition (POD) scheme, to simplify the PSA bed process for optimisation of a 2-bed/4-step PSA system for hydrogen purification from a CH_4/H_2 gas mixture. Hasan et al. [HKF$^+$11] proposed for CO_2 post-combustion the use Kriging models as low-fidelity models, built on m samples $\{(\boldsymbol{x}_i, \mathbf{y}_i)\}_{i=1}^{m}$ taken from the high-fidelity model. This was the first attempt to use Kriging surrogate models for PSA optimisation. A few publications had previously used Kriging for modular flowsheet optimisation [CG08, HM11].

In principle, the simplified optimisation strategy can be combined with any of the afromentioned optimisation strategies. By convention simplified optimisation is a one-step approach, but it can be used in an iterative procedure that occasionally compare the simpler model to the detailed one to validate and possibly further improve the simpler model around the validation points.

3.2 Optimisation methods in literature for PSA problems

Different strategies for approaching the general optimisation problem (3.1) require different optimisation procedures for solving them. Different optimisation strategies are developed for different types of computational challenges. Despite this fact, most studies for PSA design put little effort to explain or justify the choice of optimisation method. Before selecting an optimisation method, we have compiled the following list of some of the challenges that the method may face:

- Non-convexity in the objective space when mapping the objective function from the design space;
- The computational expense of the objective/constraint functions for a single evaluation;
- Objective functions with non-smoothness, non-differentiability, noisy output, or even discontinuity.
- No information available about the derivatives of the objective function;
- "Curse of Dimensionality": when computational complexity increases fast with the number of design variables, typically with an exponential rate;
- Failure to evaluate the objective function at some design points;

Another issue that should be mentioned is that the computational time to perform an optimisation algorithm can be very high. Ideally, the ratio of the algorithm time to the total time spent in the objective function evaluation should be small.

Below we have provided a brief presentation of the most popular methods used for PSA optimisation. First we will discuss optimisation for real-valued objective functions $\phi(\cdot)$, then the case when multiple objective functions are optimised simultaneously.

3.2.1 Gradient-based optimisation

Gradient-based methods start off with an initial guess $\boldsymbol{x}$ in the design space, and then select sequentially, in some search direction, a new point until the sequence of selected points converges to a local optima. Gradient-based optimisation algorithms are commonly based on steepest descent and Newton's method. Perhaps the most popular gradient-based optimiser is the Successive Quadratic Programming (SQP) method (see, e.g., [NS96]).

Successive Quadratic Programming (SQP): SQP is arguably the most popular locally convergent optimisation method for nonlinearly-constrained problems (NLP), such as (3.2). SQP is in short the application of Newton's method to the Karush-Kuhn-Tucker (KKT) conditions for nonlinear problems. Here follows a basic SQP example: consider the slack-variable formulation of the black-box problem as given by

$$\begin{aligned} \arg\min_{\boldsymbol{x},\boldsymbol{\sigma}} \quad & \phi(\boldsymbol{q}(\boldsymbol{x})) \\ \text{s.t.} \quad & \boldsymbol{W}(\boldsymbol{q}(\boldsymbol{x})) + \boldsymbol{\sigma} = \boldsymbol{0} \\ & \boldsymbol{\sigma} \geq \boldsymbol{0} \\ & LB \leq \boldsymbol{x} \leq UB \end{aligned} \tag{3.5}$$

where $\boldsymbol{\sigma} \geq \boldsymbol{0}$ are the slack variables. Then let $\mathbf{d}(\boldsymbol{x}) = \boldsymbol{x} - \boldsymbol{x}^*$. For the current point $\boldsymbol{x}^* \in \mathbb{R}^p$, the goal is to solve the following quadratic subproblem to find the next point $\boldsymbol{x}$ along the direction $\mathbf{d}(\boldsymbol{x})$,

$$\begin{aligned} \arg\min_{\boldsymbol{x},\boldsymbol{\sigma}} \quad & \phi(\boldsymbol{q}(\boldsymbol{x}^*)) + \nabla_{\boldsymbol{x}}\phi(\boldsymbol{q}(\boldsymbol{x}^*))^T\mathbf{d}(\boldsymbol{x}) + \tfrac{1}{2}\mathbf{d}(\boldsymbol{x})^T\nabla^2_{\boldsymbol{x}\boldsymbol{x}}\mathcal{L}(\boldsymbol{x}^*, \lambda^*)\mathbf{d}(\boldsymbol{x}) \\ \text{s.t.} \quad & \boldsymbol{W}(\boldsymbol{q}(\boldsymbol{x}^*)) + \nabla_{\boldsymbol{x}}\left(\boldsymbol{W}(\boldsymbol{q}(\boldsymbol{x}^*)) + \boldsymbol{\sigma}\right)^T\mathbf{d}(\boldsymbol{x}) = \boldsymbol{0} \\ & \boldsymbol{\sigma} \geq 0 \\ & LB \leq \mathbf{d} + \boldsymbol{x}^* \leq UB \end{aligned} \tag{3.6}$$

where $\mathcal{L}(\boldsymbol{x}, \lambda) = \phi(\boldsymbol{q}(\boldsymbol{x})) + \lambda^T\boldsymbol{W}(\boldsymbol{q}(\boldsymbol{x}))$ is the Lagrangian function, and $\lambda \in \mathbb{R}^p$ the Lagrange multipliers. The arguments of the multiplier vector λ are updated from one iteration to the next, using some updating scheme [NS96]. For each QP step, we solve the QP subproblem at the current $\boldsymbol{x}^*$. Under some mild assumptions, this optimisation algorithm displays a local quadratic rate of convergence like Newton's method. Whenever the "work" required to obtain the Jacobian matrix is too high, numerical approximations for the derivatives are typically introduced. Quasi-Newton methods can be deployed, downgrading the method to superlinear convergence. The use of approximations for the derivatives cause loss of accuracy and may deteriorate the optimisation performance.

In the context of PSA design, the SQP method has been used for reducing the bed length while satisfying a desired product purity [Kva97], and for lowering the power consumption of the oxygen production system at a desired product purity [CMM05]. SQP has also been used with the complete discretisation approach for air separation with RPSA [NP98].

Reduced Hessian Successive Quadratic Programming (rSQP): rSQP [BNS95] is among the current state-of-the-art for large-scale nonlinearly-constrained optimisation [BGHvBW03]. rSQP has been applied to the "simultaneous tailored" optimisation, (3.3) (see, e.g., [KSB03, JBF03, KSB05, JBF05]).

Because the variables of this large-scale nonlinear system can be divided into independent and dependent variable subsets, the QP subproblem (3.6) can be projected into the reduced space of independent variables, namely the null space of the linearised constraints. Essentially this eliminates the linearised constraints in (3.6). Also, the search directions are dealt with separately. For "simultaneous tailored" optimisation, where we have the CSS equations, this leads to a substationally reduced QP subproblem. However, due to the CSS equations, ill-conditioning in the Jacobian of the linearised equations may occur, and a dogleg step can be used to improve numerical stability [JBF03]. For the slack-variable formulation of (3.3), slack variables $\boldsymbol{\sigma}$ and initial conditions $\boldsymbol{q}_0$ are the dependent variables, which are perturbed to improve the solution of the inequality constraints, whereas the design variables $\boldsymbol{x}$ are the independent ones, which are manipulated to minimise the objective function. rSQP also displays global convergence properties when a penalty term is added to the objective function, a term proportional to the Frobenius norm of the residuals of the constraints. The code for rSQP is available in gPROMS under the name SRQPD [TB98, KSB05]. rSQP has been applied for a 5-bed/11-step PSA cycle, both in the unibed and multibed setting, to obtain high purity H_2, from a gas mixture of H_2 (73 %), N_2 (0.5%), CO_2 (16 %), CO (5 %), CH_4 (5.5 %) [JBF03, JFB04]. Also for CO_2 sequestration applications with PSA [KSB03, KSB05].

Similarly to direct determination of CSS with Newton's method, the rSQP with the "simultaneous tailored" approach, as implemented in DASPK 3.0, requires computationally expensive sensitivity calculations. When the low-order finite difference is used to calculate the sensitivities the derivatives may become noisy and causing severe deterioration of the optimisation performance [BJF05]. Because a sensitivity calculation with respect to each parameter is independent, parallisation is trivial [JBF05]. Note that the computation of the DAEs is performed at each CPU, but this computational cost is small compared to the evaluation of the sensitivities. Accurate sensitivites are reported with DASPK 3.0.

The performance of rSQP when stable, surpass the performance of the black-box approach in terms of the number of objective function evaluations required. As seen in [JBF05] with DASPK, the black-box approach needed to simulate about 1400 cycles to reach CSS, whereas the tailored approach with rSQP required about 300 cycles.

Gradient-based optimisation relies on first and/or second derivatives. The derivatives can be difficult, or even impossible, to express analytically, and thus often replaced by numerical approximations. Although, if inaccurate derivatives are employed the optimisation result can be wide of the mark and in turn very delusive. In fact, even if the PDAE solution of the mathematical model is continuous, the computational model can result in a noisy solution that is non-differentiable, which can be destructive to the efficiency of optimisation.

In conclusion, gradient-based optimisers typically exhibit impressive convergence rates with good starting guesses. On the other hand they are not always manageable: only locally convergent without adaptivity mechanisms which would spoil the rate of convergence, and have difficulty to converge in the presence of numerical noise or to tight error tolerances.

3.2.2 Derivative-free optimisation

Derivative free methods tend to require more objective function evaluations than gradient based methods to achieve similar optimisation performance (see, e.g., [CSV09]), but do not require any information about the derivatives of the objective function. This makes derivative-free methods more manageable from a computational perspective.

In [Fia10] some derivative-free methods were compared for the design of a Skarstrom PSA cycle for N_2 production from air: the Nelder-Mead Simplex method, Multidirection Search, Alternating Directions, Implicit Filtering, and Hooke-Jeeves method. The design variables considered were the cycle duration, valve constants, and the feed-gas flowrate. The objective was to maximise product recovery given the CSS conditions and a constraint on the maximal bed pressure level. The Nelder-Mead method with a multidirection search showed the most promising performance in terms of convergence. Unfortunately, it is known that Nelder-Mead methods may converge to non-optimal points even for simple test problems [Kel99], and can thus not be considered to be a reliable method.

The optimsation software BOBYQA (Bound Optimization BY Quadratic Approximation) [Pow09] has been applied in an application for CO_2 capture in IGCC power plants, with the black-box optimisation framework [DVB12]. BOBYQA is an extension to constrained optimisation of NEWUOA [Pow06] from the same author.

3.2.3 Multi-objective optimisation

For most real-world applications the optimisation problem can be formulated with a single objective function, but for some tasks a single objective cannot carry enough information to provide a definitive answer to the given problem. The design of a PSA system is a prime example where a single value rarely is enough to assess the overall performance. The two most used design criteria are the product purity and recovery, which are known to be conflicting [FFB09b]. See Figure 3.1. When solving engineering problems appearing in the real world, the overall performance is often a trade-off between technical performance attributes and economic aspects. As an example, identifying the most appropriate technology for carbon capture is one application where the actual challenge is *not* to find a technology with viable separation performance, but rather to identify an economically viable option with good enough CO_2 separation for carbon capture.

Optimisation with a single objective is conceptually straight-forward, but as soon as you add one more objective, the situation turns complex. The basic multi-objective optimisation problem can be written as:

$$\begin{aligned} \arg\min_{\boldsymbol{x}} \quad & \boldsymbol{\phi}(\boldsymbol{x}) \\ \text{s.t.} \quad & \boldsymbol{F}(\boldsymbol{x}) = \boldsymbol{0} \\ & \boldsymbol{W}(\boldsymbol{x}) \leq \boldsymbol{0} \\ & LB \leq \boldsymbol{x} \leq UB \end{aligned} \tag{3.7}$$

where $\boldsymbol{\phi}(\cdot)$ is a vector of q objective functions, that is, $\boldsymbol{\phi}(\boldsymbol{x}) = [\phi_1(\boldsymbol{x}), \phi_2(\boldsymbol{x}), \ldots, \phi_q(\boldsymbol{x})]$. Thus, the goal is to determine $\boldsymbol{x} = (x_1, x_2, \ldots, x_p)^T$, from the set of solutions satisfying the hard constraints, which yield the optimal values of $\phi_1(\boldsymbol{x})$, $\phi_2(\boldsymbol{x})$, …, $\phi_q(\boldsymbol{x})$, simultaneously. Rarely it exists a single solution $\boldsymbol{x}$ that is a global optimum to all objective functions. For this reason pure MO methods seek to find a set of "non-dominated" solutions considered to be of equivalent quality, not just to identify a single optimal solution. The objective space thus becomes partially ordered, see Definition 2.

Definition 1. *(Dominate) Let $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^q$. Then vector $\boldsymbol{a}$ is said to "dominate" $\boldsymbol{b}$, denoted by $\boldsymbol{a} \prec \boldsymbol{b}$, if and only if $a_i \leq b_i$ for all i and $a_i < b_i$ for at least one i, where $i = 1, 2, \ldots, q$.*

Definition 2. *(Non-dominated) Let a response be defined as $\boldsymbol{\phi}^\star = [\phi_1^\star, \phi_2^\star, \ldots, \phi_q^\star] \in \mathbb{R}^q$. Then, given a set of m responses, $T \in \mathbb{R}^{m,q}$, the response $\boldsymbol{\phi}^\star$ is said to be non-dominated in T, iff it* *does not exist a response* $\boldsymbol{\phi} = [\phi_1, \phi_2, \ldots, \phi_q] \in T$ *such that* $\boldsymbol{\phi} \prec \boldsymbol{\phi}^*$.

Definition 1 states that one solution dominates another if the solution is not worst for any of the objectives and better for at least one objective than the other. This actually allows more than one solution to be the "best" solution. As opposed to the wide range of methods available for single-objective optimisation, there are relatively few methods for multi-objective optimisation. To take advantage of a wide range of existing methodologies, the optimiser can be instructed, typically with a real-valued trade-off formula, how to cope with the situation of multiple and sometimes conflicting objectives within the single objective framework.

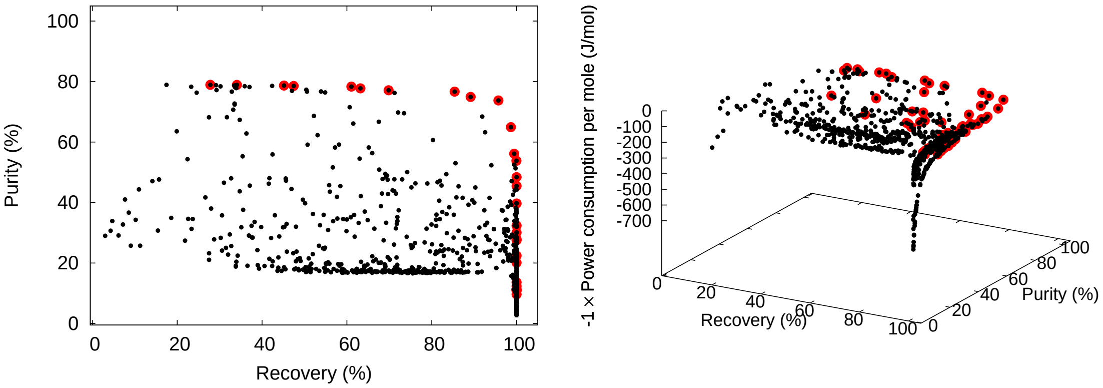


Figure 3.1: Left: 2D example of the design solutions in the objective space (Purity versus Recovery), with red dots representing non-dominated solutions. Right: 3D example of the solutions in the objective space, which also includes the Power consumption. The data used in the plot is from a 2-bed/6-step PSA system under investigation in Chapter 6.

The three most common branches of multi-objective optimisation are: multi-objective utility, goal programming, and the Pareto approach.

Multi-objective utility approach: The utility approach defines a new scalar objective function $\phi^*(\cdot) : \mathbb{R}^q \to \mathbb{R}$, which incoporates the values of all objectives of interest, to represent a compromise between the objectives. This approach "scalarises" the multi objective problem into a single objective one. The most common choice is perhaps the Weighted Aggregation [MA10]:

$$\phi^*(\boldsymbol{\phi}(\boldsymbol{x})) = \sum_{i=1}^{q} \alpha_i \phi_i(\boldsymbol{x}), \tag{3.8}$$

where the coefficients $\alpha_i \geq 0$ represent the weighting between the values of the objectives, $\phi_i(\boldsymbol{x})$. Expert knowledge and elicitation can be used to choose the weights $\{\alpha_i\}$.

A drawback with weighted aggregation is that some of the optima are unreachable if the feasible region in the objective space is non-convex [Deb01], for example of a concave shape. However, non-linear utility functions may not present such limitations [AJ05]. Common practice is to perform a few different configurations $\{\alpha_i\}$, either pre-generated or adaptively chosen during the optimisation. A set of uniformly distributed weights cannot guarantee a uniform distribution of points in the objective space [Das97]. Actually, the points tend to lump in certain regions of the objective space.

Goal programming (GP) approach: Goal programming is an lexicographic approach that turns the multi objective optimisation problem into a single objective one with additional inequality constraints. It can be seen as generalisation of linear programming to deal with multiple conflicting objective functions. It is assumed the designer provides a preference system (objectives are ordered by importance) with specified target values to be achieved. Perhaps the most popular is the ε-constraint method, first introduced in [CH83]. The ε-constraint method is optimising one objective while specifying target values (expressed as upper bounds) for the remaining ones, that is,

$$\phi_i(x) \leq \varepsilon_i,$$

for $i = 2, \ldots, q$. The main advantage with this approach is that we only need to solve a single objective problem, and the additional inequality constraints can naturally be dealt with using mathematical programming methods. Also, this method is able to identify solutions on the non-convex boundary of the feasible region that are not possible to reach using the traditional weighted aggregation approach - when the target values are specified suitably. Several solutions can be obtained by solving a sequence of ε-constrained single objective optimisation problems. A drawback is that some design objectives cannot be expressed as hard constraints in a satisfying way. The prior selection of ε is also a challenge to ensure feasible solutions. If the ε values are chosen too pessimistically, the problem formulation becomes infeasible. The GP approach is computationally more complex than the utility approach. In fact, the computational complexity of the ϵ-constraint method grows exponentially with respect to the number of objectives [LTZ06]. For an overview on goal programming methods we refer to [MDC98].

Pareto approach: The trade-off between design objectives cannot always be formalized explicitly in an adaquate way, which motivates a more general take on optimality. Instead of reducing the problem to a single objective formulation, the Pareto approach can be employed [BDMS08]. The term Pareto optimality, named after the Italian economist Vilfredo Pareto, is taken from the literature of Economic Theory: when resources have been allocated to a population, no individual can be made better off without making at least one of the other individuals worse off.

This approach, when applied in engineering design, does not search for a single best design but rather for a set of non-dominated solutions. The set of non-dominated solutions $\{\boldsymbol{\phi}(\boldsymbol{x}_i)\}$ is called the "Pareto front", and the corresponding design points $\{\boldsymbol{x}_i\}$ the "Pareto set." This non-parametric approach avoids the need to *a priori* specify a trade-off between competing objectives by identifying the Pareto front. By convention a Pareto front is presented as a curve (or hyper-surface) in the objective space. The goal of Pareto optimisation is to improve the informativeness of these non-dominated solutions. The drawbacks are mainly computational, the approach is rather exhaustive and tends to require a substantial number of evaluations of $\boldsymbol{\phi}(\cdot)$ to achieve a "good" Pareto front. In practice only an approximation of the true Pareto front is affordable. A Pareto front could be nonlinear, or even discontinuous. Hence, with a sparse Pareto front approximation it is difficult to draw any conclusions about the actual shape of the true Pareto front.

Here follows an example where the Pareto approach can be considered excessive: lets say we have been asked to minimise $\boldsymbol{\phi} = (\phi_1(\boldsymbol{x}), \phi_2(\boldsymbol{x}))^T$ with $\phi_1(\boldsymbol{x}) = \sqrt{(x_1-1)^2+(x_2-1)^2}$ and $\phi_2(\boldsymbol{x}) = \sqrt{(x_1+1)^2+(x_2+1)^2}$, for $\boldsymbol{x} = (x_1, x_2)^T \in [-2, 2]^2$. However, the designer (or decision maker) may have some design preferences, for instance, the point on the true Pareto front nearest the origin of coordinates in the objective space might be the one of most interest. This corresponds to the Utility approach with $\phi^*(\boldsymbol{\phi}(\boldsymbol{x})) = \sqrt{\phi_1(\boldsymbol{x})^2 + \phi_2(\boldsymbol{x})^2}$. The Pareto approach, when handed a small computational budget, may not afford to compute an approximation of the Pareto front with enough spread to cover the region of the Pareto front nearest the point of origin of coordinates.

Even though this approach is computationally expensive it has an unmatched ability to capture the non-dominated solution landscape. Also, it is an approach relatively easy to apply when there are no guidelines available on how the scaling/trade-off between the objectives should be made for a given application.

In summary, the Utility and GP approaches are often applied as a parametric down-scaling to avoid excessive use of computational resources. Also, because they are reducing the problem to a single-objective formulation, advantage can be taken of a wide range existing methods. However, for efficient use you should have some knowledge about the shape of the Pareto front. The Pareto approach, on the other hand, is an a posteriori approach that tends to be more informative and robust, but comes at the price of a higher computational cost. The Pareto approach gains momentum if the underlying computational model is cheap-to-evaluate. In the following chapters we will show that even "expensive" computational models for PSA simulation can be considered within the Pareto setting, if assisted by so-called surrogate models. Surrogate model techniques are introduced in Chapter 4.

The Pareto front can be very useful as decision support, even in the case when the decision maker already has preferences between conflicting objectives. The final decision can be postponed until a rich set of non-dominated solutions is generated. The decision maker is then in a position to use unformalisable knowledge such as intuition and expert knowledge to be able to select a smaller set of these non-dominated solutions, perhaps of size one, for further consideration. This aspect of the decision process is important to recognize in order to understand the usefulness of this approach.

The Pareto approach has become increasingly popular over the last decade, much thanks to the rapid increase in computer power, availibility of computer code packages, and easier access to high performance computer resources.

Some optimisation methods are specialised in generating Pareto solutions for problems with multiple objectives, adopting Pareto-based ranking schemes. The most popular class of such methods are Evolutionary Multi-Objective (EMO) algorithms: VEGA (Vector Evaluated Genetic Algorithm; [Sch85]); NPGA (The Niched Pareto Genetic Algorithm; [HNG94]); NSGA-II (The Nondominated Sorting Genetic Algorithm II; [DAPM00]); SPEA2 (The Strength Pareto Evolutionary Algorithm 2; [ZT99]); and MOGA (Multiobjective Genetic Algorithm; [FF93]). Other significant MO approaches are SMS-EMOA (S-metric selection evolutionary multiobjective optimisation algorithms; [BNE07]), MOSA (multi objective simulated annealing; [SSPC00]), AMOSA (archived multi objective simulated annealing; [BSMD08]), MOPSO (Multiple objective particle swarm optimisation; [CCL02, MS14]), NSDE (Nondominated Sorting Differential Evolution).

The main challenge for multi-objective optimisation is to drive the search, in an efficient way, so that the distance between the generated Pareto solutions and the true Pareto front is minimised, while achieving diversity in the Pareto solutions generated. The characteristics of a good Pareto front approximation are that it should be in close proximity to the true Pareto front in some distance metric, and ideally achieve the maximal spread, as well as good distribution of points in terms of spacing, coverage and richness. A variety of performance indicators have been suggested for the comparison of Pareto solutions [KC02]. The spacing of the Pareto solutions can be taken as the average interpoint distance between consecutive points of the ordered Pareto front; the coverage can be quantified by partitioning the objective space and then observe the ratio of partitions represented by the Pareto solutions, and the richness by the number of points belonging to the Pareto front. The spacing, coverage and richness indicators can be misleading as they are only weakly informative about the quality of a Pareto front. The "distance" between the Pareto front approximation and the true Pareto front is often seen as a good quality measure but needs to be defined appropriately. The most common example of a distance metric is the average distance from the points of the Pareto front to the nearest point of the true Pareto front [BDMS08], where the distance metric is the normalised Euclidean distance.

In the same spirit, to assert that a Pareto front approximation, lets call it P_1, is *better* than another approximation P_2, in an absolute sense, there must exist for each $p_2 \in P_2$, a point $p_1 \in P_2$ that dominates p_2. Only in exceptional circumstances will a set P_1 dominate a set P_2 in such a way. Another approach is to compare two Pareto set approximations in a probabilistic manner, that is, "What is the probability that P_1 is better than P_2 over a family of utility functions?" For more on the comparison of Pareto front approximations we refer the interested reader to [HJ98].

For engineering design applications, such as the PSA case with product purity-recovery, the true Pareto front is not known, otherwise what would be the point to perform optimisation in the first place? Hence, in practice, the optimal Pareto front cannot be used as a reference solution. A more practical performance indicator is instead the S-metric [ZT99], which measures the "dominated hypervolume of non-dominated solutions", henceforth denoted by $S(P)$, which is the volume spanned by the Pareto front, P, and some specified reference point (see Figure 3.2).

Definition 3. *The S-metric for Pareto front P, consisting of vectors $\phi \in \mathbb{R}^q$, can formally be*

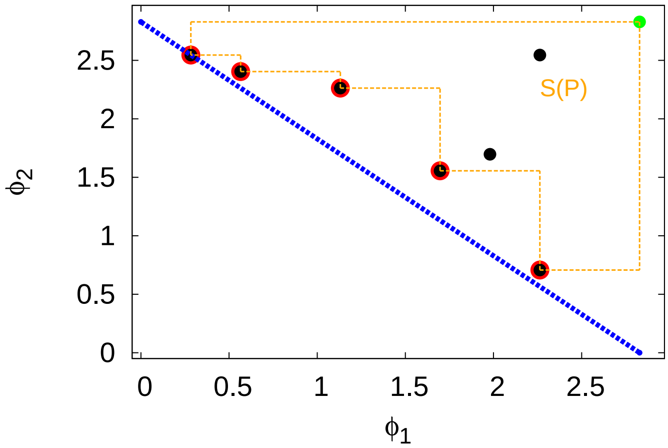


Figure 3.2: Minimise $\boldsymbol{\phi} = \left(\sqrt{(x_1-1)^2+(x_2-1)^2}, \sqrt{(x_1+1)^2+(x_2+1)^2}\right)^T$ over support $\boldsymbol{x} = (x_1, x_2)^T \in [-2,2]^2$. An example of a Pareto front approximation, and illustration of the hypervolume indicator S(P) of the Pareto front, P. The Pareto front P consists of the nondominated solutions, which are marked in red. The true Pareto front (blue dashed-line) is the hyperline segment from $(0,2\sqrt{2})^T$ to $(2\sqrt{2},0)^T$. The reference point for S(P) is marked in green, and located at $(1.8\sqrt{2}, 1.8\sqrt{2})^T$.

written as:

$$S(P) = \lambda\left(\bigcup_{\boldsymbol{\phi}\in P}\{\tilde{\boldsymbol{\phi}} \in \mathbb{R}^q | \boldsymbol{\phi} \prec \tilde{\boldsymbol{\phi}} \prec \boldsymbol{\phi}_{ref}\}\right), \tag{3.9}$$

where $\lambda(\cdot)$ denotes the Lebesgue measure. Recall, the operator $\boldsymbol{\phi} \prec \boldsymbol{\phi}'$ means that $\boldsymbol{\phi}$ dominates $\boldsymbol{\phi}'$.

Here $\boldsymbol{\phi}_{ref}$ is the reference point and is chosen such that it is dominated by *all* the points belonging to the Pareto front. The S-metric can be used for the comparison of two Pareto approximations, S(P$_1$) against S(P$_2$). Also, the metric can be used to monitor the increase in S(P) as a new point is added to the Pareto front P, that is, $\Delta^S(\boldsymbol{x}^*) = S(P \cup \phi(\boldsymbol{x}^*)) - S(P)$. There exist multi-objective methods where the goal is to maximise S(P), the dominated hypervolume. For example SMS-EMOA [BNE07] and MOEGO [EDK11]. Note that the hypervolume of the Pareto front approximation should converge to the hypervolume of the true Pareto front, if calculated with respect to the same reference point.

PSA cycles are assessed on conflicting performance criteria, but only a handful of studies have applied multi-objective optimisation for PSA applications: aJG (Jumping Gene adaptations) of the multi objective simulated annealing (MOSA-aJG) [SG07], a modified version of MOGA [FFB09b, Fia10], NSGA-II [Fia10, HMN$^+$13], and more recently, NSGA-II and NSDE for air separation with a simplified PSA model in the form of a support vector regression model [LS13]. Observe that all except MOSA-aJG are evolutionary multi-objective optimisation (EMO) methods. This is not surprising, EMOs are the most popular heuristics for solving complex optimisation problems with multiple objectives [AJ05]. In our work we will use genetic algorithms (GAs), which is a special group of EAs.

Due to the popularity of GAs, they have been applied in a wide range of applications, including process engineering. The most popular multi-objective GAs are NSGA-II, SPEA2, and MOGA. Some examples of their application are Sarkar and Modak [SMM03], Atiquzzaman et al. [ALY06], Ponsich et al. [PAPDP08], and Bandyopadhyay et al. [BB13].

3.3 Evolutionary algorithms

Evolutionary algorithms (EAs) are heuristics that apply the concept of natural selection (survival of the fittest) for solving complex optimisation problems [AJ05]. Evolutionary algorithms for single-objective optimisation have existed for several decades [SGK05]. But with Schaffer's Vector Evaluated Genetic Algorithm (VEGA) [Sch85] the researchers first noticed that evolutionary algorithms can solve multi-objective optimisation in a natural way - consistent with the Pareto approach. VEGA is the first EA implementation for problems with multiple objectives. EAs are suitable for multi-objective optimisation because they deal with populations of solutions.

EAs have over the last two decades become increasingly popular for multi-objective optimisation tasks [CLVV07]. EAs are population based, and the principal idea is to evolve a population of candidate solutions, pressured by natural selection over many generations, into a future population with improved fitness. This procedure is exercised iteratively, each iteration is initialized with the population generated in the previous iteration, with the exception of the first iteration for which an initial population is generated externally. Iterations are often referred to as "generations."

Each generation begins with a reproduction process, applied on the population, that perturbs and combines candidate solutions to generate new solutions (individuals). A natural selection is then taking place that determines which individuals of the current population that will participate in the new population. This new population is the one used to initialize the next generation. This procedure is performed iteratively, for a given number of generations, or until some stop criteria is met, see Algorithm 1. EAs are considered easy to implement, and embarrassingly parallel. Moreover, they are derivative free.

Algorithm 1 Basic procedure of evolutionary computation

```
Initialization: Generate the initial population over the design space.
Evaluation: Evaluate each individual of the initial population and perform fitness assignment.
repeat
    Rank selection: Rank the individuals by "fitness" level, and assign a higher probability of being selected to individuals of higher rank. The individuals then selected, in a random manner, are called parents.
    Recombination: Randomly combine parts of selected individuals, two or more, to generate new solutions, named offsprings.
    Mutation: Randomly alter the values of one or more parts of the offsprings.
    Replacement: Replace the parental population with the generated offsprings using the biological evolution (selection, recombination, and mutation.
until Stop criteria are met
```

Suppose that the individuals of a population are seen as chromosomes, then we are in the regime of genetic algorithms (GAs), a sub category of EAs.

A chromosome is long, stringy aggregate of genes that carries DNA and associated proteins. The DNA molecules themselves contain the genetic information. Each gene, composed of several alleles, is uniquely located on the chromosome, these positions are called loci. In the simple GAs the alleles are encoded with discrete values.

Genetic recombination is the process by which chromosomes exchange genetic information. The alleles at different loci become shuffled between chromosomes (chromosomal crossover), leading to new combinations of alleles, also known as recombinant chromosomes or offsprings. Moreover, sometimes a genetic mutation occurs that changes the coding for one or more genes.

NSGA-II [DAPM00], SPEA2 [ZT99], and MOGA [FF93], are among the most applied multi-objective GAs. All of these methods have performed well for a variety of case examples, and we have selected NSGA-II to be the preferred method, mostly because of some nice features unique to NSGA-II, as well as our own experience has been positive when using NSGA-II. We have implemented our own in-house code for real-coded GA, as well as NSGA-II. Here follows a little introduction to NSGA-II.

3.3.1 The Nondominated Sorting Genetic Algorithm II

Deb et al. [DPAM02] proposed a second generation of the original NSGA [SD94], called NSGA-II. NSGA-II is a non-dominated sorting based multi-objective evolutionary algorithm.

NSGA-II is more computationally efficient, and it uses parameter free crowded-comparison operator and elitism to attain diversity. The elitist mechanism is combining the best parents with the best offsprings, as in NSGA. The approach is made up on the idea that individuals are classified on several layers. First, note that NSGA-II is an EA, and follows Algorithm 1. In the first instance, an initial population is created, typically at random. The population is ranked according to non-domination: first, all non-dominated individuals are collected into a single category, in which all individuals are assigned the same fitness value, equal to the category number (1 is the best, 2 second best, and so on). This group of individuals in category 1 is put aside, and the procedure is repeated for the non-dominated individuals of the remaining part of the population, for category 2, and so on. This leads to several layers of non-dominated individuals, called fronts, see Figure 3.3.

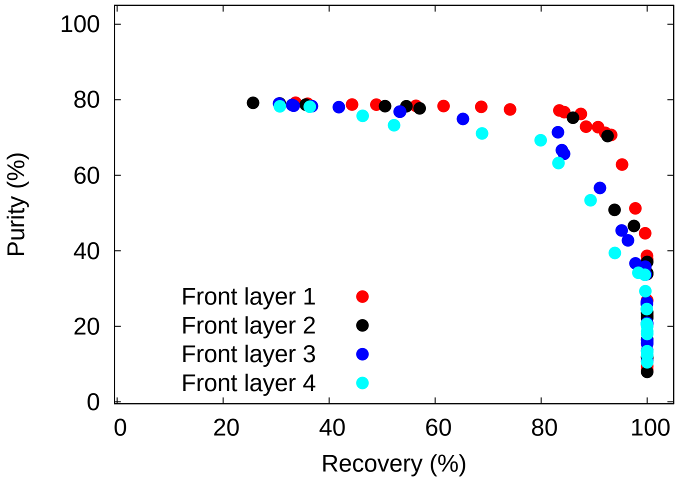


Figure 3.3: An illustration of the NSGA-II layers of non-dominated solutions. The data is taken from a NSGA-II-based optimisation, where purity and recovery are maximised simultaneously. The first layer is the Pareto front, and second layer the Pareto front of the solutions when the solutions of the first layer have been excluded, and so on.

The fitness of an individual is thus determined first by the category number it belongs to, and secondly, by its ranking within the group of individuals in the category. The ranking with the group is achieved by sorting the individuals with the crowded-comparison operator. This sorting mechanism has been shown to converge to regions of non-dominated solutions, as it maintains diversity and achieves well-spread fronts. As a GA, it then generates offsprings with crossover and mutation, and the next generation is selected according to the non-dominated sorting described above. Typically N offsprings are generated, and then the N best individuals are selected from the combined population of parents and offsprings, where N is the population size of the initial population. The original implementation uses the real-coded GA, with simulated binary crossover and polynomial mutation [DG96], which requires a few parameters to be specified: number of generations to be performed, population size, crossover rate, crossover index, mutation rate, and mutation index. The crowded-comparison operator, given in [DPAM02], is accounting for all the individual distance values between the individuals' objectives, with two boundary points included (to improve the spread of the fronts), when ranking individuals in a group. An individual is considered better than another if the following is satisfied: if the individual has a better non-dominated rank (category number), or if with same non-dominated rank but higher crowded-comparison value.

3.4 Discussion and concluding remarks

As shown, to perform computationally efficient, reliable PSA optimisation is difficult and poses several challenges. There is a need for faster and more stable optimisation algorithms, which can utilise the progress made in PSA simulation, in particular acceleration techniques to speed up the convergence to CSS. The tailored approach, which requires the calculation of sensitivities, which for large-scale problems requires a prohibitively high memory demand. The POD-based reduced-order model approach that does not require gradient information.

These optimization strategies also lead to the following future research directions:

- Optimisation of larger and more complicated PSA cycles.
- Extension to optimal operation of PSA systems using non-linear process control.
- Faster algorithms that do not require explicit calculation of Jacobian through sensitivity calculations. To avoid the computational bottleneck of sensitivity calculations adjoint approaches can be considered, and new optimisation algorithms need to be designed.

It is important to find an optimisation strategy that can utilise the progress already made in PSA simulation and accelerated successive substitution to CSS. The surrogate-based approach can be retrofitted with simulation techniques already existing for PSA processes.

Perhaps the most pressing issues are related to computational limitations. PSA models are solved under demanding computational conditions: sharp moving fronts, nonlinearity, and some level of noise due to the numerical methods. There is a need for robust optimisation procedures for computationally expensive objective functions and constraints. Although these issues have long been known, little efforts have been made to address them. In particular multi-objective optimisation for PSA has somewhat been overlooked.

The design problem becomes much more computationally demanding when the PSA model is used in a carbon capture unit, and becomes a key component in a model power plant with carbon capture. The economic implications of integrating a carbon capture unit in a coal-fired power plant is very interesting, and it cannot be fully investigated without a large-scale optimisation of the whole power plant with the carbon capture unit included, in this case PSA. One challenge is that during the design stage the PSA and the plant model react with each other, which changes the operation conditions (e.g., feed gas composition and pressure) over time. Because the PSA model is expected to be the most expensive sub-model in the whole plant model, we are interested in replacing the PSA model with an approximation model, known as a surrogate model. This will lead to significant computational savings as the surrogate models typically take less than a second to evaluate, whereas the original model can take hours. The analysis and tailoring of surrogate-based optimisation for PSA is the main contribution of this work. Real-coded GA and NSGA-II have gained much popularity over the past decade, and are robust and efficient, especially when we are able to afford a large number of objective function evaluations. For this reason, we find them to be suitable for the internal optimisation within the Surrogate-based Optimisation framework.

Chapter 4

Surrogate Models for Optimisation of PSA Systems

4.1 Motivation

Because of the detailed mathematical models required to describe the process behaviour inside a PSA adsorbent bed, many studies have been devoted to the development of simulation and optimisation strategies useful for PSA design [BJF05]. Chapter 3 covered this topic, and identified some of the computational challenges still open which should be addressed:

- The computational models used for high-fidelity simulation of PSA processes are often too computationally expensive for various analysis, such as parametric analysis, sensitivity analysis, and visualisation.
- Optimisation strategies are often simplified to save computational time. Instead of using the black-box optimisation, which perhaps is the most natural formulation of the PSA problem, other strategies have been proposed that are less costly, but can be unstable and lead to inaccurate representation. Most of these new simulation and optimisation strategies are adopting direct determination of cyclic stead state (CSS). Hence the transient process behaviour from startup to CSS is often lost.
- The ε-constraint method is often applied to the PSA optimisation problem with multiple objectives. This approach transforms the multi-objective problem into a single-objective problem that optimises one of the objectives while the remaining ones are expressed as inequality constraints. As an example, when the product purity and recovery are our objectives the ε-constraint method treats one of them as an objective and the other as an inequality constraint. On the other hand, the Pareto approach, which is generating non-dominated solutions, is more informative but computationally costly. Multi-objective

PSA optimisation has been somewhat overlooked.

- Gradient-based optimisation methods have been shown to be unstable when the response of the computational model is noisy, typically due to the presence of numerical error.

It can be concluded that efficient, robust, and reliable PSA optimisation is difficult to achieve with the current state-of-the-art. There is a need for faster optimisation algorithms that are stable under noise, and avoid computations suffering from instability, such as the sensitivity calculations performed in "tailored optimisation" for explicit Jacobian calculation. Ideally, we should also be able to utilise existing techniques for PSA simulation, especially those accelerating the convergence to CSS. Another reason to avoid the calculation of sensitivities is the prohibitively high memory demand for large-scale equation systems. Adjoint approaches have been used to reduce the computational cost associated with the sensitivities, but are introducing their own computational issues.

Even relatively fast PSA simulations are too time-consuming for the purpose of real-life PSA design. Jiang et al. [JFB04] reported a CPU time of 50-200 hours on a 2.4 GHz linux machine for a 5-bed/11-step PSA process optimisation to maximise hydrogen recovery. Multi-objective optimisation of a simple single-bed PSA process for air drying required 720 CPU hours on a 2.99 GHz Pentium IV machine [SG07].

This gives a strong motivation to develop strategies to deal with optimisation for PSA simulations. These issues are known [BJF05, ABZ09, Aga10], but little effort has been made to address this for general PSA optimisation. Here is a summary of efforts made to assist simulation and optimisation by reducing the computational cost associated with the PSA model:

- **O.J., Smith and A.W., Westerberg [SIW91]:** Model reduction with time integrated mass and energy balances. This type of low-fidelity model is faster making optimal design affordable, at the expense of using simplified models.
- **G., Fiandaca et al. [FFB09b, FFB09a]:** Model reduction with continuous stirred tank reactors (CSTRs), and a time integrated energy balance. Similar to the point above, modelling equations are greatly simplified with the use of CSTRs.
- **A., Agarwal et al. [ABZ09]:** Simplified optimisation with a reduced-order model generated by a proper orthogonal decomposition (POD) scheme. A POD scheme is used to reduce the computational complexity of the computational PSA model. Over the past decade POD has been extensively used in fluid dynamics for obtaining low-dimensional models of large spatially-distributed models [RCM04, Vol08]. This approach was tested

for hydrogen purification using a 2-bed/4-step PSA system. Truncated POD expansions are replacing the state variables in the computational model. The POD expansions consist of a linear combination of basis functions, with coefficients fitted with Galerkin projection. This approach is thus based on the physics via the PDAE model of the specific problem. Following this, the POD-based model is built around a single design solution at CSS. In this case study 5 basis functions were generated to represent a computational model with an initial discretisation level of 35 nodes. The original system consisted of 2800 DAEs, and the resulting system 200 DAEs. This is a significant speed up. It is difficult to assess the accuracy of this one-shot approximation approach. It cannot ensure a good approximation quality far from the neighbourhood of the point where the reduced-order model is set [Aga10]. The optimiser is restricted by tight bounds, because with relaxed bounds the reduced-order model failed to adequately describe the physics [Aga10]. A trust region approach was thus employed, that is, the next search point is selected by solving the corresponding optimisation problem on a trust region (tight bounds) iteratively.

- **F.M.M., Hasan et al. [HKF$^+$11]:** Simplified optimisation with a popular surrogate model technique known as Kriging. The Kriging model is “trained” on m distinct training points, each training point requires a single run of the PSA simulation model. A training point represents a single input-response relationship, and consists of the response of the model for a given design configuration. This was the first use of Kriging for PSA optimisation, but some earlier publications had used Kriging for modular flowsheet optimisation [CG08, HM11].

To overcome the major challenges related to the high computational expense of PSA simulation we propose “Surrogate-based Optimisation” using the surrogate model known as Kriging. The analysis and tailoring of surrogate-based optimisation for PSA is the main contribution of this work.

Surrogate models are fast-to-evaluate approximations built on samples of the input-response data of the PSA simulation. Surrogate models “trained” on hundreds of design points can take milliseconds to evaluate per design configuration, in contrast to the original PSA simulation that may take hours or even days. This makes surrogate models practical, and suitable in simplified optimisation.

The next section will describe the concept of “surrogate-based optimisation”, and then we illustrate the use of this surrogate-based procedure on a highly-relevant case example: a total reflux dual-piston PSA system. The next chapter will extend the work of this chapter to multiple objectives.

4.2 Surrogate modelling

A surrogate model for a computer model is a fast-to-evaluate approximative model of the model’s response, $y(\boldsymbol{x}) : \mathbb{R}^p \to \mathbb{R}$, where $\mathcal{X} \subset \mathbb{R}^p$ is the design space, p is the number of design variables. Surrogate models, also known as emulators, metamodels, and response surfaces, can be implemented in a variety of ways to allow a more efficient use of already existing methodologies for the analysis of computer models. The use of surrogate models is meaningful when a single evaluation of the computer model is very time consuming [FK09]. Many computer-aided analysis tools are only affordable for computationally inexpensive computer models, but with fast surrogates these tools may even be utilised in supercomputer applications. Surrogate models are frequently used in optimisation, visualisation, parametric studies, and global sensitivity analysis.

Global optimisation is one of many fields for which surrogates now belong to the state-of-the-art. For optimisation the most straight-forward approach is to just substitute the costly fitness and constraint functions with fast surrogates [ONK03, EN04]. Another field is global sensitivity analysis. Global sensitivity is concerned with quantifying the variation in the model output with respect to variations in the input; for example with Sobol’ indices as sensitivity measures [Sob01]. See [BBMF13] for a case example where sensitivity analysis is combined with surrogates to study the impact of stream impurities on CO_2 pipeline failures in the transportation within the CCS chain.

A surrogate model is often non intrusive, as it treats the original computer model as a black-box, hence no modification of the computer model is necessary. The first step is to select a surrogate model type, then the surrogate is fitted to some training data. The training data, denoted by D, is a set of known design points $\left\{(\boldsymbol{x}_j, y(\boldsymbol{x}_j))^T\right\}_{j=1}^{m} = \left\{\boldsymbol{X}^D, \boldsymbol{Y}^D\right\}$. This allows the surrogate model to predict the response of the underlying computer model for untried points in the design space $\mathcal{X}$. For most surrogates the training data is the only input needed to build the surrogate model. The accuracy of the surrogate depends on several factors: the dimension of the design space, the traning data in terms of size and coverage of the design space, and the smoothness of the underlying computer model response, as well as the presence of noise. Moreover, for the surrogate approach to be meaningful the computational cost of the computer model has to be high. For some computer models the response for a given design point exhibits some randomness, but this special case will not be considered in this work.

For computer models with multiple outputs a surrogate model is typically constructed for each output seperately, hence the computational cost increases linearly with the number of outputs considered.

Most surrogates can be written in the following form:

$$\hat{y}(\boldsymbol{x}) = \sum_{k=1}^{q} \beta_k h_k(\boldsymbol{x}) + \epsilon(\boldsymbol{x}). \tag{4.1}$$

Here $\hat{y}(\boldsymbol{x})$ is the computer model output at design input $\boldsymbol{x}$, and is equal to a sum of q regressors, $\{h_k(\cdot)\}$, with real-valued coefficients, $\{\beta_k\}$, and a residual random process, $\epsilon(\cdot)$.

Among the most popular surrogate techniques we find Polynomial response surface, Kriging, Radial Basis Functions (RBF), Artificial Neural Networks (ANN), Multivariate Adaptive Regression Splines (MARS), and Support Vector Regression (SVR) [FK09]. It is hardly possible to know a priori what surrogate model technique that will deliver the best approximative model (in some norm) of the computer model.

4.3 Surrogate models in optimisation

In any optimisation procedure we can in principle substitute the computationally expensive computer model with a much cheaper surrogate model (see Figure 4.1). The robustness of this direct approach depends on the surrogate model's ability to mimic the shape of the high-fidelity model's response surface. Even if the surrogate model can approximate the computer model sufficiently on average over the design space, we still need to perform point-wise validation against the computer model to gain confidence in the design solution. Furthermore, even if the surrogate response would match well with the computer model for the solution obtained with optimisation on surrogates, there still may be doubt if this solution coincides with the solution to the original design problem. To circumvent this uncertainty, surrogate models are used to assist and guide the optimiser. This approach is often referred to as "Surrogate Based Optimisation (SBO)" (see Figure 4.3).

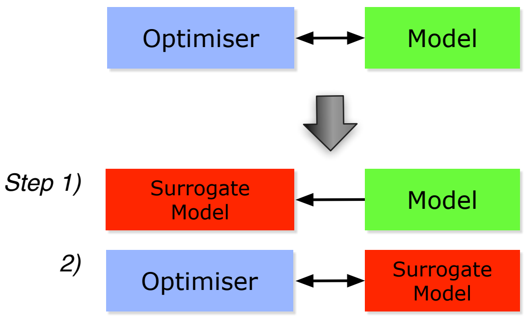


Figure 4.1: Direct approach with surrogate models

SBO is an optimisation strategy that in an iterative manner optimises some chosen design criterion based on the surrogate of the original computer model output. For a given design configuration, the criterion value may not only depend on the surrogate's prediction, but also on our confidence in this prediction. At the end of each iteration, the design to the best solution is evaluated using the original computer model, and the training data supplied to the surrogate is updated with this new design point. The most popular surrogate model type for optimisation is the so-called Kriging, and will be described in detail in the following section.

4.4 Kriging surrogate model

Kriging is a very popular statistical approach that treats $y(\boldsymbol{x})$ as an unknown function, except for the design points that already are known. It has been applied for approximating deterministic computer simulations in areas such as mechanical engineering, aerospace engineering, chemical engineering, and design optimisation. Kriging is an interpolation method, which can be labelled as a generalised regression [Gol62] as it accounts for the correlation between the error residuals. In other words, the approximation errors are assumed to be correlated in the design space. Kriging originated in engineering from the work of Danie Krige [Kri51], a South African minner engineer, and later formulated by the French mathematician Georges Matheron [Mat63], the founder of geostatistics. The idea of Kriging is to model a random field from observations on a number of spatial sites for the purpose of making predictions at unobserved sites in close proximity to the observations. Sacks et al. [SWMW89] proposed Kriging for approximating deterministic computer experiments with high-dimensional data.

Formally, the goal of Kriging is to infer $y(\cdot)$ over the design space, $\mathcal{X}$, from training data consisting of m design points, and predict the output at untried design points $\boldsymbol{x}^* \in \mathcal{X}$. Following the strategy given in [SWMW89], the quantity of interest is modelled by a Gaussian random field, denoted by $Y(\cdot)$, with $E\left(Y(\boldsymbol{x})^2\right) < \infty$ for all $\boldsymbol{x} \in \mathcal{X}$. The resulting surrogate model is the *Kriging*, also known as the Gaussian Process (GP) model. The ordinary Kriging is considered in this work, which is characterised by a constant mean and covariance structure $\Sigma(\boldsymbol{x}, \boldsymbol{x}') = \sigma^2 K(\boldsymbol{x}, \boldsymbol{x}')$ for $\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}$, where $K(\cdot, \cdot)$ is the correlation function. There is variety of functions that can be deployed for describing the spatial correlation [RW06], with the most popular being the squared exponential (SE) correlation function [RW06],

$$K(\boldsymbol{x}, \boldsymbol{x}') = \exp\left\{-\sum_{i=1}^{p} \frac{(x_i - x_i')^2}{\xi_i^2}\right\}, \tag{4.2}$$

with hyper-parameters $\boldsymbol{\xi} = (\xi_1, \xi_2, \ldots, \xi_p)$. Here σ^2 is a positive constant that represents the process variance. Other popular correlation structures are the power exponential function and the Matèrn function [Ste99]. Both examples are generalizations of SE and therefore allow more realistic covariance, although at the expense of higher computational complexity and data demand. The SE correlation function (4.2) is stationary, which means that the smoothness of $y(\cdot)$ is assumed to be the same for all regions of $\mathcal{X}$. This assumption typically oversimplifies the correlation structure for computer model outputs, but yet it tends to perform well. Also, when only a small training data is used, the modelling of a non-stationary correlation is difficult, because better coverage of the design space is required to ensure all regions are well-representated by the data.

$\boldsymbol{Y}^D$, consisting of m design points, is assumed to follow the multivariate normal distribution $\mathcal{N}_m(\beta, \boldsymbol{\Sigma})$. Here β is a constant, and $\boldsymbol{\Sigma}$ is the $m \times m$ variance-covariance matrix with the (j,k)th element given by $\Sigma(\boldsymbol{x}_j, \boldsymbol{x}_k)$ for $\boldsymbol{x}_j, \boldsymbol{x}_k \in \boldsymbol{X}^D$. Let us define the correlation matrix as $\boldsymbol{K} = \sigma^{-2}\boldsymbol{\Sigma}$.

Given the training data, $\boldsymbol{Y}^D$, and covariance parameter values, $\boldsymbol{\theta} = (\sigma^2, \boldsymbol{\xi})^T$, the inference of $y(\cdot)$ is made by the conditional posterior distribution of $Y(\cdot)|\boldsymbol{Y}^D, \boldsymbol{\theta}$ for all $\boldsymbol{x}$. It was shown in [HS93] that this conditional posterior distribution is a shifted Student's t-distribution on $m-1$ degrees of freedom (dof) with mean

$$\hat{y}(\boldsymbol{x}) = \hat{\beta} + \boldsymbol{k}(\boldsymbol{x})\boldsymbol{K}^{-1}(\boldsymbol{Y}^D - \mathbf{1}\hat{\beta}), \tag{4.3}$$

and variance

$$\hat{s}^2(\boldsymbol{x}) = \sigma^2\left\{K(\boldsymbol{x}, \boldsymbol{x}) - \boldsymbol{k}^T(\boldsymbol{x})\boldsymbol{K}^{-1}\boldsymbol{k}(\boldsymbol{x}) + \gamma^T(\boldsymbol{x})\left(\mathbf{1}^T\boldsymbol{K}^{-1}\mathbf{1}\right)^{-1}\gamma(\boldsymbol{x})\right\}, \tag{4.4}$$

where $m \times 1$ vector $\boldsymbol{k}(\boldsymbol{x})$ has entry j given by $K(\boldsymbol{x}, \boldsymbol{x}_j)$ for $\boldsymbol{x}_j \in \boldsymbol{X}^D$, $\gamma(\boldsymbol{x}) = 1 - \mathbf{1}^T \boldsymbol{K}^{-1} \boldsymbol{k}(\boldsymbol{x})$, and $\hat{\beta} = (\mathbf{1}^T \boldsymbol{K}^{-1} \mathbf{1})^{-1} \mathbf{1}^T \boldsymbol{K}^{-1} \boldsymbol{Y}^D$.

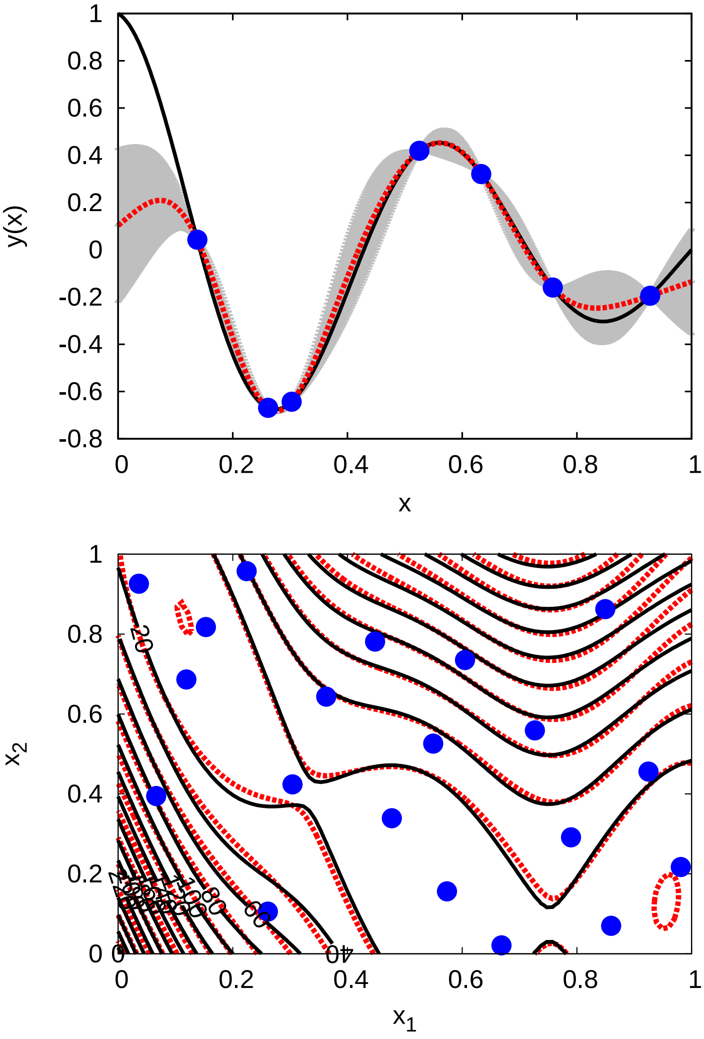


Figure 4.2: Kriging mean predictions for two simple examples. The translucent area is the 95 % confidence band. Top: 1D Santner2003 function, $\tau^2 = 10^{-11}$ and $\xi_1 = 0.14$. Bottom: 2D Branin function, with $\tau^2 = 10^{-11}$ and $\boldsymbol{\xi} = (0.37, 1.46)^T$.

The (ordinary) Kriging predictor is given by (4.3), whereas the Kriging variance is given by (4.4). Noteworthy is that (4.3) is the best unbiased linear predictor (BLUP) for $Y(\cdot)$, and (4.4) is the Mean Squared Error (MSE) of $Y(\cdot)|\boldsymbol{Y}^D, \boldsymbol{\theta}$.

An important and rather unique feature to the Kriging is that by using the variance estimate $\hat{s}^2(\boldsymbol{x})$ we can calculate confidence bands around the predictive mean: the upper and lower confidence bounds are $\hat{y}(\boldsymbol{x}) \pm t(1-\alpha, m-1)\frac{\hat{s}(\boldsymbol{x})}{\sqrt{m}}$. When $t(1-\alpha, m-1)$ is the 95th percentile of this probability distribution, we have $\alpha = 0.95$. See Figure 4.4 (Top). Note that for regions where the confidence bands are tight, we are with high probability well-informed about the unknown function $Y(\cdot)$.

As seen, there are many practical assumptions made, for example that Kriging relies on intrinsic stationarity, and that enough training data are available for the estimation of the covariance. But perhaps the strongest one is that correlation between errors is directly related to the metric distance between the data points [JSW98]. On the other hand, linear regression assumes an even stronger assumption of independent errors.

Empirical Kriging: For computer models the values of the covariance parameters, $\boldsymbol{\theta}$, are rarely known. The variance σ^2 can be estimated by $\hat{\sigma}^2 = \frac{1}{m-3}(\boldsymbol{Y}^D - \mathbf{1}\hat{\beta})^T \boldsymbol{K}^{-1}(\boldsymbol{Y}^D - \mathbf{1}\hat{\beta})$, whereas the correlation parameters, $\boldsymbol{\xi}$, are typically estimated using Maximum Likelihood Estimation (MLE), that is, to maximise the integrated likelihood function of $\boldsymbol{\xi}$. As shown in [SWMW89] an equivalent optimisation problem is to minimise

$$-2\ln\{L(\boldsymbol{\xi})\} = m\ln\{\hat{\sigma}^2\} + \ln\{|\boldsymbol{K}|\}. \tag{4.5}$$

The search domain in MLE for $\boldsymbol{\xi}$ is here denoted by $\Xi \subseteq \mathbb{R}^p$. Note that the MLE computation to fit the Kriging needs the determinant and the matrix inversion of the $m \times m$ correlation matrix $\boldsymbol{K}$ for a large number of $\boldsymbol{\xi}$. For this, usually Cholesky factorization is deployed. Cholesky is the factorization of $\boldsymbol{K}$ into $\boldsymbol{C}\boldsymbol{C}^T$, which is an computationally expensive operation for large m, more precisely $\sim \mathcal{O}(m^3)$. Before the MLE computation, the training data is normalised for both $\boldsymbol{X}^D$ and $\boldsymbol{Y}^D$, by scaling the mean, and divide by the standard deviation. This enables a simple yet informative sensitivity analysis, where the values of the correlation parameter associated with different input variables are compared. A higher correlation parameter value indicates a lower contribution to the total variability, and vice versa.

The computationally stable approach: The Cholesky factorizaton of $\boldsymbol{K}_{\boldsymbol{\xi}}$ suffers from numerical instabilities due to ill-conditioning, which occurs when the matrix is nearly singular. To prevent near singularity, which could cause the Cholesky decomposition to fail, a so-called nugget parameter ($\tau^2 \in \mathbb{R}$) can be added to the diagonal elements of the correlation matrix $\boldsymbol{K}$. The nugget parameter, typically very small in size, is mostly used to provide a more stable computation of the determinant and the inverse of the "true" $\boldsymbol{K}$, but results in a different Kriging model that no longer enforces the interpolatory property with respect to the underlying data $\boldsymbol{Y}^D$. The interpolatory property is considered to be one of the strengths of the Kriging. Of course, in some cases the discrepancy between the Kriging predictor and $\boldsymbol{Y}^D$ is desired. For example, small discrepancies tend to smooth out the Kriging predictor surface around the design points $\boldsymbol{X}^D$, and result in more conservative variance estimates. The use of a nugget parameter is also justified whenever there is a randomness in $y(\cdot)$ that should be accounted for, sometimes referred to as residual variability.

A trade-off between robustness and the desire to be "close" to interpolatory can be made by the addition of a penalty term to the likelihood function $L(\boldsymbol{\xi})$, to control the approximation error at the design points $\boldsymbol{X}^D$. Following [AC12], we replace $L(\boldsymbol{\xi})$ in (4.5) by $\pi(\boldsymbol{\xi}, \tau^2)L(\boldsymbol{\xi})$, with

$$\pi(\boldsymbol{\xi}, \tau^2) = \exp\left\{ -2\frac{\frac{1}{m}\left(\boldsymbol{Y}^D - \hat{y}(\boldsymbol{X}^D)\right)^T\left(\boldsymbol{Y}^D - \hat{y}(\boldsymbol{X}^D)\right)}{\epsilon\frac{1}{m}\left(\boldsymbol{Y}^D - \mathbf{1}\hat{\beta}\right)^T\left(\boldsymbol{Y}^D - \mathbf{1}\hat{\beta}\right)} \right\}, \tag{4.6}$$

where $\epsilon > 0$ is the fraction considered acceptable between the MSE error at $\boldsymbol{X}^D$, and the "maximum" error defined by $\frac{1}{m}\left(\boldsymbol{Y}^D - \mathbf{1}\hat{\beta}\right)^T\left(\boldsymbol{Y}^D - \mathbf{1}\hat{\beta}\right)$. The regions of $\boldsymbol{\Xi}$ with larger approximation error are therefore penalised more severely. The MLE is then performed jointly with respect $\boldsymbol{\xi}$ and τ^2. To simplify, we will judiciously select a handful of values for τ^2, and for each minimise $-2\ln\{\pi(\boldsymbol{\xi};\tau^2)L(\boldsymbol{\xi})\}$, in order to find the MLE estimates ($\hat{\boldsymbol{\xi}}$,$\hat{\tau}^2$) for $(\boldsymbol{\xi}, \tau^2)$ that maximises the integrated likelihood. τ^2 is here restricted to $T = [0, \hat{\sigma}^2]$.

The Kriging predictor and variance in this work are given by (4.3) and (4.4), respectively, with plug-in estimates $(\hat{\sigma}^2, \hat{\boldsymbol{\xi}}, \hat{\tau}^2)$ for $(\sigma^2, \boldsymbol{\xi}, \tau^2)$.

A direct computation of $\ln\{|\boldsymbol{K}|\}$ could run into underflow: the determinant may become so small it cannot be represented by a double precision floating point number. To mitigate the risk of underflow, we utilise that $|\boldsymbol{K}| = \prod_j \boldsymbol{K}_{jj}$, since $\boldsymbol{K}$ is a triangular matrix, and that $\boldsymbol{K} = \boldsymbol{C}\boldsymbol{C}^T$ (Cholesky), to obtain the following more robust expression: $\ln\{\det(\boldsymbol{K})\} = m\ln\{|\boldsymbol{K}|^{\frac{1}{m}}\} = m\ln\{\prod_j \boldsymbol{C}_{jj}^{\frac{2}{m}}\}$ [LNS02]. There are other similar ways to obtain a computationally stable Kriging model (see, e.g., [LNS02, RHK11]).

4.5 Surrogate-Based Optimisation (SBO): a step-by-step procedure for PSA

Surrogate-based optimisation (SBO) is a popular approach for using surrogate models with already existing optimisation procedures. The general algorithm is presented as a sequential diagram in Figure 4.3. In each iteration of SBO, the optimiser (e.g. a genetic algorithm) is applied on some design criterion based on knowledge gained from the surrogate model. Then, from one iteration to the next the surrogate is updated with the design solution we found. This procedure thus builds adaptive surrogates that are made to target the regions of the design space predicted to contain solutions of most interest. Hence, if poor predictions are made the surrogate learns from this new knowledge by re-calibration. The initial training data supplied to the surrogate model does not need to be so large it fills the entire design space, because the training data is updated sequentially with new design points during the search. A good coverage is however desired to ensure the efficiency of the approach. Also, instead of simply optimise the surrogate response, more complex surrogate-based criteria can be deployed, for example by making trade-offs with "code uncertainty" to ensure exploration as well as to improve the prediction quality. By code uncertainty we mean uncertainty arising from our inability to run the computer model for enough design configurations to know the model's output over the entire design space. This uncertainty can be estimated by the distance in some metric between the computer model output and the surrogate prediction over the design space.

We here put forward an efficient optimisation method based on surrogate models to compliment an appropriate choice of optimisation formulation for the PSA design problem. We have identified the following key properties for an efficient optimisation strategy:

- aim to use as few evaluations of the objective function as possible;
- guarantee theoretical global convergence in a bounded hyperrectangle domain, and not getting "trapped" at local optima;
- capable to be used efficiently for problems with multiple objectives;
- demonstrate a good trade-off between exploration and exploitation;
- be able to utilise high performance computing, and
- if interupted it should be capable to restart at interuption point.

We also feel the following assumptions should be made:

- the design space is continuous, and is represented by at most ten design variables;
- the computer simulations of the underlying computer model are deterministic, and considered computationally expensive;
- no modification of the underlying computer model implementation is allowed, and
- one surrogate model is applied to each of the quantities of interest (Purity, Recovery, etc.) from the computer model.

We will now present a step-by-step procedure customised to PSA optimisation, which respects the requirements given above. The proposed procedure may need to be modified if the requirements are different for other PSA applications. Nonetheless, we feel confident this procedure is appropriate to efficiently solve a wide range of PSA design problems. Multi-objective optimisation will be mentioned, but addressed in the next chapter.

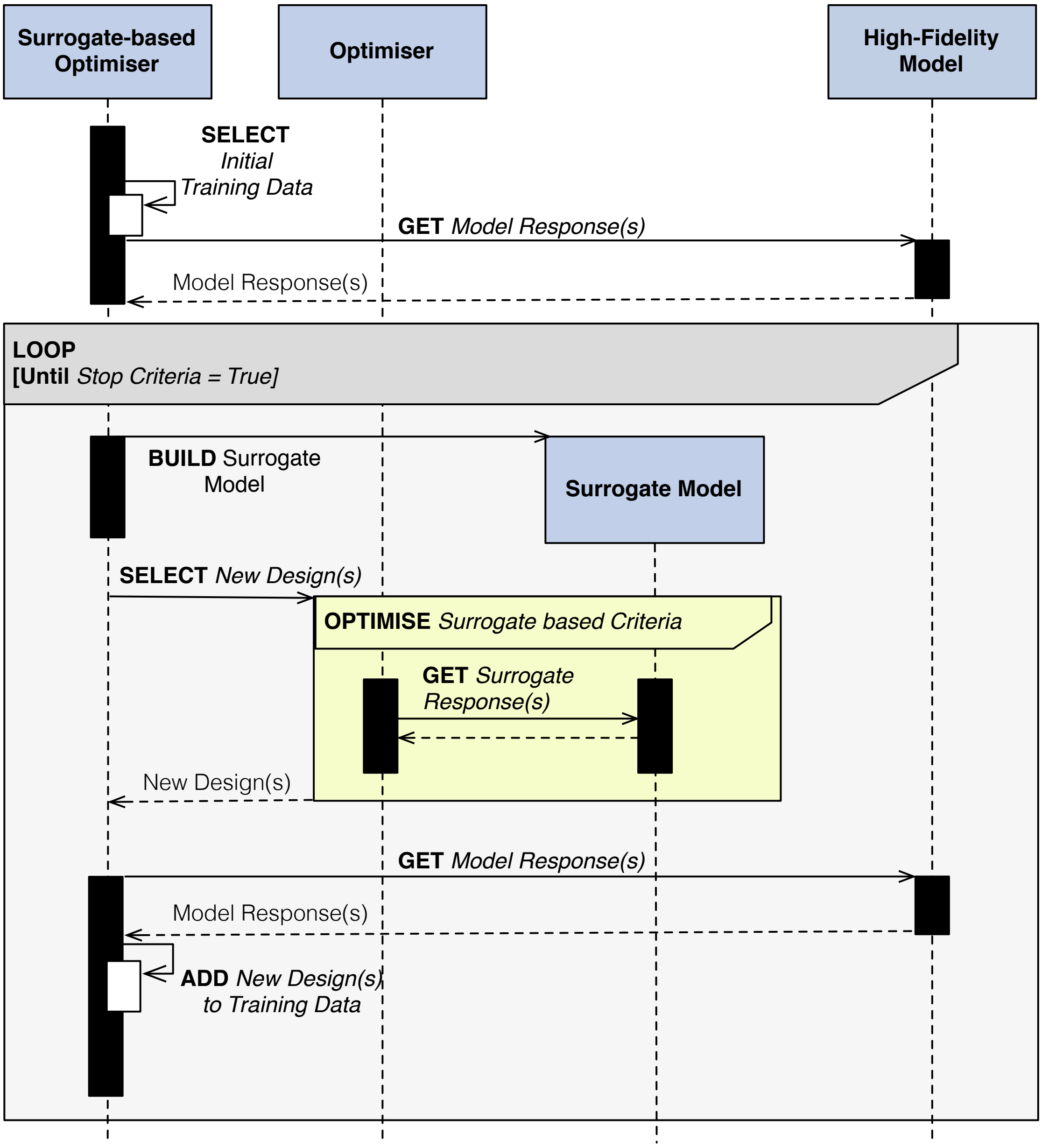


Figure 4.3: A sequential diagram of the interaction between the surrogate model, the optimiser, and the detailed model, representing the core of conventional surrogate-based optimisation (SBO).

Step 1. Choice of surrogate model

There are many different surrogate modelling techniques available, and it is nearly impossible to know beforehand which technique that would result in the best approximative model of the computer model's response. We will here discuss briefly the reasoning behind our consistent belief in Kriging for SBO, as long as the underlying computer model output is rather smooth with respect to the design variables. However, we will identify what we believe to be the advantages and the limitations of Kriging, and indicate when other surrogate techniques may be more advantageous.

Surrogate modelling techniques can be divided into regression with or without the interpolatory requirement [Jon01]. Regression with interpolatory requirement is when the response values of the surrogate model coincide with the responses of the original model, for all data points that belong to the training data, that is, $\hat{y}(\boldsymbol{x}_i) = y(\boldsymbol{x}_i), \forall \boldsymbol{x}_i \in \boldsymbol{X}^D$. Recall that $\hat{y}(\cdot)$ is the surrogate model and $\{(\boldsymbol{x}_i, y(\boldsymbol{x}_i))^T\}_{i=1}^m$ is the training data.

The most common surrogate modelling technique is the Polynomial Regression (PR) which often is used as benchmark when new surrogates are developed. PR is a surrogate model expressed as a sum of products of polynomials along the different input directions. Typically a low-order polynomial model is used because of the computational cost increases fast with the polynomial order. This approach has been successful in producing informative response surfaces, as well as in sensitivity analysis to characterise the influence of different inputs and mixed effects on the model response. However, PR as a surrogate for SBO could mislead the search (see [Jon01]). The reason is that non-interpolation methods such as PR focus on improving the average fit, rather than emulating the shape of the objective function [Jon01, Jin05]. Mimicing the shape is essential for optimisation where the global optimum should be localised, otherwise the optimiser may get caught in a local basin. Interpolation methods are thus preferred to non-interpolation methods in the context of global optimisation. There is scattered data interpolation, and interpolation based on selection of points. Scattered data interpolation aim to solve the problem of finding a smooth function of "minimal degree" that enforces the interpolatory property. Interpolation methods based on selection of points aim to solve the problem of selecting the most appropriate set of training points to achieve an accurate approximation. We are interest in the use of scattered data, because we then have more "freedom" in the selection of points, and the method will be applicable even if some design points are not feasible for the underlying computer model.

The Kriging model is an interpolation method that can be built on scattered data. Kriging also possess a rather unique statistical interpretation that provides without any extra computational effort an estimate of the MSE at any untried design points in the design space $\mathcal{X}$. The MSE estimate provides pointwise uncertainty predictions in $\mathcal{X}$ that can be utilised in SBO to identify the regions most-in-need of further exploration. There has been many Kriging-based design criteria proposed for SBO to make trade-offs between exploitation and exploration. The most popular method being "Efficient Global Optimisation", EGO. This will be discussed in Step 3.

Step 2. Initial sampling of the design space

The design space (also known as the design domain) is defined as the space which is bounded by the upper and lower bounds of the design variables represented. Experimental designs are techniques used to sample the design space $\mathcal{X} \subset \mathbb{R}^p$ to obtain a discrete representation $\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_m)^T$ of the continuous space. In computer experiments, when the evaluation of the computer model associated to a design point is computationally expensive, the design points have to be sampled more sparsely. Before building the Kriging surrogate model, an experimental design type should be chosen for the selection of the training data. The experimental design problem is concerned with sampling the design space to obtain training data $\boldsymbol{X}^D$, and sequentially or subsequently compute $\boldsymbol{Y}^D$, to fit the surrogate model with as few design points as possible. PSA applications deal with deterministic computer models with high-dimensional input data. Classical techniques to solve the experimental design problems are, e.g., full-factorial and central composite design that originally were proposed for physical experiments [Mon06]. They are typically symmetrical and optimal when random noise is present. See Figure 4.5.

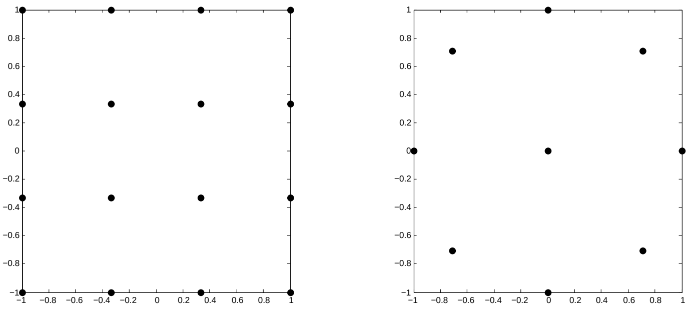


Figure 4.4: Examples of classical experimental designs in a two-dimensional domain: full-factorial design (Left) and central composite design (Right).

The classical designs are impractical for most applications whenever more than 3 design variables are considered, since the total number of points of the design grows exponentially with the number of design variables (p), $\Pi_{\ell=1}^{p} m_\ell$, where m_ℓ is the number of points in each direction.

It is important to distinguish between experimental design for computer and physical experiments. The most noticeable difference is that physical experiments suffer from aleatory uncertainty whereas computer experiments do not. Uncertainty can be classified as either aleatory or epistemic. Aleatory uncertainty is related to variability in experimental units and from other physical factors which usually are unavoidable, whereas epistemic is representing human ignorance. In this study only deterministic computer experiments are considered, for which aleatory uncertainties are not present. We do instead refer the interested reader to [BPS99, Pis95] and references therein for literature on design and modelling under uncertainty, and to the work of Babuska et al. [BNT07] which is addressing the importance of acknowledging the presence of uncertainty in computational science. The deterministic nature of computer experiments leads to improved computer model transparency, and the absence of random error makes systematic bias the only affecting factor [SWMW89]. The experimental design selection and size may also differ between computer and physical experiments. For physical experiments for instance the design should respect the experimental setup and our possibilities to measure the quantities of interest, whereas computer experiments instead put emphasis on the computational costs and the validity of the underlying model. McKay et al. [MBC79] proposed the first experimental design for computer experiments, called the Latin Hypercube design (LHD). LHD is a popular experimental design for computer experiments, especially for $p > 3$. Other experimental designs for deterministic computer experiments [SLC01, Mon06] are, e.g., Hammersley sequence sampling, maximin(Mm) and minimax(mM) designs, Taguchi designs, and orthogonal arrays. More recent studies have been devoted to improve experimental design selection using Kriging, see e.g., [KVB04, Kle09].

Experimental designs for deterministic computer experiments can be distinguished with the help of the following characteristics:

Space filling: Points distributed to represent all regions of the design space;

Uniform: Points distributed evenly across the design space;

Customised: Points selected with respect to the underlying problem (not genetic);

Distance based: Points are well spread from each other with respect to some distance metric;

Sequentialised: Points selected in a sequential manner.

If possible, it is desired that the experimental design is customised to the underlying problem. For global optimisation the space filling property is desired to enhance prior knowledge of the shape of the objective function. A theorem given in [TZ89] states that “in order to converge to the global optimum for a general continuous function, the sequence of iterates must be dense.” The full-factorial design (see Figure 4.5 Left) is space filling, but the total number of design points increase with an exponential rate as the number of design variables increases. This is commonly referred to as “Curse of Dimensionality.”

To generate the initial training data to the surrogate model, we restrict our attention to LHDs. The choice of using LHDs is motivated by its simplicity and track record of being competitive other experimental designs [BJ11]. Also, Kriging models are known to be difficult to fit on full factorial and central composite designs [MBS$^+$01]. LHDs, with m design points in a p-dimensional design space, are sampled with the following steps:

1. Divide the hypercube design domain $\mathcal{X}$ into m^p distinct hypercube sub-domains (the ranges of each of the p directions are partitioned into m intervals);
2. All directions are weighted equally in probability ($= \frac{1}{p}$). For all directions we sample randomly from the uniform distribution a point in each interval ($p \times m$);
3. Finally, pair randomly the outcomes from each direction to obtain m design points.

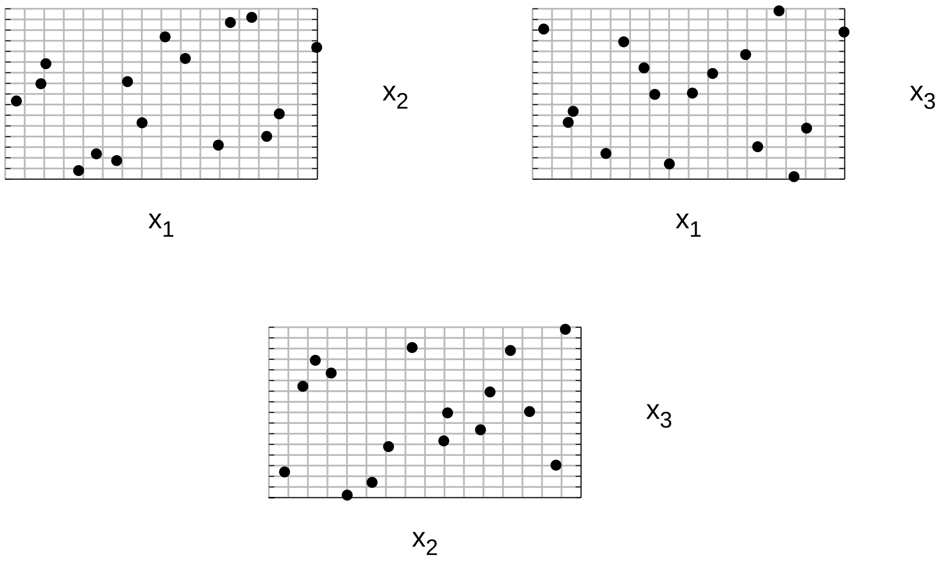


Figure 4.5: A LHD ($m = 16$) sampled in a 3-D design space.

LHDs are space filling and do not suffer from the "Curse of Dimensionality", therefore suitable for deterministic computer experiments. To further improve the spread of the initial training data $\boldsymbol{X}^D$, we will use the Maximin(Mm) LHDs [MM95]. The maximin(Mm) and minimax(mM) criteria (see [JMY90]) can be used to quantify the spread of a design and its coverage in the design space $\mathcal{X}$, which make them suitable as experimental design criteria. The Mm criterion tends to maximise the minimum interpoint distance of the design $\boldsymbol{X}^D$, to improve the spread, whereas the mM criterion minimises the distance from any point in the design space, $\mathcal{X}$, to a point in $\boldsymbol{X}^D$, to promote coverage of $\mathcal{X}$. In practice the mM criterion is considered to be the most informative of the interior of the design space, but is difficult to compute, and for this reason needs to be approximated. Mm and mM are typically conflicting [BJ11], and it is not obvious which criterion most advantageous for training the Kriging. Because the LHDs are easy and fast to compute, a good trade-off between the Mm and mM can be achieved with MmLHD [BJ11]. A MmLHD can be computed by first generating a pool of LHD candidates $\{\boldsymbol{X}^{D_i}\}_i$, and then select the one with highest Mm value. Also, perhaps even more important to the predictive quality than the positioning of the data points, is the choice of training data size. Initially, for the first step in the analysis, $10p$ design points in the training data are often sufficient to achieve a good enough coverage [LSW09]. That is, when considering a 10-dimensional design problem, we should start off the optimisation with training data of size 100.

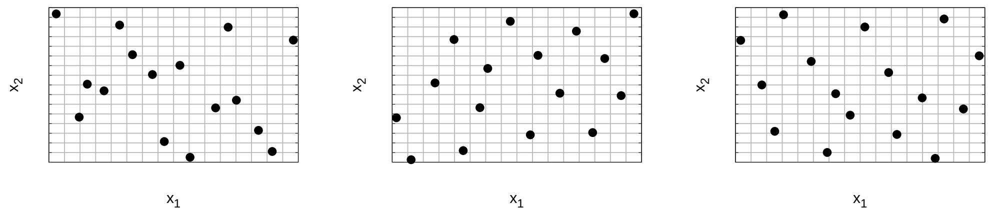


Figure 4.6: 2-D Examples of LHDs. Left: One-shot LHD. Middle: MmLHD selected from 10^7 LHDs, Right: mMLHD selected from 10^4 LHDs.

Sequential design strategies could have been used instead (see, e.g., [GL09]), for which we would expect a slight improvement in prediction accuracy, but we consider MmLHD to provide a better trade-off between computational complexity and prediction performance. Sequential designs gain momentum when we have some stop criteria that should be met (computational budget, accuracy, specified coverage level).

Step 3. Choice of surrogate based criterion

In optimisation we want to minimise (or maximise) some objective function, where the input values $\boldsymbol{x}^*$ of the global minimum of the objective function is desired:

$$\boldsymbol{x}^* = \operatorname*{arg\,min}_{\boldsymbol{x} \in \mathcal{X}} y(\boldsymbol{x}), \tag{4.7}$$

where the objective function is defined as $y(\boldsymbol{x}) : \mathcal{X} \to \mathbb{R}$. When $y(\cdot)$ is computationally expensive, we can reformulate the problem to an "approximative" optimisation problem using surrogate models. The naïve approach is to solve the same optimisation problem but on the surrogate model:

$$\hat{\boldsymbol{x}}^* = \operatorname*{arg\,min}_{\boldsymbol{x} \in \mathcal{X}} \hat{y}(\boldsymbol{x}), \tag{4.8}$$

where $\hat{y}(\cdot)$ is the surrogate model for $y(\cdot)$, and $\hat{\boldsymbol{x}}^*$ is the minimum to this "approximative" optimisation problem. The major issue with this approach is that $\hat{\boldsymbol{x}}^*$ can turn out to be very different from $\boldsymbol{x}^*$. In principle, the better approximative model we are able to build, the more confident we will be in the solution $\hat{\boldsymbol{x}}^*$. This is the motivation for the SBO approach. In SBO we are updating in a sequential manner the surrogate model with the best solution found for some chosen surrogate based criterion, $\mathrm{SbC}(\cdot)$:

$$\boldsymbol{x}_k^* = \operatorname*{arg\,max}_{\boldsymbol{x} \in \mathcal{X}} \mathrm{SbC}(\boldsymbol{x}, D_k, \hat{y}^{D_k}(\cdot), \hat{s}^{D_k}(\cdot)), k = 1, 2, \ldots, \tag{4.9}$$

where $D_k = D \bigcup_{i=1}^{k-1} \boldsymbol{x}_i^*$. The most common surrogate based criteria for the optimisation problem (4.7) are listed below.

Predictive Mean:

$$\mathrm{SbC}(\boldsymbol{x}, \cdot) = \hat{y}(\boldsymbol{x}) \tag{4.10}$$

Maximal Variance (MV):

$$\mathrm{SbC}(\boldsymbol{x}, \cdot) = \hat{s}^2(\boldsymbol{x}) \tag{4.11}$$

Expected Improvement: When the underlying function $y(\cdot)$ is modelled as a probability distribution (Kriging), a popular approach is to find the point $\boldsymbol{x}$ expected to deliver the largest improvement relative to the best solution we already observed [JSW98]:

$$\mathrm{SbC}(\boldsymbol{x},\cdot) = E_Y\left(\max\{y_{min} - Y(\boldsymbol{x}), 0\}\right). \tag{4.12}$$

When Normal distribution is assumed, this optimisation problem can be written in closed form:

$$\mathrm{SbC}(\boldsymbol{x},\cdot) = \hat{s}(\boldsymbol{x})\left(\left(\frac{y_{min} - \hat{y}(\boldsymbol{x})}{\hat{s}(\boldsymbol{x})}\right)\Phi\left(\frac{y_{min} - \hat{y}(\boldsymbol{x})}{\hat{s}(\boldsymbol{x})}\right) + \phi\left(\frac{y_{min} - \hat{y}(\boldsymbol{x})}{\hat{s}(\boldsymbol{x})}\right)\right) \tag{4.13}$$

where y_{min} is the reference solution (current minima, e.g., $\min\{\boldsymbol{Y}^D\}$), Φ is the cumulative distribution and ϕ is the probability density function for the conditional posterior probability distribution of the Kriging. This criterion is useful when applied iteratively to improve the global approximation while searching for the optima.

Probability of Improvement: Select a target value $y_{ref} < y_{min}$ as a reference point [Moc94]:

$$\mathrm{SbC}(\boldsymbol{x},\cdot) = \Phi_Y\left(\frac{y_{ref} - \hat{y}(\boldsymbol{x})}{\hat{s}(\boldsymbol{x})}\right). \tag{4.14}$$

By convention y_{ref} is taken to be some factor smaller than the current best solution, that is, $y_{ref} = y_{min} - \alpha|y_{min}|$ for some $\alpha > 0$.

"Maximal Variance" aims to reduce the code uncertainty by selecting the point where the uncertainty is highest. The major drawback is that it does not attempt to target the regions of most interest for the original optimisation problem, and therefore on its own is not passable for efficient optimisation of "expensive" computer models. The last two criteria are Bayesian approaches to global optimisiation, which perform average case analysis. First a prior distribution is chosen for the objective function $y(\cdot)$, denoted by $P(y)$. This distribution is then updated by plugging in the known observations $D = \{(\boldsymbol{x}_i, y_i)^T\}_{i=1}^m$ in Bayes' rule. This becomes our posterior distribution for $y(\cdot)$. Although deterministic computer experiments in general do not have a "true" distribution, this approach has been successful for representing our knowledge about $y(\cdot)$. The only known values of $y(\boldsymbol{x})$ are the ones given in the training data, hence $y(\cdot)$ can be seen as a random function except for the points already observed. Here the two criteria deviate: "Expected improvement" wants to minimise the expected deviation from the global minimum using the posterior distribution of $y(\boldsymbol{x})$, to decide which the next observation $(\boldsymbol{x}_{i+1}, y(\boldsymbol{x}_{i+1}))^T$, this is then repeated to form an iterative optimisation procedure. For the "probability of improvement" the goal is to select the next observation $\boldsymbol{x}_{i+1}$ where the probability to exceed some target value is the highest. The "probability of improvement" tends to sample around the current best point, until the $\hat{s}(\boldsymbol{x})$ in the region becomes small enough to force the search to continue in other promising regions. Under certain mild assumptions, the global convergence is guaranteed. The drawback is that we need to specify y_{ref}: too small improvement (that is, $y_{min} - y_{ref}$) leads to an exhaustive local search around the current best solution, and with too large we are not fully utilizing the given information on where the current best solution lies. The "expected improvement" criterion guarantees global convergence as well [Loc97], and favoured to "probability of improvement" as we do not need to specify any target y_{ref}. The drawback is that whenever the conditional variance $\hat{s}^2(\boldsymbol{x})$ is under-estimated, the same problem as with small improvement for "probability of improvement" might occur (that is, extreme local search).

The surrogate-based optimisation based on "expected improvement" (EI) is often referred to as "Efficient Global Optimization" (EGO), and will be used in our case studies. The advantage with using the posterior distribution of $f(\boldsymbol{x}), \forall \boldsymbol{x}$ derived from Kriging is that both the mean and variance (conditional on the available data D) are given directly without any extra work.

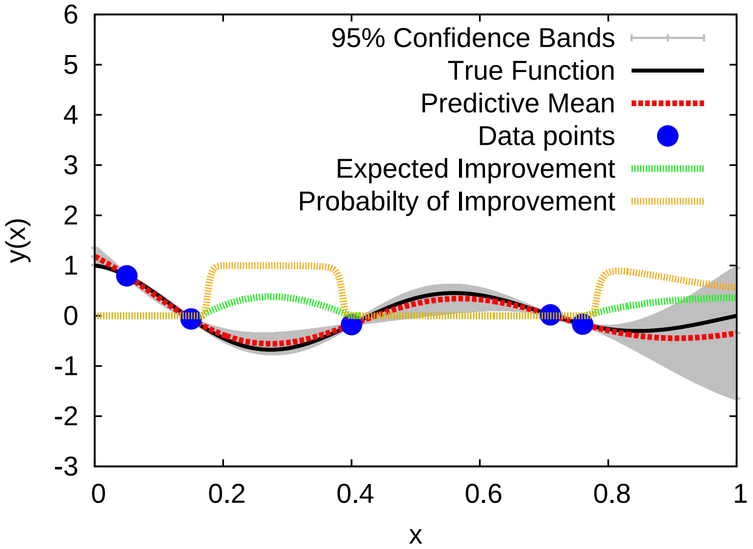


Figure 4.7: Illustrative example: Expected Improvement, and Probability of Improvement with $y_{ref} = y_{min} - 0.4|y_{min}|$ as reference point.

This EI criterion provides a compromise between exploitation and exploration. They also have natural statistical interpretations. Another type of approach is to combine several criteria, either applied in different iterations or merged into a single criterion. For example, we can specify some desired trade-off between prediction and its esimated uncertainty:

$$\hat{\boldsymbol{x}}^* = \arg\min_{\boldsymbol{x} \in \mathcal{X}} \hat{y}(\boldsymbol{x}) - \alpha \hat{s}(\boldsymbol{x}). \tag{4.15}$$

Here the trade-off is specified by some weighting parameter $\alpha > 0$. This can be seen as a statistical lower bound [CJ97]. If α is chosen to be conservative, e.g. $\alpha = 3$ (99.7- normal percentile), we can use a branch-and-bound algorithm. However, global convergence for continuous functions cannot be guaranteed given that this bound is an *estimate*.

Alternatively, shifting between minimisation $\hat{y}(\boldsymbol{x})$, and maximising $\hat{s}(\boldsymbol{x})$ from one iteration to the next can turn out to be sensible a choice [VK10].

Step 4. Choice of optimiser for surrogate based criteria

Global optimisation (GO) methods [TZ89] guarantee the convergence to a global optimum which make them attractive for optimal design. GO methods have the ability to explore more widely and avoid local basins in which local optimisers would become “trapped.” They tend to have worse convergence rates than local optimisers because of the extra effort devoted to exploration. In other words, GO requires more model evaluations, and for this reason impractical for the expensive computational models typically appearing in engineering. Naïve global optimisation methods put too much emphasis on exploration. Random search is such an example. More effective methods make appropriate trade offs between exploitation and exloration, e.g. metaheuristic algorithms such as genetic algorithms and simulated annealing. Only a few contributions to design optimisation of PSA systems have adopted global optimisation routines:

- An aJG adaptation of the multi objective simulated annealing (MOSA-aJG) has been applied for a 2-bed 4-step PSA system. A modification of MOSA-aJG was necessary to accommodate for the high computational demand of the high-fidelity PSA model employed. for a O_2/N_2 gas mixture [SG07]
- Multi objective genetic algorithm (MOGA) for a 2-bed/4-step PSA system for production of N_2 from air [FFB09b]. Here a simple CSTR-based PSA model was used to reduce computational requirements.

However, if the objective space is convex, the local optimum is in fact the global optimum. This makes local optimisers advantageous due to their potentially fast convergence towards the optima, unless the numerical noise of the underlying computer code render misleading gradients.

Experience suggests that the objective space more often than not is convex for the purity and recovery of the product, as shown for a case example for production of N_2 from air [FFB09b]. But with more complex cases, and different quantities of interest, the objective space could be non-convex and thus causing difficulty for gradient-based optimisers.

To start optimisation with a global optimiser and finish it by some local optimiser is often deemed reasonable. The use of a local optimiser can be useful for applications where the precision of the solution is considered to be important. On the other hand, mathematical models for engineering design are often not accurate enough to justify the use of some local method to finish the optimisation. PSA models tend to provide good esimates of performance, but with some imprecision, therefore we have not followed up the global methods by some local method.

Step 5. Choice of stop criteria

One of the challenges with iterative strategies for optimisation is to decide the when to stop. The most common criteria when to stop are:

- When reached the maximum allowed number of iterations, or some computational budget (e.g. given in CPU time)
- When the changes in the objective space (or design space) between successive iterations is small enough (that is, below a given limit)

4.6 Parallel implementation

To exploit parallelism across multiple CPU processors we propose the use of so-called multi-point selection [Sch98, GLRC10, PFKH12]. Multiple CPUs allow, at any given time, each CPU to work independently on a single design point. With multi-point selection, instead of only selecting a single design point per iteration, we select several design points. Typically the number of detailed model evaluations should be proportional to the number of available CPUs. This basic HPC algorithm may lead to a speed-up proportional to the number of CPUs. However, because PSA simulations are assigned to different CPUs, and usually finalised at different times, the CPUs on your computer system may become idle for long periods of time, see Figure 4.8. An alternative approach is to use an asynchronous distributed implementation

[JLRGG12]. Such an approach allows a new simulation run to begin as soon as a CPU processor becomes available.

The goal is to select k design points. First, the design points are ordered from most to least promising with respect to the surrogate based criterion. One way is then to choose the k most promising design points as a batch. This typically is not efficent, because it is highly probable the several of the k most promising are neighbouring design points. An efficient procedure is instead to choose k design points sequentially, without undertaking any evaluations before the selection process is finalised. Also, inbetween each design point selection the Kriging is updated, otherwise the same point may be selected more than ones. If there are two equally promising design points, which also are considered to be the most promising, one option is to select one of them on random.

Ginsbourger et al. [GLRC10] formalised two Kriging update strategies that can be used while undergoing the multi-point selection process:

Kriging Believer strategy: Update the Kriging with the selected design point $\boldsymbol{x}^*$ and the corresponding dummy response value $y^* = \hat{y}(\boldsymbol{x}^*)$, using the current Kriging model's predictive mean (4.3). A possible consequence is a misguided Kriging model.

Constant Lier strategy: Update the Kriging with $\boldsymbol{x}^*$ and its response value y^* chosen for the specific-problem, e.g. $\min\{\boldsymbol{Y}^D\}$, $\text{mean}\{\boldsymbol{Y}^D\}$, and $\max\{\boldsymbol{Y}^D\}$. This is an approach that can be utilised to dictate the behaviour of the SBO method, for example to encourage exploration rather than exploitation, or vice versa.

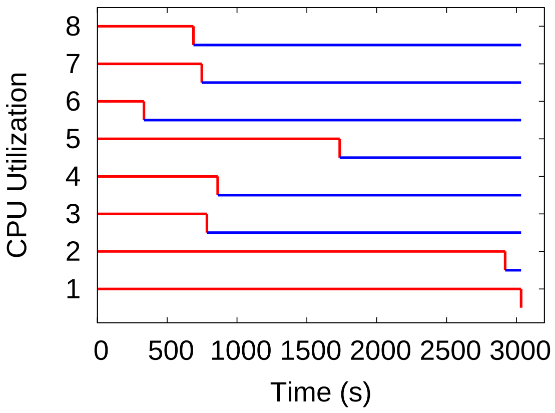


Figure 4.8: Illustration of CPU utilization of 8 CPUs for a single batch of 8 design points for evaluation by the detailed PSA model. Red lines represents active CPU usage, whereas blue lines represents idle CPU.

According to [GLRC10] the Constant Lier strategy with $y^* = \min\{\boldsymbol{Y}^D\}$ performed best in test for a simple example, the Branin-Hoo test function. The Kriging Believer is the strategy that will be used throughout our work, because we rely on the Kriging model already, and do not want to manipulate the search by the Constant Lier without fully understanding the implications of it. There are however concerns with the error propagation as a result of using the Kriging Believer with too many point selections in sequence.

4.7 Dual-Piston PSA case study

In this section we will demonstrate the use of SBO for the optimisation of a closed dual-piston PSA system for N_2/CO_2 separation.

4.7.1 The Dual-Piston PSA

Conventional PSA systems, as those described in Section 1.4, typically have long cycle duration to allow enough time for the bed to be pressurised. This however has a negative effect on the system's productivity. Productivity is defined by the amount of product produced per unit mass of the adsorbent per unit time, and is strongly correlated with the cycle time: the shorter cycle time, the higher productivity is achieved [RFK93].

Two well-known PSA systems that allows short-cycle-time designs are Rapid PSA (RPSA) and Dual-piston PSA (DP-PSA). RPSA is constructed from two process steps: pressurisation by feed, and countercurrent depressurisation with internal purging. RPSAs are well-established commercial products [FTR98], and the first one appeared in the early 1970s for N_2-CH_4 separation [TK71]. These systems are attractive under strict economic conditions because of their simple design. The other system, DP-PSA, consists of a single adsorption bed, with two pistons accommodated by cylinders, one on each side of the bed. The DP-PSA apparatus developed by Keller and Kuo [KK82] is the first known example.

The rapid movement of the pistons makes the fluid flow in the adsorption bed, generating steep pressure changes causing the gas mixture to separate. The two pistons can be controlled independently; for example, when operating the pistons out of phase, specified through the offset angles, $\phi_{1,2}$, the separation performance is likely to improve. A feed unit can be introduced at an intermediate position of the bed. Thanks to this unique setup, the DP-PSA can be used for the testing and characterization of adsorbent materials [DFB13].

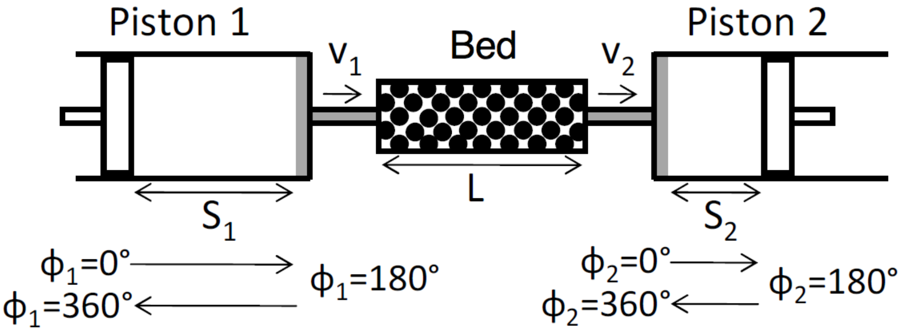


Figure 4.9: Dual-Piston PSA System

When the system is run as a closed system, while inside an temperature-controlled oven, a wide range of experiments can be completed in a relatively short period of time. With this setup the cycle times can be reduced to a few seconds. The closed system setup leads to a total reflux process. There is a pressing need for further research on adsorbents for CO_2 separation from flue gas streams [DFB13].

4.7.2 The modelling and simulation of DP-PSA

The closed DP-PSA system considered in this work is shown in Figure 4.9. As the operation of the binary separation progresses the light component moves in one direction, while the heavy component moves in the opposite direction, ending up in the piston chambers positioned at the two ends of the bed.

We are interested in predicting the separation performance using computer simulations that rely on accurate mathematical models describing the process behaviour of the DP-PSA. The model equations used to describe the DP-PSA process can be found in [SJ97, AFR02], and the computer simulator is implemented as described in [FFB13]. We have not used any acceleration schemes for faster convergence to CSS (see [FFB13]); instead we adopt the successive substitution approach for the simulation. Courtesy to Prof. Brandani at the University of Edinburgh for providing the DP-PSA simulator code for this work.

The underlying mathematical model relies on the following assumptions:

- The gases obey the ideal gas law
- Axial dispersed plug flow
- No frictional pressure drop along the bed
- Isothermal system
- Langmuir adsorption model

- LDF approximation of the adsorption rate for the mass transfer
- The gases in the piston chambers are well-mixed
- The dead volume is of equal size in the two cylinders

The isothermal assumption is considered appropriate based on experiments presented in [FFB13], where the most significant deviation from the temperature mean was 2K. During the first hour of the experiment the mean temperature over a cycle increased by up to 3K, but considered to be a result of piston friction forces. We are interested in short cycle times for which the friction effects should be negligible. The pressure drop, also monitored, was consistently below 2.5mbar. The governing mathematical equations for the adsorption bed, given in Section 2.2, are here accompanied by a set of ODEs related to the pistons, see Table 4.1. The subscripts $p1, p2$ refer to Piston 1 and 2, respectively.

Table 4.1: The model equations related to the pistons.

The mass balance for piston 1 and 2:

$$\begin{aligned} \frac{dy_{i,p1}}{dt} - \frac{u_1 - |u_1|}{2V_{p1}}\left(y_{i,p1} - y_{i,0}\right) = 0 \\ \frac{dy_{i,p2}}{dt} - \frac{u_2 - |u_2|}{2V_{p2}}\left(y_{i,L} - y_{i,p2}\right) = 0 \end{aligned} \tag{4.16}$$

The position of the piston described by a sinusoidal cycle movement:

$$S(t) = S_{0,j} + \frac{S_{L,j} - S_{0,j}}{2}\left(1 - \cos\left\{\frac{2\pi t}{t_c} + \phi_j\right\}\right) \tag{4.17}$$

The velocity at the bed ends:

$$\begin{aligned} u_0 &= -\frac{V_{p1}}{p\varepsilon_b\pi r_b^2}\frac{dp}{dt} - \frac{1}{\varepsilon_b\pi r_b^2}\frac{dV_{p1}}{dt} \\ u_L &= \frac{V_{p2}}{p\varepsilon_b\pi r_b^2}\frac{dp}{dt} + \frac{1}{\varepsilon_b\pi r_b^2}\frac{dV_{p2}}{dt} \end{aligned} \tag{4.18}$$

The complete DP-PSA model is here solved by the Method of Lines: PDAEs are converted with spatial discretisation into set of DAEs which then are simulated forward in time with time integration techniques. The mathematical model is discretised in the spatial coordinate with a flux-limiting finite volume scheme using the van Leer flux limiter [Lan98] with 51 spatial finite volume elements. This spatial discretisation scheme is conservative, and suitable for systems with sharp, moving fronts. Conservative schemes are particularly advantageous when simulating a closed system as they preserve throughout the run the total mass in the system. The total mass is an aggregation of the volume of the pistons, dead volume, bed gas phase and the mass adsorbed. The resulting semi-discretised computational model, represented by a system of DAEs, is then integrated in the temporal domain using the SUNDIALS solver suite [HBG$^+$05], with the backward differentiation formula (BDF) of order 5.

The CSS condition is checked at the end of each cycle when performing the successive substitution, and considered satisfied if the difference between the state variable profiles in the beginning and the end of the cycle is ε-small in the Euclidean norm on scaled values, for some chosen tolerance $\varepsilon > 0$.

Because the DP-PSA is a closed single-bed system, the simulation is relatively fast, which is ideal for testing methodology. For the simulation runs made in this optimisation study, the average CPU time is roughly 3 minutes per simulation run of the detailed DP-PSA model. The DP-PSA is also sufficiently complex to provide a challenge for the optimisation.

4.7.3 Case Setup

In this case study, the goal is to find optimal process conditions that lead to a CO_2-rich gas mixture in the piston chamber located on the left-hand side of the DP-PSA system. We consider a binary gas mixture of 85% N_2 / 15% CO_2, which is the typical flue gas composition in the feed streams that would enter post-combustion units. We run the DP-PSA system, with an adsorption bed packed with zeolite 13X pellets, at total reflux. Zeolite 13X is a benchmark adsorbent material for separation of CO_2 from flue gas [KSB03]. A single-site Langmuir adsorption isotherm model is adopted with parameters found in [XZW$^+$08] derived from the dual-site Langmuir adsorption isotherm model. The axial dispersion coefficient is assumed to be constant, and equal for both gas components. The simulation results with the DP-PSA simulator has shown a very good agreement to experimental data, in a case example with 80% N_2 / 20% CO_2 on a zeolite 13X packed adsorption bed [FFB13]. The process conditions are similar to the ones we use, and this will provide confidence in the design outcomes of our optimisation study.

To initialise the simulation the adsorbent is set to be in equilibrium with the gas phase concentration, and the pressure is set at 1 bar. See Table 4.2 for the adsorption bed and piston parameters.

DP-PSA Optimisation Problem. This is the first optimisation study using detailed computer simulations for tuning the DP-PSA process conditions to improve the separation performance. For the system in total reflux we are mostly concerned with finding the operating

Table 4.2: Adsorption bed and piston parameters for the DP-PSA unit.

Symbol	Parameter	Value	Unit
L_b	Bed length	0.13	m
r_b	Bed radius	0.0076	m
ϵ_b	Bed void fraction	0.4	-
r_p	Pellet radius	2.2×10^{-3}	m
ϵ_p	Pellet void fraction	0.625	-
D_{N_2,CO_2}	Axial dispersion coefficient	5×10^{-5}	m^2s^{-1}
L_p	Piston length	0.107	m
R_p	Piston radius	0.025	m
$S_{0,1}$	Start position of Piston 1	0.05	m
$S_{L,1}$	End position of Piston 1	0.1	m
$S_{0,2}$	Start position of Piston 2	0.0	m
$S_{L,2}$	End position of Piston 2	0.1	m

conditions that lead to the most CO_2-rich piston chamber at CSS. Hence, we are interested to maximise the following quantity of interest:

$$\text{Purity of } CO_2 \ (\%) \text{ in piston chamber 1 at CSS} = 100 \times \frac{\int\limits_{t_c} u y_{CO_2,p1} dt}{\int\limits_{t_c} u \sum\limits_{j=\{CO_2,N_2\}} y_{j,p1} dt} \tag{4.19}$$

The parameters allowed to be varied during the analysis are the cycle time (t_c), bed temperature (T_b), volume of piston chamber 1 and 2 (V_{p1}, V_{p2}), and offset angle of piston 1 and 2 (ϕ_{p1}, ϕ_{p2}). The process conditions have been selected judiciously along with the value ranges covered, see Table 4.3.

Table 4.3: Decision variables for the DP-PSA optimisation problem.

Variables		Range	unit
t_c	Cycle time	[1,20]	s
T_b	Bed temperature	[288.15,343.15]	K
V_{p1}	Volume of piston chamber 1	$[0.5,15]V_0$	m^3
V_{p2}	Volume of piston chamber 2	$[0.5,15]V_0$	m^3
ϕ_{p1}	Offset angle of piston 1	$[0,2\pi]$	-
ϕ_{p2}	Offset angle of piston 2	$[0,2\pi]$	-

The movement of the pistons in terms of phase angles controls the pressure variation in the bed. The starting position of the pistons will influence the minimum and maximum pressure, and in turn the separation performance. Productivity and purity are known to be conflicting, so faster cycles are expected to penalise the purity while enhancing productivity. The bed temperature is affecting the adsorption. As we run the system in total reflux, the volume of the piston chambers with respect to the total volume are expected to affect the separation.

4.7.4 Optimisation implementation details

For this analysis we follow the surrogate-based optimisation procedure described in Section 4.3 in order to demonstrate the performance improvement when surrogate models are used. This approach is compared to a stand-alone real-coded GA. The real-coded GA, with simulated binary crossover and polynomial mutation [DG96] is performed for 100 generations, and the following algorithm parameter settings were used: Population size 16, mutation rate 1/6, mutation index 20, crossover rate 0.9, crossover index 5. These settings are, or in close proximity to, the default values in the original real-coded GA.

The surrogate model selected is the Kriging model as described in Section 4.4, and is MLE-fitted to the training data with a real-coded GA, given the same settings as above with the exception that we now can afford to use a larger population size, 64. To avoid excessive fitting of the correlation parameters for Kriging, the number of generations is reduced to 50. The computationally-stable Kriging approach is deployed where we iterate over the nugget parameter values, $\hat{\tau}^2 = 10^{-i}$ for $i = 7, 8, \ldots, 14$, and for each estimate, perform the MLE. The nugget parameter choice resulting in the highest MLE is selected. The MLE procedure for the estimation of the correlation parameters is restricted to the domain $[0.3{,}15]^6$. This domain choice is reasonable: 0.3 represents a non-smooth response surface, whereas with 15 the squared exponential basis function shape resembles a constant. A larger value than 15 would not be meaningful and likely to lead to over-confidence in the Kriging predictions.

Following the internal optimisation, where design points are ranked based on some chosen surrogate model-based criteria, we enforce diversity in the design space to avoid clustering: if the distance between the most promising design point candidate $\boldsymbol{x}$ and the training data $\boldsymbol{X}^D$, given by the Euclidean norm

$$\left\| \tilde{\boldsymbol{x}} - \tilde{\boldsymbol{X}}^D \right\|_2 = \min_{\tilde{\boldsymbol{x}}^* \in \tilde{\boldsymbol{X}}^D} \left\{ \sqrt{\left(\sum_j^p (\tilde{x}_j - \tilde{x}_j^*)^2 \right)} \right\} \tag{4.20}$$

is $\varepsilon-$small, the design candidate is marked as an undesired candidate. Then the diversity is calculated for the second most promising design point candidate, and so on until a viable candidate is found. Here $\tilde{\boldsymbol{x}}$ and $\tilde{\boldsymbol{X}}^D$ are the normalised values of $\boldsymbol{x}$ and $\boldsymbol{X}^D$. Henceforth $\varepsilon = 10^{-3}$.

The Kriging model is updated and refitted to the training data every eight evaluation of the DP-PSA model. However, the Kriging computational cost as mentioned increases fast with the size of the training data, $\sim \mathcal{O}(m^3)$. Eventually the Kriging computation becomes intractable. We use a standard remedy to circumvent this problem: when the data consists of more than 60 design points, the Kriging correlation parameters are no longer updated.

4.7.5 Preliminary analysis

As part of our preliminary analysis, we performed a pathwalk consisting of hundred design points along a hyperline segment between two distant points in the six-dimensional hyper-rectangle design space, $\boldsymbol{x}^* = \boldsymbol{x}_1 + \lambda(\boldsymbol{x}_2 - \boldsymbol{x}_1)$, where $\lambda = [0, 1]$. See Figure 4.10, where $\boldsymbol{x}_1 = (15.7, 0.04, 1.27, 11.7, 7.73, 278)$ and $\boldsymbol{x}_2 = (4.76, 1.52, 0.34, 1.78, 10.4, 330.2)$. The figure reveals two important issues: a) the multi-modality in the objective space, and b) the numerical noise in the simulator output. Both these issues could lead to a premature convergence for local optimisers which are relying on gradient information. The optimisation problem on the surrogate can become easier than the original problem because the surrogate smooths out the artifical noise arising from the numerical error from solving the PDAEs. The presence of numerical noise in the objective function response has been reported when using a computer simulator for the analysis of PSA for N_2 separation from air.

See Figure 4.11 for an analysis of the Kriging predictor error for the DP-PSA model. The accuracy of the prediction is measured using the root-mean-square error (RMSE) given by

$$RMSE = \sqrt{\left(\frac{1}{1000}\sum_{j=1}^{1000}(\hat{y}(\boldsymbol{x}_j) - y(\boldsymbol{x}_j))^2\right)} \tag{4.21}$$

where $\hat{y}(\cdot)$ is the predictive values, and $y(\cdot)$ is the true responses of the detailed DP-PSA model. The RMSE is calculated using a hold-off set of detailed model evaluations for 1,000 design points generated with a MmLHD. The results indicate that the Kriging should be capable in guiding the optimisation process, even for small training samples. Moreover, from the figure we can observe that the accuracy only shows subtle improvement as the number of evaluations of the detailed model increases.

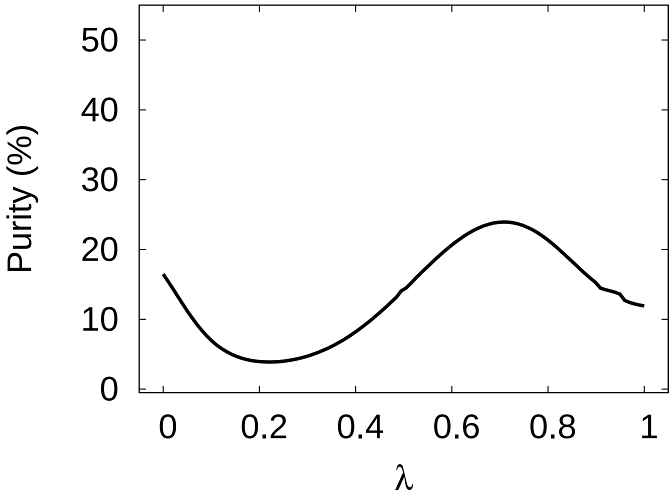


Figure 4.10: Pathwalk along a hyperline segment between two random distant points in the six-dimensional design space $\boldsymbol{X}^D$.

This is expected, as shown in [LSW09], the improvement that is expected in the global accuracy over the entire design space is subtle after the size of the training data exceeds $11 \times p$ design points, where p is the number of design points. Noteworthy is also the significant reduction observed in the variability in the prediction as the training data is extended.

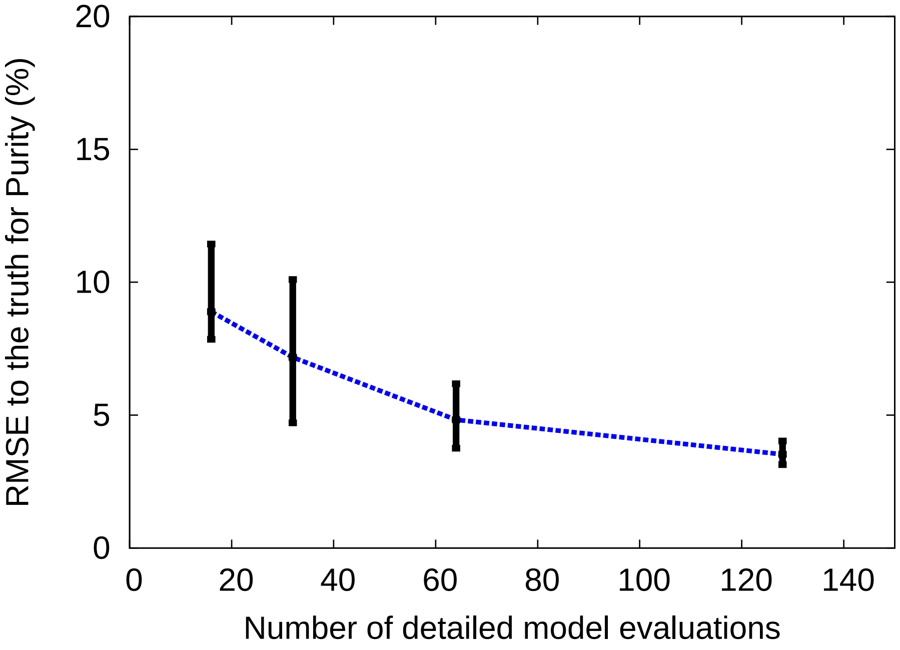


Figure 4.11: Kriging as an approximation for the DP-PSA model. The RMSE approximation error (4.21) is displayed with error bars based on averages over 10 sets of training data of sizes 16, 32, 64 and 128.

The total CPU time given to build the Kriging, and the time to execute the subroutine to fit the Kriging to the training data, for data of size 16, 32, 64, and 128, are presented in Table 4.4.

Table 4.4: CPU time for Kriging construction for DP-PSA case study, using training data of size m with six-dimensional data points. The CPU time is presented as averages over five runs.

Training data size, m	Average CPU time (s)
24	184
32	240
48	365
64	394
80	460

The remaining algorithm time is small in comparison and thus omitted.

4.7.6 Numerical results

The separation performance is measured by the CO_2 purity obtained in piston chamber 1 at CSS. This is the first optimisation study for DP-PSA and for this reason we cannot directly compare our results with any previous study.

The optimisation is performed with a given computational budget of 500 computer model evaluations of the DP-PSA system. The outcome of the optimisation is stochastic, because of the inherent variability of the internal optimisation methods adopted, namely GAs, and the initial training data upon which the Kriging is fitted. The results are therefore presented as averages over 5 runs, each with different initial sample. The impact of the GA parameters has been studied before for PSA with the conclusion that the generation number and the population size are more influential than the operation parameters (crossover, mutation, etc.) [FFB09b]. We have here fixed the population size to 16, but monitored the evolution of the performance with each new generation.

We are comparing the different optimisation approaches in terms of the computational work needed (represented by the number of evaluations of the detailed DP-PSA model) to achieve some desired level of separation performance. This measure of optimisation performance is advantageous, alongside the recorded CPU time, since it does not depend on the underlying computer system or the efficiency of the computer code implementation.

With the SbGA approach, because of the diversity control used in the design space, the search cannot reach the global optimum if a design point already has been selected to the training data that are within ε distance from the optimum, where ε is the parameter used in the diversity algorithm. We consider searching for near-optimal designs in this case to be acceptable.

Table 4.5: Optimisation results for DP-PSA case study

Performance variable	**value**
CO_2 purity	89.50 (%)
Decision variables	**value**
Cycle time (t_c)	20 (s)
Bed temperature (T_b)	342 (K)
Volume of piston chamber 1 (V_{p1})	$9.5V_0$ (m^3)
Volume of piston chamber 2 (V_{p2})	$15V_0$ (m^3)
Offset angle of piston 1 (ϕ_{p1})	0.875π
Offset angle of piston 2 (ϕ_{p2})	0.233π

The optimisation results are shown in Figure 4.12 for GA, SbGA, and EGO. The runs were performed on Intel Core i7 core 2.3 GHz machine with 8GB RAM. The best design configuration found resulted in 89.5 % of CO_2 purity, see Table 4.5. The separation performance delivered is promising considering the initial CO_2 concentration of 15 % in this closed-bed system. The construction of the Kriging is the computational bottleneck in the surrogate-based algorithms. For the SbGA and EGO runs, the total CPU time spent on building Kriging models and fitting the models to the training data was about 30 minutes, and could therefore be said to be roughly equivalent to 10 DP-PSA computer model evaluations.

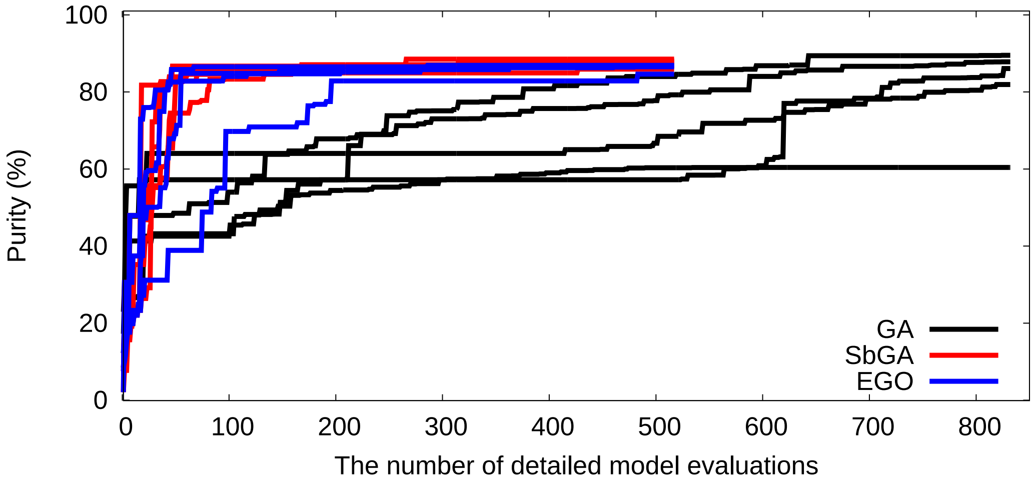


Figure 4.12: Computational results with GA, SbGA and EGO

Both EGO and SbGA outperform the stand-alone real-coded GA. SbGA obtained good design solutions already after 100 design points, that is, above 80% CO_2 purity, whereas GA for the same number of points was consistently below 65%.

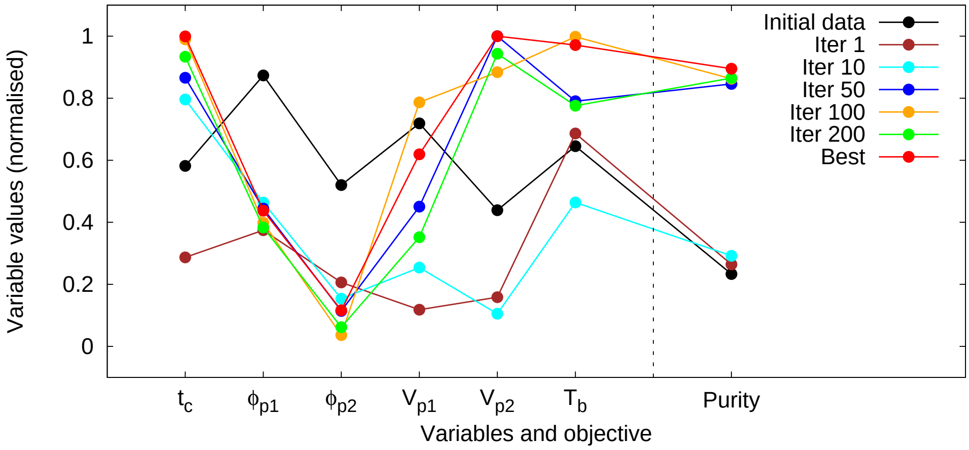


Figure 4.13: Parallel coordinate system for visualisation of the most promising design solutions obtained when using SbGA. The best solution is displayed for the initial design and some selected iterations, as well as an overkill solution (89.5 $\%$ Purity).

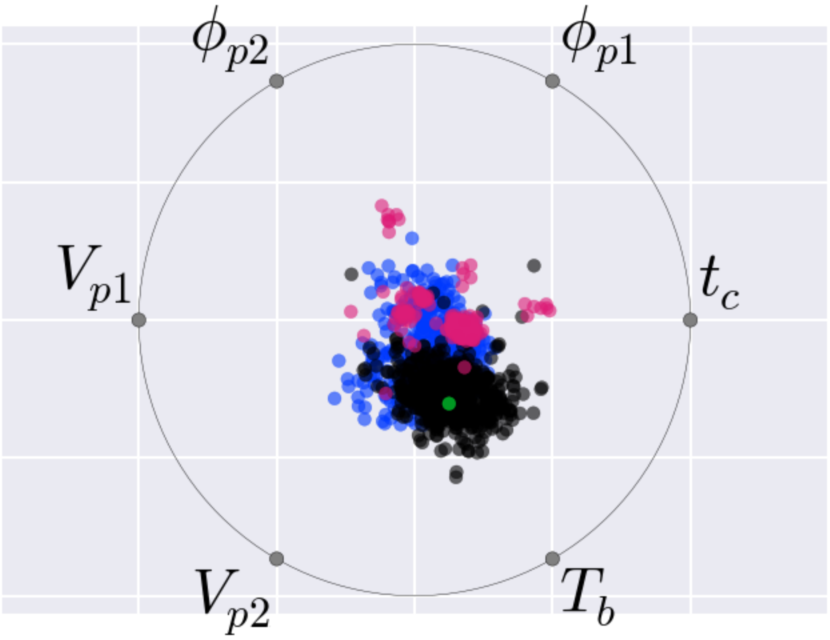


Figure 4.14: RadViz visualization for the design points explored for the different approaches: SbGA (black), EGO (blue), GA (red), and best solution (green).

The performance of the EGO algorithm and SbGA are comparable. SbGA are more direct towards maximising the Kriging predictive mean, and the performance with different initial datasets implies that the product purity is reasonably easy to approximate when the goal is to drive optimisation. The analysis we performed beforehand is in agreement with this claim. EGO is more conservative than SbGA because it relies on a trade-off between the predictive mean and variance. This more conservative approach can be slower, but more robust when the response surface of the predictive mean is devious. One of our five initial datasets shows that EGO could have a slower rate of convergence if too much emphasis is put on exploration rather than exploitation.

The results presented in Figure 4.15 indicate that a DP-PSA system should be designed as follows: long cycles to operate close to equilibrium, intermediate offset between piston 1 and 2, small chamber volume for piston 1 and large for piston 2, and the bed should operate at a high temperature. These design guidelines are reached by optimising the purity alone. The seemingly large gap in the value of ϕ_{p2} between the design point for initial dataset 3 and the other datasets is in fact quite small due to the offset angles being periodic with period 2π. The piston chamber volumes should be representative for the initial gas concentration, that is, a smaller volume for the chamber for the CO_2-enriched gas. The offset angles obtained are comparable to the ones in [AFR02], where the offset angles 0° and 220° for piston 1 and 2, respectively, lead to the best separation performance for air separation.

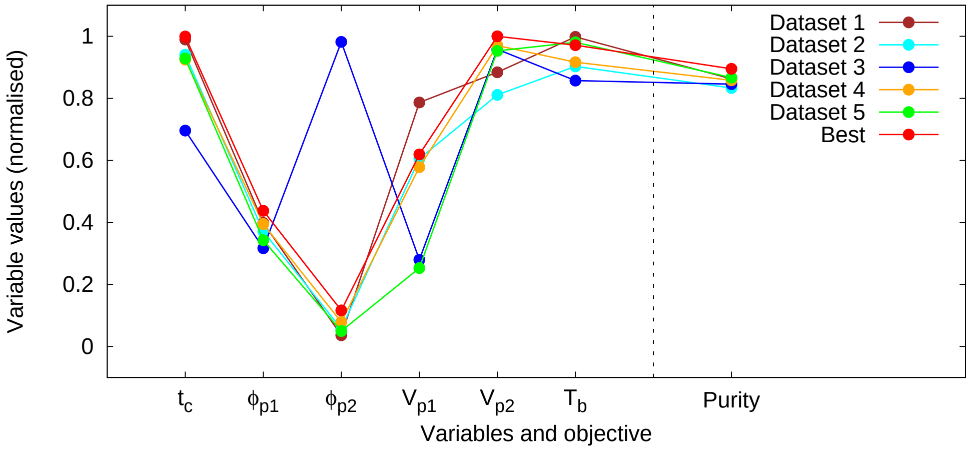


Figure 4.15: parallel coordinate system for the design solutions obtained using SbGA, when 100 design points have been explored. The solutions from the five different initial data sets (of size 16) are presented. Moreover, the best solution found (89.5 % Purity) is included for comparison.

The numerical results presented in Figure 4.15 indicate that a DP-PSA system should be designed as follows: use longer cycle times to operate close to equilibrium, intermediate offset between piston 1 and 2, the piston chamber volume should be small and large for piston 1 and 2, respectively, and the column should be at a high temperature. These design guidelines are reached by optimising the purity alone. The influence of further objectives such as recovery and power requirements will be investigated in the following chapters.

4.8 Concluding remarks

We have presented a surrogate-based optimisation, SBO, procedure, with step-by-step instructions on how to modify the procedure to achieve better performance for the problem at hand. The different surrogate-based criteria in SBO performed similarly, and they outperformed the stand-alone GA. This is the first SBO method applied to the PSA optimisation problem that sequentially update the surrogate model over a large number of iterations. Also, the accuracy of the Kriging surrogate model for a PSA model has not been assessed before. This approach targeted in a sequential manner the most interesting regions of the design space by using the surrogate model of the PSA model. The SbGA performed well for the dual-piston PSA case study, which gives us further confidence in the use of evolutionary algorithms in SBO. The next step is to take the SBO further and explore its use for more challenging design problems, with design constraints and multiple objectives. The SBO method identified designs that are close to the optimal performance in terms of CO_2 purity ($\approx 80\%$) with only 100 PSA simulations, whereas the conventional GA required at least 500 PSA simulations. That is an improvement of at least a factor of 5.

Chapter 5

Surrogate-based Optimisation for Multi-Criteria PSA Design

To find the optimal PSA cycle configuration for a specific application can be computationally challenging. It requires advanced simulation strategies that can resolve the rigorous mathematical models describing the PSA process, as well as some procedure to overcome the high computational cost of running the resulting simulator over a variety of designs and operating conditions. This fact becomes painfully apparent when optimisation using conventional means is performed on the PSA simulator to identify appropriate design configurations for large-scale industrial applications [BJF05]. The decision making process in such circumstances often requires a higher level of detail in the optimisation problem formulation, typically with two or more design objective to be optimised simultaneously, while meeting some design specifications (given as inequality constraints). A set of design parameter values, such as for the valve constants, bed pressures, cycle times, flow rates, and bed geometry are varied over their value ranges by an optimiser obeying the design criteria. The most common design criteria for PSA are the purity and recovery of the product, the productivity and the system's power consumption. This means we are interested in solving problems with two to four design criteria. The purity and recovery are the technical attributes that should be as high as possible, and productivity and power consumption are often included in the design specification. Depending on the specific problem, some design criteria are naturally expressed as inequality constraints, whilst some are not. Goal programming is one way to turn one or more design criteria into inequality constraints with target values, often because of the need of reducing the computational complexity of the multi-objective formulation.

The main challenge for evolutionary multi-objective optimisation (EMO) is that the objective and constraint functions are often too expensive-to-evaluate. For each evaluation of these functions there is a need to run a time-consuming PSA simulation. Furthermore, the evolutionary multi-objective optimisation algorithm may need hundreds or even thousands of such evaluations. The simulation is however needed to reveal the complex relationships that exist between the PSA system's response (purity, recovery, etc.) and its design configuration (bed pressure, cycle times, valve constants, etc.). Chapter 4 demonstrated that the utility of the Kriging surrogate model can reveal the input-output relationship for the PSA model. Kriging models on some training data can make fast predictions for any untried point in the design space. The prediction quality is expected to improve with larger training data.

The use of surrogate models has made a huge impact on global optimisation practice [VSBT14]. Existing optimisation methods can utilise surrogate models, and their ability to make fast predictions at untried design points, to guide the search more efficiently. Surrogate models were initially introduced to solve problems with scalar outputs [JSW98, Jon01, ONK03, Jin05], but more recently appear in multi-objective optimisation [EGN06, KN08, VK10]. Chapter 5 demonstrated that a real-coded GA with the Kriging surrogate model greatly improved the GA method for the dual-piston PSA case example where the CO_2 purity was maximised.

The main goal of our work is to develop a surrogate-based optimisation (SBO) approach that will assist the selection process for the most efficient configuration of the PSA system with respect to multiple conflicting criteria. There is little difference in the use of surrogate models for single- and multi-objective optimisation, the formulation of the surrogate-based criteria is the most significant difference. The robustness become more of an issue in multi-objective optimisation. This chapter presents SBO for multi-objective optimisation, and constrained optimisation. Also, a few adjustments are proposed to make SBO more efficient when solving challenging PSA optimisation problems. The Utility approach to multi-objective optimisation is also considered, where the utility function is a multi-objective version of the surrogate-based "expected improvement" (EI) criterion (see Section 4.5), which can be solved with the real-coded GA. This EI-based method is called MOEGO.

5.1 Optimal design of PSA cycles

The design of PSA cycles can be formulated as a multi-objective black-box optimisation problem:

$$\begin{aligned} \arg\min_{\boldsymbol{x}} \quad & \boldsymbol{\phi}(\boldsymbol{q}(\boldsymbol{x})) \\ \text{s.t.} \quad & \boldsymbol{W}\left(\boldsymbol{q}(\boldsymbol{x})\right) \leq \mathbf{0} \\ & \boldsymbol{x} \in X \end{aligned} \tag{5.1}$$

where $\boldsymbol{\phi}(\cdot) = [\phi_1(\cdot), \phi_2(\cdot), \ldots, \phi_q(\cdot)]^T$ is a vector of q objective functions, $\boldsymbol{W}(\cdot)$ are the design and process constraints, $\boldsymbol{q}(\boldsymbol{x}) \in \mathbb{R}^n$ the vector of n (spatially-discretised) state variables, and $\boldsymbol{x} \in \mathbb{R}^p$ is the vector of p design variables. The aim is to identify the design configuration $\boldsymbol{x}$ that yields the best values of $\phi_1(\boldsymbol{x})$, $\phi_2(\boldsymbol{x})$, $\ldots$, $\phi_q(\boldsymbol{x})$ in a Pareto optimal sense.

For a typical 2-bed/4-step PSA system [RMER08] (see Figure 5.1), following the notation given in Chapter 2, the product purity and recovery of CO_2 in the product stream can be defined as:

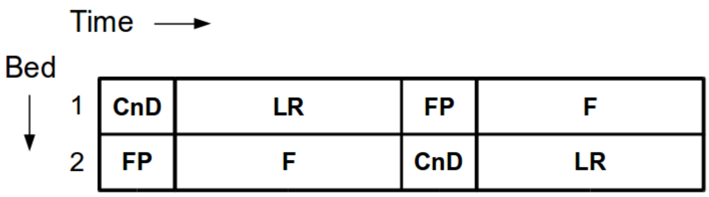
CO_2 Purity. The purity in the product stream are given as the number of moles passing through the feed units:

$$\text{Purity}_{CO_2} = \frac{\displaystyle\int_F u c_{CO_2}\bigg|_{z=L} dt}{\displaystyle\int_F \frac{up}{R_g T}\bigg|_{z=L} dt} \tag{5.2}$$

CO_2 Recovery. The recovery of CO_2 is accordingly calculated as follows:

$$\text{Recovery}_{CO_2} = \frac{\displaystyle\int_F u c_{CO_2}\bigg|_{z=L} dt - \int_{LR} -u c_{CO_2}\bigg|_{z=L} dt}{\displaystyle\int_{F,FP} \frac{up}{R_g T}\bigg|_{z=0} dt} \tag{5.3}$$

Here $\displaystyle\int_F$ is integration over process step F, and so on.



1	CnD	LR	FP	F
2	FP	F	CnD	LR

Figure 5.1: Schematic of a 2-bed/4-step PSA. The process steps are: feed pressurisation (FP), feed (F), countercurrent depressurisation (CnD), and light reflux (LR).

Systems that yield a high product purity, tend to perform worse in terms of the product recovery [FFB09b], and vice versa. Similarly, an increase in productivity usually results in a raise in the system's power consumption [HMN$^+$13]. According to [ZWX08] an increase in product recovery typically requires more power usage. Similarly, the product purity is said to be positively correlated with the system's power consumption. Further analysis of the conflicting behaviour between purity, recovery, and power consumption is needed. An expression for the system's power consumption has been omitted, but will be defined for our main case study in Chapter 6. To obtain any reliable power consumption estimate many assumptions need to be made that vary on a case by case basis [HLAW06]. Because there does not exist a single design configuration $\boldsymbol{x}$ that generates the maximal possible product purity and recovery of the system, simultaneously, the use of evolutionary multi-objective optimisation (EMO) is warranted if we only could overcome the high computational cost of objective functions, in our case time-consuming PSA simulations.

EMOs are population-based and can be used to identify the set of "non-dominated" solutions, called the Pareto front. One solution is said to dominate another if the solution is not worst for any of the objectives and better for at least one objective than the other. This means that more than one solution can be undisputed as the best solution in the Pareto optimal sense. EMO has been presented in Section 3.3.

5.2 Surrogate based criteria for multiple objectives and constraints

Surrogate modelling has become the technique of choice to overcome the computational barriers caused by expensive objective and constraint functions [FK09]. For multi-objective optimisation, the advances made are mostly for surrogate-based EMOs [BBLP05, VK10]. By convention, this is often done by extending already existing surrogate based criteria to the multi-objective setting [FK09]. An alternative strategy is to reduce the multi-objective optimisation problem to a single objective one and solve it within the surrogate-based optimisation (SBO) framework. The utility approach or the goal programming approach can be used for this end, see Section 4.5.

The SBO procedure (presented in Section 4.5) performs for set number of iterations, and in each iteration, one or multiple design points are selected based on some chosen surrogate based criteria. One of the strengths of the SBO framework is that *only* the surrogate based criterion needs to be adapted to this new multi-objective setting, the remaining steps can be left untouched.

In this section we suggest some efficient surrogate based criteria for multi-objective optimisation, and for constrained optimisation, that can be used for the PSA design problem.

5.2.1 Surrogate based evolutionary multi-objective optimisation

The main challenge for surrogate based criteria in EMOs are that the evolutionary computations are population based and return non-dominated Pareto solutions. The naïve approach is to directly apply the EMO on the surrogate model, and then select one or several non-dominated solutions for further consideration. The main concern with this criterion is that the non-dominated solutions generated by the surrogate model are not put in comparison with the solutions of the current Pareto front. This spoils the multi-objective performance, in terms of solution diversity, richness, and coverage (see Section 3.2). A more efficient approach is thus to utilise the current Pareto front with the surrogate-based Pareto front when selecting new design points to be evaluated. See Figure 5.2, for an illustration of this procedure using NSGA-II on the Kriging model. In this way the EGO procedure obeys the nature of the multi-objective optimisation.

5.2.2 MOEGO: a multi-objective expected improvement criterion

The expected improvement (EI) criterion (see Section 4.5) used in the Kriging-based efficient global optimisation (EGO) method has attracted much attention in the global optimisation community [VSBT14], and is considered to be one of the most efficient, as well as reliable, search criteria that guarantees global convergence [Loc97]. Many attempts have been made to follow up on this success by extending EI to multi-objective optimisation. One way is to apply the EI criterion to each of the objective functions, independently, and then employ some multi-objective optimisation method (such as EMO) [JO05]. Another is to use the weighted aggregation approach and then solve it with some single-objective optimisation method (ParEGO [Kno06]; WS-EI [LZT$^+$07]). But perhaps the MOEGO criterion by Emmerich et al. [EDK11] is the most pleasing from a statistical perspective.

Consider the multi-objective black-box optimisation problem (5.1), then we want to identify the design configuration $\boldsymbol{x}$ optimising the set of multiple objective functions. To keep the notation used in Section 4.5, we denoted the vector of p objective functions by $[y_1(\cdot), y_2(\cdot), \ldots, y_q(\cdot)] \in X \subset \mathbb{R}^p$. The MOEGO is based on the assumption that each of our objective functions $y_1(\cdot), y_2(\cdot), \ldots, y_q(\cdot)$ can be modelled independently via Kriging as a prob-

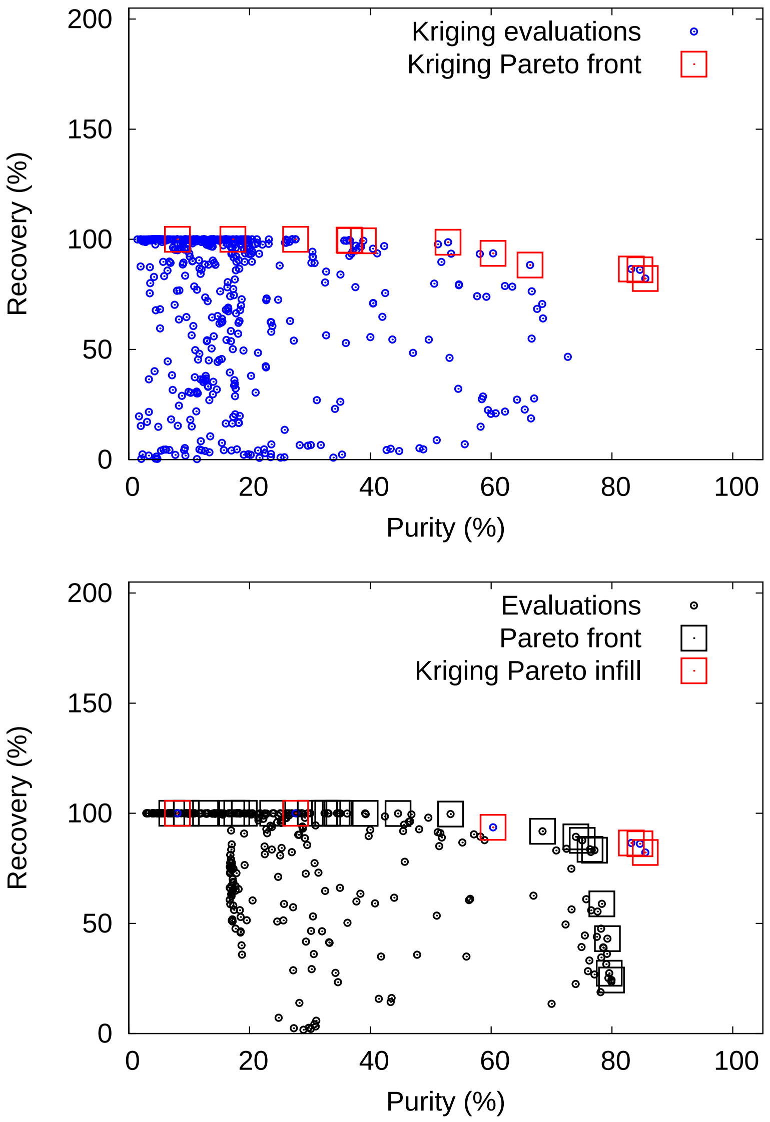


Figure 5.2: From the Pareto front computed with NSGA-II on the Kriging model (Top), the solutions that also are non-dominated by the Pareto set of the current set of PSA simulation responses (Bottom) are considered to be promising candidates for design point selection.

ability distribution. Under this assumption the method is sequentially selecting points $\boldsymbol{x}$ that are expected to generate the largested improvement in the dominated hyper-volume (S-metric) of the Pareto front [EDK11]:

$$\mathrm{SbC}(\boldsymbol{x}, \cdot) = E_{\bar{Y}} \left(S(\boldsymbol{Y}^P \cup \bar{Y}(\boldsymbol{x})) - S(\boldsymbol{Y}^P) \right), \tag{5.4}$$

where $\boldsymbol{Y}^P$ is the Pareto front of $\boldsymbol{Y}^D$, and $\bar{Y}$ is the vector of the random variables $Y_1(\cdot), Y_2(\cdot), \ldots, Y_q(\cdot)$ that are jointly normally distributed. The definition of the S-metric is given in Section 3.2. By convention the normal distribution is assumed for the probability distributions of the objectives $y_i(\cdot)$. Also, as shown in [EDK11], this criterion can be computed exactly using the same tools as for the EI criterion, namely the cumulative and probability density functions for the normal conditional posterior distribution derived from the Kriging model. One drawback with MOEGO is that the S-metric is known to be computationally prohibitive for more than four objectives [WHBH06].

5.2.3 Surrogate based criteria for constrained optimisation

The criteria discussed so far have not shown any consideration to optimisation problems with expensive constraints. This is mainly because we only recently begin to see work devoted to the development of more efficient surrogate based criteria for solving such problems [PHFK10, PFKH12]. See [FSK08, FK09] for more on the use of surrogate models for constrained optimisation.

PSA design problems are often formulated as a single objective optimisation problem for the purity or recovery. This is often because either the purity or the recovery is specified as an inequality constraint to meet some design requirement, or to reduce the computational time (that is, goal programming). Both purity and recovery are quantities that are computationally expensive to obtain, but for which can be computed together through a single simulation, without any extra effort. In the surrogate-based optimisation framework, when purity is used as the objective, and the recovery is used as a constraint, or vice versa, both are represented by mutually independent surrogate models, typically upon the same design points. Any constraint that is considered computationally expensive, should be modelled by a surrogate. The inexpensive ones would just be treated as usual. Henceforth we will refer to the surrogate models of the constraints as “constraint surrogates.”

For the surrogate-based GAs we are interested in the Deb’s constraint handling method

[Deb00]. In this approach the fitness function, here denoted by $f(\boldsymbol{x})$, is penalized for design point $\boldsymbol{x}$, when any of the constraints are violated:

$$f(\boldsymbol{x}) = \begin{cases} \hat{y}(\boldsymbol{x}) & \text{if } \hat{w}_i(\boldsymbol{x}) \geq 0, \forall i = 1, 2, \ldots, n \\ \max\left\{\hat{y}(\boldsymbol{X}^D)\right\} + \sum_{i=1}^{n} |\hat{w}_i(\boldsymbol{x})| & \text{otherwise} \end{cases}$$

where $\hat{w}_i(\boldsymbol{x})$ is the constraint surrogates of the ith row of the n design constraints, represented by $\boldsymbol{W}(\boldsymbol{x}) \leq \boldsymbol{0}$ in the optimisation problem. Observe that the maximal value in the vector $\hat{y}(\boldsymbol{X}^D) : X^m \to \mathbb{R}^m$ is introduced to ensure that any infeasible design solution is strictly dominated by a feasible one. This type of one pass penalty function relies to a high degree on the global accuracy of the constraint surrogates. Surrogate constraints are required to be accurate, as Deb's constraint handling method is sensitive to the actual values of the constraint functions [PFKH12]. This could be understood from the preferential system adopted: solutions that violate the constraints, are given a lower priority than any of the solutions that satisfy them.

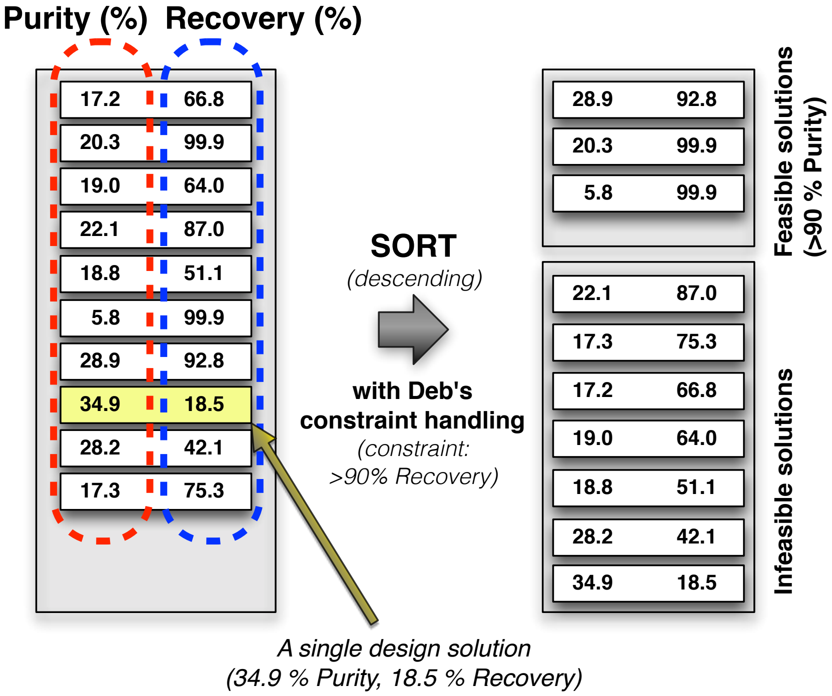


Figure 5.3: An illustrative example of Deb's constraint handling, where a set of ten design points is sorted with the aim to maximise the product purity, while the product recovery is specified as an inequality constraint (recovery > 90 %).

5.3 Premature convergence

One challenge in SBO is to ensure the selection process of new design points *is not only* guided by Kriging's predictive mean, but also by the desire to improve the surrogate's global approximation quality. The SBO with the predictive mean criterion can cause premature convergence [Jon01]. The stagnating convergence may occur when the surrogate prediction leads to an excessive point selection in an already well-explored region of the design space, because the SBO lacks the desire to explore the regions of the design space of higher uncertainty.

The SBO methods that rely on either the EI, PI or the MV are exhibiting exploration to regions of higher uncertainty, as opposed to the standard surrogate based evolutionary multi-objective optimisation. To circumvent this issue, the combination of surrogate based criteria can be employed [VK10], where the surrogate based criterion adoped can be different from one SBO iteration to the next.

Kriging is privileged with its predictive variance $\hat{s}^2(\boldsymbol{x})$, Eq. (4.4), that could estimate the "uncertainty" over the design space. To allow the surrogate model to learn, and become a better global approximative model, we turn to Active Learning (AL) from machine learning (see, e.g., [RW06]). One such technique is Active Learning MacKay (ALM) [Mac92, GL09], where the goal is to identify the design point that maximises the Shannon entropy. The Shannon entropy is from information theory and measures the uncertainty in a random variable [CT06]. It turns out the Shannon entropy taken at a candidate point is proportional to the Kriging variance at the point ($\propto \hat{s}^2(\boldsymbol{x})$), which makes this optimisation problem equivalent to finding the point in the design space that maximises the predictive variance. A similar technique is Active Learning Cohn [Coh96], where the goal is to the maximise the integrated variance reduction over the entire design. The reduction in variance is typically mesured point-wise over the design space by deploying a large number of reference points at which the variance reduction is measured. The variance reduction at a point $\boldsymbol{x}$ can be estimated by using the Kriging's predictive variance $\hat{s}^2(\boldsymbol{x})$, see calculation in [GL09]. In the early stage of the selection process, ALM tends to select many points on the boundaries of the design space. Early on the boundary points are less informative than interior points [KSG08]. To put this in perspective, for a regular grid with m^p points, where p is the number of design variables, $m^p - (m-2)^p$ of them are boundary points. This implies that the percentage of the total points that are on the boundary, $100(1-(1-\frac{2}{m})^p)$, grows rapidly with the dimension p. For $p = 6$, the design space dimension, with $m = 10$, almost 74% of the points lie on the boundary of the design domain.

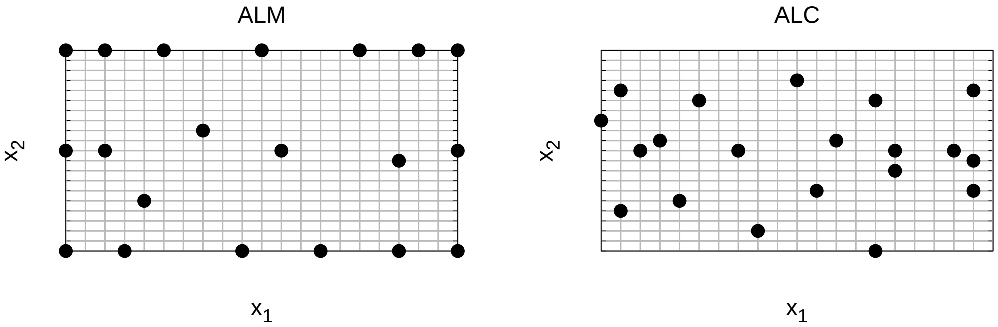


Figure 5.4: Standard deviation estimate with Kriging over the design space for Branin function. The design sites are marked in green. The correlation lengths are $\boldsymbol{\xi} = (0.37, 1.46)^T$, with $\tau = 10^{-11}$. Left: ALM. Right: ALC.

ALC on the other hand has been shown to abstain from choosing points directly on the boundaries. See Figure 5.4 for an illustration of ALM and ALC designs for a popular 2D case example, the Branin function. Over a wide range of different problems, restricted to small sample sizes, ALC has been shown to perform better than ALM in terms of the mean squared error (MSE) [SWGO00]. As the amount of training data increases for the Kriging, the difference between ALM and ALC tends to become smaller. ALM is the significantly cheaper option, and thus the preferred choice to improve the Kriging prediction quality over the design space. The ALM or equivalent has been adopted in [VK10, SSW$^+$11].

Constraint surrogates are demanding because they need to be good global approximations, otherwise the SBO procedure can become very inefficient, see Section 5.2.3. The ALM approach can be a remedy to overcome such issues in the early stage of the optimisation, when the training data is small.

It has been demonstrated that alternating between optimising the Kriging's predictive mean and the ALM criterion is a reliable strategy [VK10].

5.4 Failure in PSA simulator response

Computer simulations of PSA processes are typically prone to numerical instability that can lead to simulator failure. A simulator failure here means that no output, or an error flag, is returned to the optimiser. The causes of the simulation failures are often difficult to pinpoint, but often a failure to solve the DAEs. For PSA simulation the failures can be a consequence of the difficult dynamical behaviour that may occur in the adsorbent bed, or in some cases the issue may even be a simple error on the implementer's part. Some simulation approaches such as those relying on Newton's method are also known to be unstable because of the calculations involved [BJF05].

Even though the PSA simulation is successful for one design point, another point may cause failure. The most robust simulation approach is probably the successive substitution, but even this approach is suffering from numerical instability because of the challenge of solving the PSA simulation problem over a wide range of model parameter choices. Hence, it is expected that some of the design points selected during the optimisation for PSA simulator evaluation can cause simulation failure. If no precautionary measures are taken to address this issue, most optimisation methods are terminated on objective function failure, in particular those that heavily rely on gradient information.

For the SBO approach, we wish to incorporate the knowledge about the design points for which a failure has taken place. Although no output of the PSA simulator is observed, these design points should be included in the training data, more specifically, their location in the design space, $\boldsymbol{X}^{D^F}$, where $D^F = \{\boldsymbol{X}^{D^F}, \boldsymbol{Y}^{D^F}\}$ is the training data representing the failed simulator runs. D^F is informative about the possibility of PSA simulator failure at, or in close proximity to, the design points already known to cause failure. We propose that $\boldsymbol{X}^D$ should be extended with D^F when using the Kriging's predictive variance (4.4). The dependence the Kriging's predictive variance (4.4) has on $\boldsymbol{Y}^D$ is only through $\hat{\sigma}^2$, which in the implementation will be unchanged. In other words, the Kriging's predictive variance can be calculated, using the covariance parameters $\hat{\boldsymbol{\xi}}$ and $\hat{\sigma}^2$ estimated by D, with the extended training data $D \cup D^F$, by assigning dummy values to the unknown responses $\boldsymbol{Y}^{D^F}$, e.g., $\boldsymbol{Y}^{D^F} = (1, 1, \ldots, 1)^T$. Kriging's predictive mean, in contrast to the predictive variance, is affected by the choice of dummy values, and thus should be performed with D.

By providing this set of data to the Kriging variance (4.4) the "uncertainty" is modelled more accurately, because we have evaluated the design points even though the response value is not accessible due to the simulator failure. SBO methods that are assisted by e.g. ALM or EI, and use this implementation, can avoid selecting design points that belong to the vicinity of any of the design points that already have caused PSA simulator failure.

5.5 Transformed Kriging for purity and recovery computations

The product purity and recovery are both represented as ratio values, confined to the interval $[0, 1]$, in the PSA simulator. These are physical constraints not enforced by the surrogate model, because the Kriging surrogate prediction by definition is made on the real line (see, e.g., [RW06]). This means the Kriging predictor can make predictions higher than 1. By respecting these physical constraints, the performance of the SBO is improved. The perhaps most straightforward approach to enforce the bounds are to set the Kriging prediction values that fall outside the feasible regime to the extreme values of the bounds. Our concern with this approach is the manipulation of the Kriging response. We instead propose the following transformation of the training data supplied to the Kriging surrogate model:

$$\boldsymbol{U}^D = \log\left\{\frac{\boldsymbol{Y}^D}{(1+\epsilon) - \boldsymbol{Y}^D}\right\}, \tag{5.5}$$

for some tiny $\epsilon > 0$. This extends the feasible interval from $[0, 1+\epsilon]$ to the real line $\mathbb{R}$, which suits the Kriging model. To reverse the transformation (also known as back-transformation) we use $\hat{y}(x) = \dfrac{(1+\epsilon)\exp\{\hat{u}(x)\}}{1+\exp\{\hat{u}(x)\}}$, where $\hat{u}(\cdot)$ is the transformed Kriging predictor. Here ϵ is a small perturbation made to allow $y = 1$. Also ϵ is needed to ensure (5.5) is defined when there exists a small numerical error produced by the PSA simulator, e.g. if the product purity returned is 1.0001 then ϵ should at least greater than 0.0001. The ordering of objective function values is preserved under this transformation, that is, if one response value dominates another in the original space, this response dominates the same one in the transformed space. Another transformed Kriging approach is the log-normal Kriging (see, e.g., [Cre89, DR07]). The training data supplied to fit the Kriging is log-transformed in order to reduce the influence of the few high values observed in data generated from the lognormal distribution. In Figure 5.5 it can be seen the optimisation performance on the Kriging model under transformation (5.5) is improved and avoids non-physical Kriging responses.

5.6 Summary

In this chapter the PSA design problem has been formulated for multiple design criteria, such as product purity, recovery, and power consumption. We proposed some modifications of the surrogate based criteria to adapt the surrogate based optimisation (SBO) framework for this

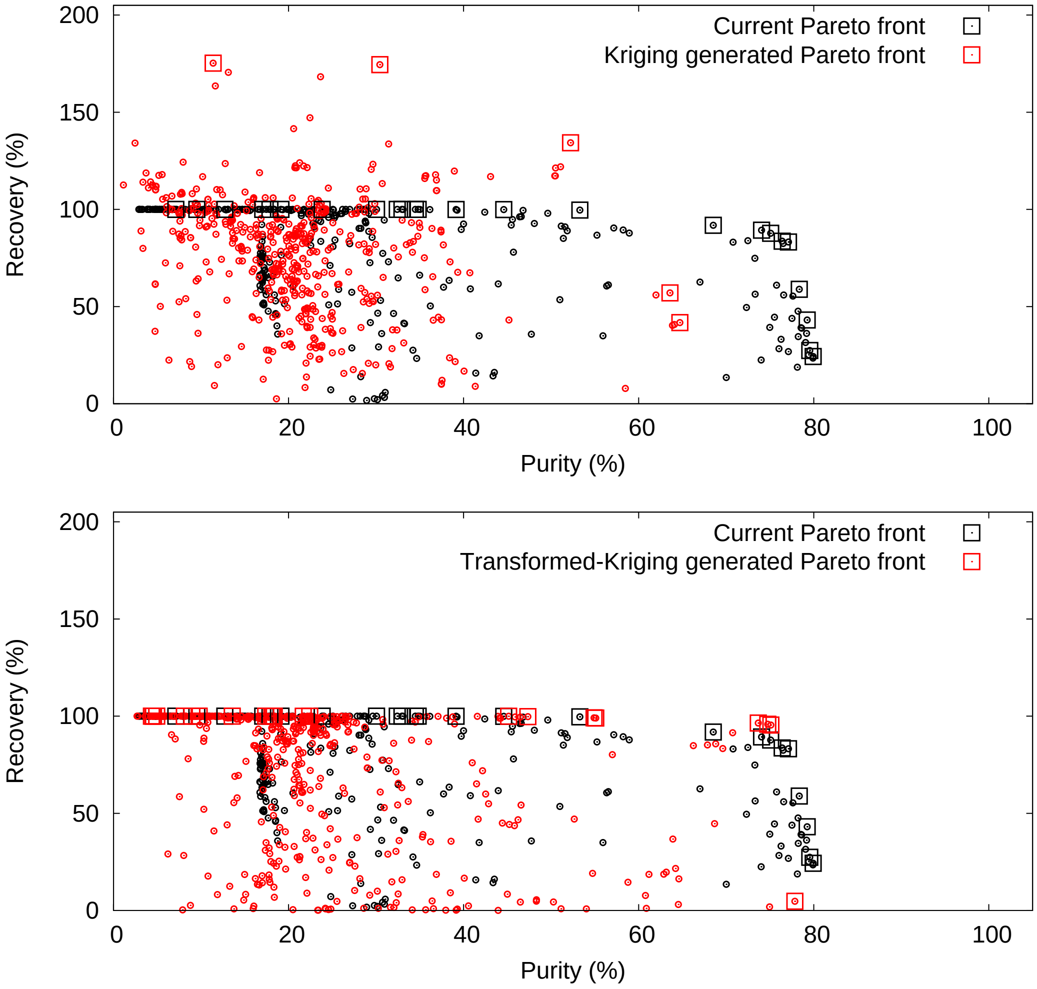


Figure 5.5: Top: The Pareto front approximation generated by responses from the Kriging predictor. Bottom: The Pareto front approximation generated by the transformed-Kriging. This illustates the benefit of using Kriging in the transformed space when performing multi-objective PSA optimisation, when there are physical constraints for any of the objective functions.

multi-criteria setting. Two surrogate based criteria were discussed, one for evolutionary multi-objective optimisation, and one that extends the expected improvement criterion to multiple objectives by using the S-metric that measures the hyper-volume dominated by a set of non-dominated solutions. We also identified and proposed solutions to some SBO issues that are particularly relevant to the PSA problem:

- Many PSA design problems in the literature are using design constraints, that could be computationally expensive to compute. We proposed the use of constraint surrogates together with the use of Deb's constraint handling method.
- The use of the Kriging model to predict the response of the PSA simulations has been successful, see Chapter 4. However, there are circumstances were the SBO optimiser can lead to premature convergence. We here proposed the use of an active learning method called ALM that selects the design points of largest Kriging variance. The ALM criterion should be used for some SBO iterations instead of the original surrogate based criterion, to promote exploration in regions considered by the Kriging model to be most "uncertain."
- The PSA simulators are known to be fragile due to the difficulty to numerically solve the coupled PDAEs involved. This means that some of the design configurations attempted during the optimisation run could lead to PSA simulation failure. PSA simulation failures should be accounted for by the Kriging model, otherwise the Kriging variance is not representing the actual "uncertainty." If the design points causing simulator failure are provided, the Kriging model can identify those points as known and not revisit them.
- The product purity and recovery, when represented as ratio values, are restricted to the interval $[0, 1]$. However, the Kriging is making predictions that are not enforcing these constraints. The Kriging model can therefore generate values above 1. To avoid this, we have proposed a transformation of the ratios to the entire real line, and then use Kriging for SBO in this transformed space. To our knowledge this transformation has not been used for process design. Same with the issue of physical constraints on the output in the multi-objective setting has not been highlighted before.

All the strategies presented in this chapter will be employed in the next chapter for a challenging PSA case study for CO_2 separation. The combination of these modifications to the SBO method promotes robustness, and is new framework for robust multi-objective optimisation with surrogate models for process design.

Chapter 6

Surrogate-based Multi-Criteria Design: CO_2 Capture by PSA

6.1 Introduction

A two bed, six step PSA system is the case study used to show that fast, and robust, optimisation can be achieved with the surrogate based optimisation (SBO) procedure, demonstrated in Chapter 4. This SBO procedure is here tailored to a typical multi-criteria PSA design problem using the strategies presented in Chapter 5.

As discussed in Chapter 3, some recent work on PSA design for carbon capture have yield frameworks for flowsheet design to ease the search after interesting PSA designs for CCS, in particular when applied for post-combustion in coal-fired power plants. The frameworks and the expertise are in place, but the optimisation strategies could benefit from the performance boost of SBO. The SBO framework is a versatile speed-up procedure that can be applied on top of any of the design and optimisation formulations proposed in literature for CO_2 capture.

PSA is a promising CO_2 separation technique with relatively low operating and capital costs, and thus of great interest to "Carbon Capture and Storage." For disposal of CO_2 in deep ocean, ageing oil fields, and gas reservoirs, it would be necessary to enrich the CO_2 up to a level of 99% for the compression and transportation costs to be satisfactory. So when the feed is high in CO_2 concentration, above let us say 25%, the PSA design problem is not challenging, given that we want above 70% recovery, using a one-stage PSA [CKY$^+$95]. Unfortunately with flue gas of the coal fired power plant, the CO_2 concentration can be as low as $10 - 15\%$, which makes it difficult to recover 99% CO_2 with a single PSA unit. Hence, a two-stage PSA is required to enrich CO_2 from flue gas to acceptable purity levels [IOA$^+$96].

The application of PSA for the separation of CO_2 from flue gas is indeed a very challenging design problem, and the two-stage PSA process is perhaps a more viable strategy than using a single PSA system. In the two-stage PSA process the first stage produces a CO_2-enriched product of roughly $50-60\%$, and the second-stage brings the performance up to the desired CO_2 level of about 95%. The design problem is foremost an optimisation problem of the technical performance of a 2-bed/6-step PSA system design structure with fixed process step. One of the greastest challenges of applying the the CO_2 capture unit to the power plants is the high power consumption associated with it. Hence, we include the power consumption as an objective for our analysis. By using this example study, the goal is to address the first-stage of two-stage PSA process, motivated by the results presented in [PBKC02] for the same case study.

Previous studies have explored the effect of the different process parameters (such as feed flow rate, purge ratio, and product flow rate) on the PSA performance, both experimentally and through detailed computational simulations, see, e.g., [CKY+95, PBKC02, CKY+03, ZWX08]. Either by varying one parameter at the time, or using scatter plots with one output of interest against one of the design variables. Both of these approaches may not capture the global sensitivity.

We are interested to investigate the effects of the design variables and cycle configuration of the first-stage PSA system for the two-stage strategy. For CCS we are required to achieve product recovery of at least 90%, with as high product purity as possible. For a two-step system, around 95% is desired in each. With a single step with a two-bed PSA system, the work of Liu et al. [LGL+11] achieve 58% product purity for 93% recovery [LGL+11] from 15% CO_2-85% N_2, but with silicalite and not the more established Zeolite 5a. Some of the earlier optimisation studies for CO_2 capture include [CKY+03, KSB03, KSB05]. Ko et al. [KSB05] concluded that fractionate vacuum PSA under high temperature is more advantagerous than PSA in terms of product purity and recovery, but with higher average power consumption. Choi et al. [CKY+03] identified the optimal adsorption step time and reflux ratio for a given PSA system.

In this chapter we will use the SBO procedure for a variety of design problems using a challenging 2-bed 6-step PSA system as the underlying case example.

6.2 The 2-bed/6-step PSA system

The PSA cycle considered is defined by the following 6 steps: feed pressurisation (FP), feed/adsorption (F), light end equalisation (LEE), countercurrent depressurisation (CnD), light reflux (LR), and ligh end pressurisation (LEE). The LEE steps are the pressure equalisation.

This PSA cycle is a Skarstrom with a pressure equalisation step and has been investigated before in [PBKC02] for recovering CO_2 from flue gas.

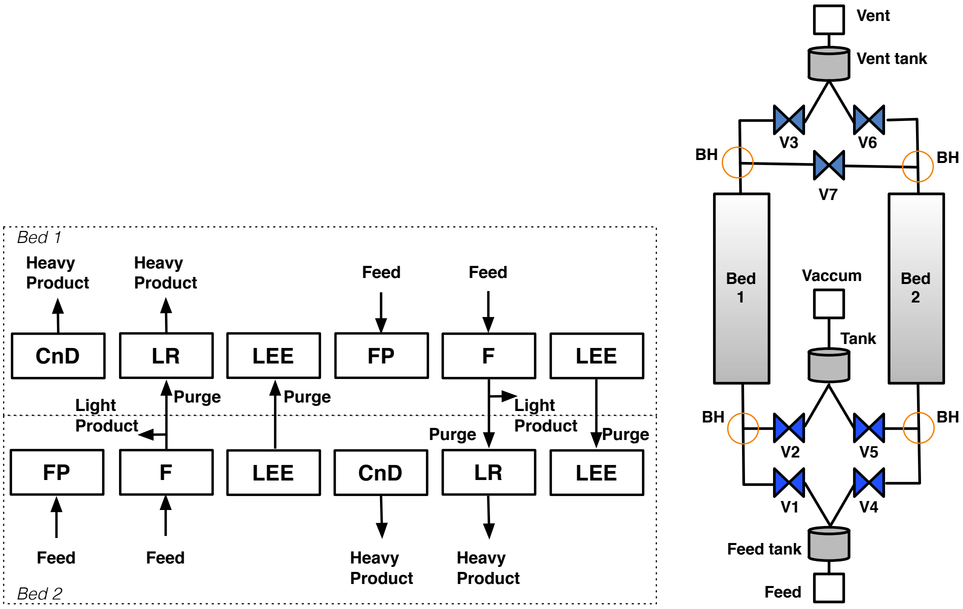


Figure 6.1: 2-bed/6-step PSA Skarstrom Cycle

While there are many adsorbents for CO_2 capture, in this study, we consider adsorption beds packed with zeolite 13x pellets. Zeolite 13X is perhaps the most popular adsorbent material for separation of CO_2 from flue gas [KSB03]. This adsorbent is a good first approach to investigate the PSA for CO_2 capture [ZWX08].

The schematic is given in Figure 6.1 (Left). The transition between the process steps over time are regulated by the stem positions of the valves. The valves involved in the system are presented in Figure 6.1 (Right). The system is symmetrical with the axis of symmetry going through the feed and vent units. On both sides of of each bed is a bed header which is usually used to ensure a homogeneous flow distribution in the bed. The units labelled 'Feed', 'Vaccum' and 'Vent' provide the boundary conditions for the PSA system. Briefly, the 'Feed' unit is an inlet which provides the gas mixture to separate; the 'Vacuum' unit is an outlet which provides vacuum pressure for the purge and blowdown steps; the 'Vent' unit is an outlet at atmospheric pressure. These three units are refered to as feed units. The tanks next to the feed units are buffering the flow so that the pumps can be operated continuously. The tanks and bed headers are connected by valves which control the flow rates in the system and thus the cycle steps.

With the pressure equalisation step the CO_2 purity can be enriched [PBKC02], and it is expected this will only lead to a small penalty in power consumption. In this case, the light reflux step is used to produce a higher CO_2 purity. We would expect that with higher CO_2 purity, more power is consumed at the vacuum pump.

The stem positions open and close at specific times during the course of a cycle to control the PSA operation. See Table 6.2 for the stem positions for the different process steps of this 6-step PSA Skarstrom cycle. Here 0 means that the valve is closed, 0.5 half open, and 1 fully open. The PSA cycle is performed through the coordinated operation of the 7 valves. The feed unit supplies a gas mixture of constant pressure, temperature, and feed composition, and therefore held at the initial operating conditions. The flow rate on the other hand is set to

$$F = \begin{cases} F_i, & \text{if } \left|\dfrac{p-p_n}{p}\right| > 0.01, \\ \frac{1}{2}F_i\left(1+\cos\left\{100\pi\left(0.01-\left|\frac{p-p_n}{p}\right|\right)\right\}\right) & \text{otherwise,} \end{cases}$$

where F_i is the prescribed flow rate and p and p_n are the pressures in the feed unit and the neighbouring unit, respectively. This means that when the relative pressure difference is lower than 1%, the flow rate will approach zero in a smooth way as p approaches p_n. This choice is out of computational convenience.

Table 6.1: Stem positions for the valves amounting to the different steps. The numbers represent the fraction of the corresponding valve which is open.

Steps for bed 1	V1	V2	V3	V4	V5	V6	V7
FP	1	0	0	0	1	0	0
F	0	0	r_3	0	1	0	r_7
LEE	0	0	0	0	0	0	1
CnD	0	1	0	1	0	0	0
LR	0	1	0	1	0	r_6	r_7
LEE	0	0	0	0	0	0	1

Table 6.2: System parameters for the PSA unit.

Parameter		value	unit
L_b	Bed length	0.12	m
r_b	Bed radius	0.0175	m
ϵ_b	Bed void fraction	0.387	-
V_{CH}	Bed header volume	5.2×10^{-5}	m^3
r_p	Pellet radius	9.15×10^{-4}	m
ϵ_p	Pellet void fraction	0.35	-

The design problems that will be considered are all using one or several of the following performance indicators:

CO$_2$ Purity. The purity of CO_2 in the product stream during cycle k are calculated from the number of moles passing through the feed units:

$$\text{Purity}_{CO_2}^k = \frac{n_{vac,CO_2}^k - n_{vac,CO_2}^{k-1}}{\sum_{j=1}^{N_c}(n_{vac,j}^k - n_{vac,j}^{k-1})}. \tag{6.1}$$

CO$_2$ Recovery. Similary to the calculation of the purity, the recovery of CO_2 is

$$\text{Recovery}_{CO_2}^k = \frac{n_{vac,CO_2}^k - n_{vac,CO_2}^{k-1}}{n_{feed,CO_2}^k - n_{feed,CO_2}^{k-1}}. \tag{6.2}$$

Power Consumption per Mole. The power consumption per mole is the work done by all boundary units, that is, feed, vacuum and vent, and is calculated by:

$$\text{Pow} = \begin{cases} \frac{\gamma}{\gamma-1} R_g T F \left\{ \frac{p_n}{p_{atm}}^{\frac{\gamma-1}{\gamma}} - 1 \right\}, & \text{if } p_n > p_{atm} \text{ and } F < 0, \\ \frac{\gamma_n}{\gamma_n-1} R_g T_n F \left\{ \frac{p_{atm}}{p_n}^{\frac{\gamma_n-1}{\gamma_n}} - 1 \right\}, & \text{if } p_n < p_{atm} \text{ and } F > 0, \\ 0 & \text{otherwise.} \end{cases} \tag{6.3}$$

$$\frac{dW}{dt} = \text{Pow}, \quad W(0) = 0. \tag{6.4}$$

Here $\gamma = \frac{\tilde{c}_P}{\tilde{c}_V}$ is the ratio of the molar heat capacities at constant pressure and constant volume. The number of models are calculated through the following ODEs

$$\frac{dn_i}{dt} = \frac{F + |F|}{2}\frac{c_{i,n}}{c_{T,n}} + \frac{F - |F|}{2}\frac{c_i}{c_T}, \quad i = 1, 2, \ldots \tag{6.5}$$

with $n_i|_{t=0} = 0$.

The power consumption is a function of the molar flow rate, compression ratio and the inlet gas temperature. The contribution of the vent is zero as it is at atmospheric pressure.

The goal will later be to compare the specific power consumption to the system's performance in terms of product purity and recovery.

6.2.1 Model equations

The bed equations are given in Table 6.3, and the nomenclature is given in Table 6.4.

Temperature gradients within the pellet were neglected, and heat transfer from the pellet was entirely due to the fluid film resistance. The adsorber wall energy balance was also included in the model since it has been experimentally observed that the accumulation of heat in the bed wall affected the bed dynamics.

Table 6.3: The model equations for the modelling of the adsorption bed.

Component mass balance (axial dispersed plug flow model):

$$\frac{dc_i}{dt} + \frac{1-\epsilon_b}{\epsilon_b}\frac{d\bar{Q}_i}{dt} + \frac{\partial(uc_i)}{\partial z} + \frac{\partial J_i}{\partial z} = 0 \tag{6.6}$$

$$\frac{d\bar{Q}_i}{dt} := \epsilon_p \frac{dc_i^m}{dt} + (1-\epsilon_p)\frac{d\bar{q}_i}{dt} = k_i^p \frac{A_p}{V_p}(c_i - c_i^m)$$

Energy balance for the adsorbate in the gas phase:

$$\begin{aligned} \epsilon_b \frac{d\check{U}_f}{dt} + (1-\epsilon_b)\frac{d\check{U}_p}{dt} + \epsilon_b \frac{\partial(\check{H}_f u)}{\partial z} + \frac{\partial J_T}{\partial z} \\ + \sum_{i=1}^{N_c} \frac{\partial(J_i \tilde{H}_i)}{\partial z} + h_w \frac{A_c}{V_c}(T_f - T_w) = 0 \end{aligned} \tag{6.7}$$

Energy balance for the adsorbate in the solid phase:

$$\frac{d\check{U}_p}{dt} := \epsilon_p \frac{d\check{U}_{p,f}}{dt} + (1-\epsilon_p)\frac{d\check{U}_{p,s}}{dt} = h_p \frac{A_p}{V_p}(T_f - T_p) \tag{6.8}$$

Energy balance in the bed wall:

$$\rho_w \hat{c}_{p,w} \frac{dT_w}{dt} = -h_w \frac{A_c}{V_w}(T_w - T_f) - U\alpha_{wl}(T_w - T_\infty) \tag{6.9}$$

LDF equation:

$$\frac{d\bar{q}_i}{dt} - k_i^{cr}\left(q_i^* - \bar{q}_i\right) = 0 \tag{6.10}$$

Ergun equation:

$$-\frac{\partial p}{\partial z} = \frac{150\mu}{4r_p^2}\frac{(1-\varepsilon_b)^2}{\varepsilon_b^3}u + \frac{1.75\rho_g}{2r_p}\left(\frac{1-\varepsilon_b}{\varepsilon_b^3}\right)u|u| \tag{6.11}$$

Langmuir isotherm:

$$q_i^* = \frac{q_{i,s}^1 b_i^1 c_i R_g T}{1 + \sum_{j=1}^{N_c} b_j^1 c_j R_g T} \tag{6.12}$$

$$b_j^l = b_{j,0}^l \exp\left\{\frac{-\Delta\tilde{H}_j^l}{R_g T}\right\}$$

For the inlet connected to the bed, the flow rate is calculated by:

Table 6.4: Nomenclature

Parameter	Description	Units
A	Surface area	m^2
c_i	Concentration of component i	$\mathrm{mol\,m^{-3}}$
D_L	Effective axial dispersion coefficient	$\mathrm{m^2\,s^{-1}}$
F	Molar flow rate	$\mathrm{mol\,s^{-1}}$
F_{feed}	Feed flow rate	$\mathrm{mol\,s^{-1}}$
h_p	Pellet-to-fluid heat transfer coefficient	$\mathrm{W\,m^{-2}\,K^{-1}}$
h_w	Fluid-to-wall heat transfer coefficient	$\mathrm{W\,m^{-2}\,K^{-1}}$
$\breve{H}$	Enthalpy per unit volume	$\mathrm{J\,m^{-3}}$
$\tilde{H}$	Molar enthalpy	$\mathrm{J\,mol^{-1}}$
J_i	Diffusive flux of comp. i in the fluid phase	$\mathrm{mol\,m^{-2}\,s^{-1}}$
J_T	Thermal diffusive flux in the fluid phase	$\mathrm{J\,m^{-2}\,s^{-1}}$
k_i^{cr}	LDF coefficient of comp. i in the adsorbent crystal	$\mathrm{m\,s^{-1}}$
k_i^p	LDF coefficient of comp. i in the pellet	$\mathrm{m\,s^{-1}}$
L_v	Bed length	m
$n_{feed,i}$	Number of moles of comp. i entering at the feed unit	mol
$n_{vac,i}$	Number of moles of comp. i leaving at the vacuum unit	mol
p_{feed}	Feed pressure	bar
p_{vac}	Vacuum pressure	bar
R_g	Gas constant	$\mathrm{J\,mol^{-1}\,K^{-1}}$
r_b	Bed radius	m
r_p	Pellet radius	m
r_j	Stem position of valve j	-
t_c	Cycle time	s
t_{feed}	Feed/purge time	s
t_{pe}	Pressure equalisation time	s
t_{pr}	Pressurisation/depressurisation time	s
T_f	Fluid temperature	K
T_{feed}	Feed temperature	K
U	Internal energy	J
$\breve{U}$	Internal energy per unit volume	$\mathrm{J\,m^{-3}}$
u	Interstitial velocity	$\mathrm{m\,s^{-1}}$
V	Volume	m^3
W	Work	J
α_{wl}	Mean surface area to volume ratio of the bed wall	$\mathrm{m^{-1}}$
ϵ_b	Bed void fraction	-
ϵ_p	Pellet void fraction	-
λ_L	Effective axial thermal dispersion coefficient	$\mathrm{W\,m^{-1}\,K^{-1}}$
μ	Fluid viscosity	Pa s
ρ	Density	$\mathrm{kg\,m^{-3}}$
Subscript		
b	Bed	
f	Fluid phase	
n	Neighbouring unit	
p	Pellet	
s	Solid phase	
w	Bed wall	
Superscript		
cr	Crystal	
L	Axial	
m	Macropore	
p	Pellet	

$$F = k(p_n - p). \tag{6.13}$$

The pressure drop between the bed and the bed header is neglible if k is assigned a sufficiently large value. The bed header and the tanks are modelled as continuously stirred tanks. The equations are given by:

$$\begin{aligned} F_j &= -F_{n_j}, j = 1, 2, \quad \ldots, N_I, && (6.14)\\ V\frac{dc_i}{dt} &= \sum_{j=1}^{N_I}\left(\frac{F_j + |F_j|}{2}\frac{c_{i,n_j}}{c_{T,n_j}} + \frac{F_j - |F_j|}{2}\frac{c_i}{c_T}\right), && (6.15)\\ \frac{dU_f}{dt} &= \sum_{j=1}^{N_I}\left(\frac{F_j + |F_j|}{2}\tilde{H}_{f,n_j} + \frac{F_j - |F_j|}{2}\tilde{H}_f\right) + h_w A(T_w - T_f), && (6.16) \end{aligned}$$

where N_I is the number of connections. The number of connections for the different tanks is 3, in this case. The regulation of the flow rate exiting the valve is controlled by the following valve equation:

$$F = CV_j c_T\sqrt{\frac{|p_0 - p_{L_b}|}{\rho_f}}. \tag{6.17}$$

Here $CV_j = r_j c_v$, r_j is the stem position, c_v the valve cofficient, p_0 and p_{L_b} are the pressures at the two inlets, respectively, c_T is the total concentration and ρ_f is the fluid density. The pressure in the tank is given by the ideal gas law.

The boundary conditions for the gas phase concentrations and the enthalpy are given by the Danckwerts boundary conditions for flow into the bed and the no diffusive flux for flow out of the bed. With the conventions that the positive flow directions is from 0 to L$_c$ these can be written in a combined form as

$$\begin{aligned} J_T|_{z=0} &= \frac{u + |u|}{2}\left(\check{H}_{f,0^-} - \check{H}_{f,0}\right) && (6.18)\\ J_T|_{z=L_b} &= \frac{u - |u|}{2}\left(\check{H}_{f,L_b^+} - \check{H}_{f,L_b}\right) && (6.19)\\ J_i|_{z=0} &= \frac{u + |u|}{2}\left(c_{i,0^-} - c_{i,0}\right) && (6.20)\\ J_i|_{z=L_b} &= \frac{u - |u|}{2}\left(c_{i,L_b^+} - c_{i,L_b}\right) && (6.21) \end{aligned}$$

where the superscripts $-$ and $+$ indicate the concentration values to the left and right of the boundary, respectively.

A finite volume scheme, using 40 elements along the spatial direction, for the discretisation of the mathematical models, leading to a semi-discrete computational model. The BDF method of 5th order is used for time integration from cycle-to-cycle, using the method of successive substitution.

6.3 SBO setup for different PSA design problem formulations

To demonstration the use of surrogate-based optimisation (SBO) for multi-criteria PSA design we are addressing the following types of optimisation problems:

1. Maximise product purity with constraint on the recovery
2. Maximise purity-recovery, in a Pareto sense
3. Maximise purity-recovery, while minimising the system's power usage, in a Pareto sense

All the design problems can be formulated as in (5.1). The design variables are given in Table 6.5 with the value ranges considered for the optimisation.

Table 6.5: Design variables for the PSA design problem.

Variable	Description	Range	Unit
CV_7	Purge-to-feed	[0,1]c_v	-
t_{feed}	Feed/purge time	[0,200]	s
F_{feed}	Feed flow rate	[5×10^{-4},8×10^{-3}]	mol s^{-1}
p_{vac}	Vacuum pressure	[0.02,0.4]	bar
CV_3, CV_6	Valve parameter for 3, 6	[0.3,1]c_v	-
T_{feed}	Feed temperature	[290,340]	K

This configuration is sufficiently complex to provide a challenging, multi-criteria design example for the SBO procedure. Moreover, a single-criteria problem is also included, in which the product purity is maximised.

The SBO method applied follows the procedure described in Section 4.5, and the tailored strategies given in Chapter 5 are adopted. The choice of surrogate model is the Kriging, which is described in Section 4.4. The SBO settings are described below:

Kriging setup

Section 4.4 describes, in detail, the ordinary Kriging surrogate model technique used in this work. The design problems considered have costly objective and constraint functions since they rely on PSA simulations of the product purity, recovery and the system's power consumption. The SBO procedure assigns a Kriging model for each of these "expensive" functions. The Kriging predictor and the variance estimator are given by Eqns (4.3) and (4.4). The plug-in

Kriging formulas employed are empirical with estimated values for correlation parameters $\boldsymbol{\xi}$ and process variance σ^2. The computationally-stable Kriging approach is adopted, and the parameters $\boldsymbol{\xi}$ in the SE correlation function (4.2) are fitted with MLE using the real-coded GA (population size 64, mutation rate 1/6, mutation index 20, crossover rate 0.9, crossover index 5, for 50 generations), within the search domain $\Xi = [0.3, 15]^6$. In this computationally-stable approach we use several different nugget parameter values, $\hat{\tau}^2 = 10^{-i}$ for $i = 7, 8, \ldots, 14$, and for each we perform the MLE. The nugget value selected is the one with highest MLE. Before the MLE computation, the training data is normalised in both the set of design points, $\boldsymbol{X}^D$, and the corresponding responses, $\boldsymbol{Y}^D$, by scaling the mean, and divide by the standard deviation. The Kriging model is updated (and refitted) for every eight design solution of the PSA simulator added to the training data. Whenever more than 80 design points are in the training data, the Kriging's correlation parameters are no longer refitted,in order to avoid the computationally demanding MLE fitting for large design sets.

The sampling method utilised to generate the training data is the maximin Latin hypercube design (MmLHD), see Section 4.5. The size of the initial training data is set to 16. The size of the training data is large enough to generate a decent coverage, as well as being a good starting point for SBO, based on our experience.

The Kriging models for the product purity and recovery are modified in order to restrict the Kriging predictions to the $[0, 1]$ interval, since these quantities are ratios. This transformed Kriging for the product purity and recovery computations is described in Section 5.5.

Choice of surrogate based criteria and internal optimisation routines

In the inner optimisation of the SBO procedure, the design points are ordered according to the preference ordering of the chosen surrogate based criterion. Here, in contrast to Chapter 4, we are interested in surrogate based criteria suitable for multi-objective optimisation, and for constrained optimisation. Section 5.2 presented some criteria for multi-objective PSA optimisation.

For SBO, we are consistently using the real-coded GA and the NSGA-II (see Section 3.3). Their success is well-documented, and because they make few assumptions about the objective functions, they are suitable for a wide range of surrogate based criteria. Also, the constraint functions can be easily introduced by using penalty terms on the original objective function.

Here follows some SBO methods proposed in this work (defined by the choice of surrogate based criteria and inner optimisation procedure):

Case I: Maximise product purity

- **[SbGA]** *Maximise Kriging predictor, with real-coded GA*
- **[EGO]** *Maximise EGO, with real-coded GA*
- **[SbGA-ALM]** Switch between *maximise Kriging predictor, with real-coded GA* and *maximise ALM, with real-coded GA*

Case II: Maximise product purity with recovery constraint

- **[SbGA-Ch-ALM]** Switch between *Maximise Kriging predictor for purity, with Deb's constraint handling on constraint surrogate for recovery, using real-coded GA*, and *Maximise ALM for purity-recovery, with the Pareto approach, using NSGA-II*

Case III: Maximise product purity-recovery

- **[SbNSGAII]** *Maximise Kriging predictors for purity and recovery, with the Pareto approach, using NSGA-II*
- **[MOEGO]** *Maximise MOEGO for purity and recovery, using real-coded GA*
- **[SbNSGAII-ALM]** Switch between *Maximise Kriging predictors for purity and recovery, with the Pareto approach, using NSGA-II* and *Maximise ALM for purity and recovery, with the Pareto approach, using NSGA-II*

Case IV: Maximise purity-recovery, and minimise power consumption

- **[SbNSGAII-ALM]** Switch between *Maximise Kriging predictors for purity, recovery and power consumption, with the Pareto approach, using NSGA-II* and *Maximise ALM for purity, recovery and power consumption, with the Pareto approach, using NSGA-II*

The abbreviations given inside the brackets ([,]) are henceforth used to refer to the particular SBO method. Here 'Sb' stands for 'Surrogate based', and 'Ch' for 'Constraint handling'. To switch between surrogate based criteria has been shown to work well for a variety of challenging test functions [VK10].

The real-coded GA and the NSGA-II are using simulated binary crossover and polynomial mutation [DG96], with the following parameter settings: population size, 64; crossover rate, 0.9; crossover index, 5; mutation rate, 1/6; mutation index, 20; number of generations, 50.

The constraint surrogates are implemented as described in Section 5.2.3. Recall that the ALM is a strategy for improving the global approximation of the Kriging model (Section 5.3). The ALM criterion selects the design point with largest Kriging variance. The MOEGO is a multi-objective version of the EGO, and is given in Section 5.2.2. The SbNSGAII procedure is described in Section 5.2.1.

Same as in the DP-PSA case study in Chapter 4 we are enforcing diversity by avoiding the selection of “neighbouring” design points, see the diversity constraint (4.20). The diversity parameter is again set to $\varepsilon = 10^{-3}$. The diversity in EGO (and MOEGO) is handled by the Kriging variance estimate, that is, the design points with low Kriging variance is less desired because we believe we know those points better than points of high variance. This is a neat way of representing our “uncertainty” spatially over the design space. These approaches, which are introduced to promote diversity, are not adapted to the case when some design points cannot be evaluated by the PSA simulator - since the search is likely to revisit such points. To avoid this issue, we are incorporating into the diversity handling the location of any design point that failed to be evaluated, see the strategy suggested in Section 5.4.

Many optimisation methods are inherently parallelisable, and an efficient optimisation procedure preferably be able to utilise the CPUs available. All the surrogate based criteria can be evaluated in parallel, and we can achieve multi-point selection, that is, the selection of multiple design points. The procedure needed to use any of the surrogate based criteria above for multi-point selection, see Section 4.6.

The training data for the Kriging model is generated by the MmLHD. LHDs are known to generate a good coverage of the design space, and if the training data is large enough, it can be utilised before initialising the SBO routine for the analysis of the underlying computer simulator over the design space. The results from such an analysis could potentially be used to identify design variables that have little or no impact on the performance indicators, which can be useful in model dimension reduction to ease the optimisation that follows. However, exercise caution before making strong assumptions based on such results.

6.4 Preliminary analysis

The first analysis is the visualisation of the PSA simulator’s response along a hyperline segment between two points in the hyper-rectangle design space $X \subseteq \mathbb{R}^p$, $\boldsymbol{x}^* = \boldsymbol{x}_1 + \lambda(\boldsymbol{x}_2 - \boldsymbol{x}_1)$, where $\lambda = [0, 1]$. Hundred design points were used to represent the hyperline segment (which entails hundred PSA simulation runs). The pathwalk produces a curve for the product purity along the hyperline segment, see Figure 6.2. This is the same procedure used in the DP-PSA study in Chapter 4. The Figure 6.2 shows most of the pathwalks are showing little variability, with the exception of the one line that partially exceeds 60% product purity. This suggests that the larger portion of the design space will lead to low purity, and that some regions are exhibiting nonlinearity. Furthermore, there are some visual gaps in the curves, these gaps are the response values excluded due to simulator failure. This may cause problems for many optimisation methods, especially those relying on gradient information (such as SQP). In Section 5.4 we proposed a modification of the Kriging variance expression that makes the SBO procedure robust against PSA simulator failures. The strategy informs the Kriging variance about the design points where the PSA simulator failed, so that the SBO procedure can avoid revisiting those points, as well as any nearby points in the design space.

The computational model used is the finite volume scheme using 40 volume elements with a Van Leer flux limiter. A single run of the computer simulator ranged between 10 min to an hour depending on the design configuration used. The simulation strategy used is the successive substitution as it is a stable approach to simulate the process behaviour to cyclic steady state.

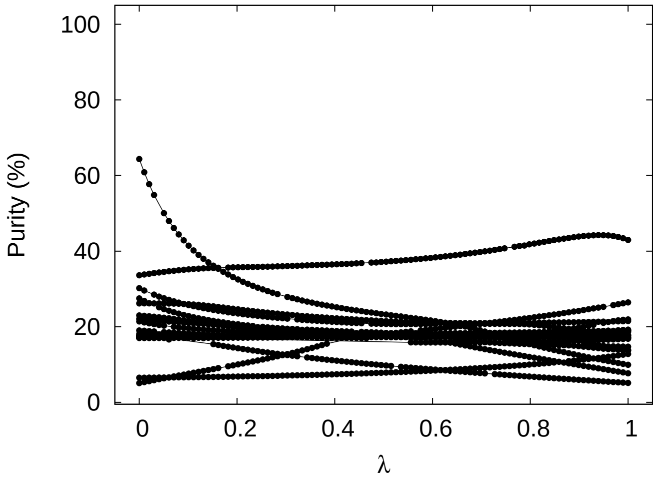


Figure 6.2: Pathwalk along a hyperline segment in the design space for the product purity.

The discrepancy between the Kriging and the PSA simulator response over the design space X can be represented by the normalised root-mean-square error (NRMSE):

$$NRMSE = \frac{\sqrt{\left(\frac{1}{M}\sum_{j=1}^{M}(\hat{y}(\boldsymbol{x}_j) - y(\boldsymbol{x}_j))^2\right)}}{\max_j y(\boldsymbol{x}_j) - \min_j y(\boldsymbol{x}_j)}, \tag{6.22}$$

where $y(\boldsymbol{x})$ is the scalar PSA simulator response, and $\hat{y}(\boldsymbol{x})$ the Kriging predictor response, for $\boldsymbol{x} = (x_1, x_2, \ldots, x_p)^T \in X \subseteq \mathbb{R}^p$. This error has been computed for product purity, recovery, and power consumption, defined by eqns (6.1), (6.2), and (6.3), respectively. The validation data used is a hold-off set consisting of the PSA simulator responses evaluated at $1,000$ design points ($M = 1000$) generated by MmLHD. The Kriging model for all the three objectives are below 10% in relative error in terms of NRMSE, whenever the training data exceeds 60 design solutions. See Figure 6.3. The correlation parameters used are: $\boldsymbol{\xi}_{purity} = (1.153, 4.693, 1.615, 15.0, 14.393, 15.0)^T$, $\boldsymbol{\xi}_{recovery} = (3.235, 1.834, 1.708, 5.627, 8.467, 15.0)^T$, $\boldsymbol{\xi}_{power} = (5.463, 7.53, 0.987, 2.721, 15.0, 15.0)^T$. This actually expected, according to the analysis presented in [LSW09], the error convergence rapidly decreases when the training data to the Kriging is about 11 times larger than the dimension of the design space, that is, $11p$.

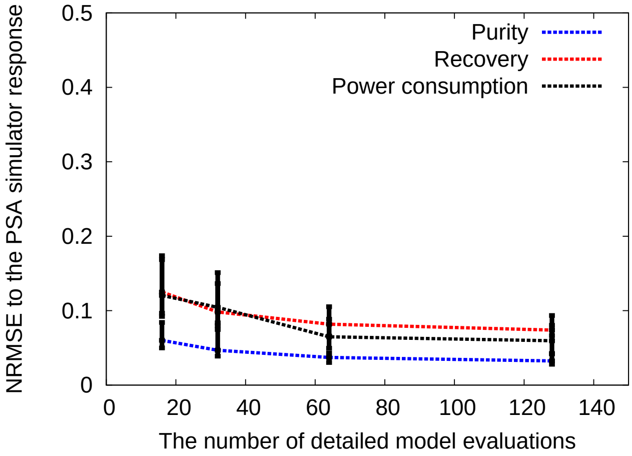


Figure 6.3: NRMSE for the Kriging model to training data for product purity, recovery, and power consumption calculated using the detailed PSA simulator. The error bars represent the variability over ten sets of training data (min-mean-max).

This analysis is usually not computationally viable in practice as it requires many new PSA simulation runs. An estimate of the NRMSE can be obtained by using cross-validation, but from our experience that approach will severely underestimate the error estimate when dealing with the PSA model outputs. We performed this analysis to provide confidence in the use of Kriging in SBO for PSA optimisation. In the next section an efficient global sensitivity analysis is proposed for the PSA design formulation, which only requires the initial training data to be of moderate size.

6.5 Global sensitivity analysis on a small initial training data

In this section we are using the sensitivity measure proposed by Sobol' [Sob93] to understand the effect of the variability in the design configurations to the PSA simulator outputs (purity, recovery, and power consumption per mole).

Global sensitivity analysis (GSA) is concerned with quantifying how the variation in the simulator's output depends on different sources of variation over the entire design space, here treated as random input data, by providing quantitative importance measures that relate the variance of the output with each design variable considered. This form of analysis of model sensitivity has been already applied to parts of the CCS chain, to the geological storage of CO2 by Kovscek and Wang [KW05] where the effect of porosity and permeability on reservoir performance was assessed, as well as for understanding the impact of the impurities in CO_2 transportation on the outflow following pipeline failure [BBMF13].

Global sensitivity analysis with the Sobol' method [Sob93] applied on the Kriging model can be an efficient means of quantifying the impact that each design variable has on the PSA simulator outputs that are considered in the optimisation study to be performed. The Sobol method is related to analysis of variance (ANOVA) and decomposes the model variation into a number of effects that represent the influence of each design variable, represented by a probability distribution, and their interactions. Many different methods have been proposed to compute the integrals required to calculate these effects, of these the most widely applied are Monte Carlo sampling [Sob01] and the Extended Fourier Amplitude Sensitivity Test (EFAST) [MTS82]. This approach typically requires large sample sizes to provide accurate estimations of the sensitivities, making them impractical when the underlying model is computationally expensive. Sobol' based GSA has for this reason been considered impractical, but thanks to some recent advances [OO04, Sud08, MILR09], which are utilizing surrogate models for reducing the expense of the Sobol' method, the Sobol' method has found practical use for a wide range of modelling applications, for example modelling of landslides [RF11], groundwater flow [FGI$^+$13], and CO_2 pipeline failure [BBMF13].

Sudret [Sud08] proposed a method for computing Sobol sensitivities by approximating the model's output by a polynomial expansion, known as generalised polynomial chaos (gPC) [GS91]. The gPC expansion is a linear combination of suitable global polynomial approximations in probability space, for which the statistical moments, expected value and variance, are known exactly from the coefficients of the expansion. The family of orthonormal mono-dimensional polynomials is selected in accordance with the general Askey scheme with respect to the probability measure of each random input variable [XK02].

The stochastic collocation method [BNTT11] has been applied to build a gPC expansion on tensor grids, suitable for high dimensional random input data, to mitigate the so-called “Curse of Dimensionality.” This method constructs a gPC expansion, often interpolatory, on input points, collectively known as “sparse grid” [BG04]. This “sparse” gPC expansion has been shown to use fewer model evaluations than the other methods identified above (see comparison in [BBMF13]), which means that the use of GSA for complex numerical models such as the computational PSA model is possible. Here follows an account of the Sobol’ method, and the computational strategy used.

6.5.1 Global sensitivity analysis with Sobol’ indices

Let $\boldsymbol{Z}(\omega) = (Z_1(\omega), Z_2(\omega), \dots, Z_n(\omega), \dots, Z_N(\omega)) : \Omega \to \mathbb{R}^N$ represent N independent and identically distributed (i.i.d.) random variables, $\Gamma_n \subseteq \mathbb{R}$ the image set of the random variable Z_n, and $\Gamma = \prod_{n=1}^{N} \Gamma_n$. Hence the joint probability distribution function $\rho : \Gamma \to \mathbb{R}$ of $\boldsymbol{Z}$ can be factorised as $\rho(\boldsymbol{Z}) = \Pi_{n=1}^{N} \rho(Z_n)$, where $\rho(Z_n)$ is the marginal probability distribution function of Z_n. Let $(\Gamma, \mathrm{B}(\Gamma), \rho(\boldsymbol{Z})\mathrm{d}\boldsymbol{Z})$, where $\mathrm{B}(\Gamma)$ is the Borel σ-algebra on Γ, and $\rho(\boldsymbol{Z})\mathrm{d}\boldsymbol{Z}$ is the probability distribution measure of $\boldsymbol{Z}$ on Γ. $L^2_\rho(\Gamma)$ denotes the Hilbert space consisting of square integrable functions on Γ with respect to the measure $\rho(\boldsymbol{Z})\mathrm{d}\boldsymbol{Z}$.

A function $y \in L^2_\rho(\Gamma)$ can be expanded as an ANOVA decomposition

$$y(\boldsymbol{Z}) = y_{\mathbf{0}} + \sum_{\mathbf{j} \subseteq \mathcal{J}} y_{\mathbf{j}}(\boldsymbol{Z}_{\mathbf{j}}), \tag{6.23}$$

for which $\boldsymbol{Z}_{\mathbf{j}} = (Z_{j_1}, Z_{j_2}, \dots, Z_{j_{|\mathbf{j}|}})$ is a vector including the components of $\boldsymbol{Z}$ indexed by $\mathbf{j}$, where $\mathbf{j}$ represents a non-empty subset of the coordinate indices $\mathcal{J} = \{1, \dots, N\}$ with cardinality denoted by $|\mathbf{j}|$. For example, for $\mathbf{j} = \{2, 3\}$ and $\{1, 3, 4\}$, $|\mathbf{j}| = 2$ and 3, respectively. Let $\Gamma^{\mathbf{j}}$ denote the $|\mathbf{j}|$-dimensional hyper-rectangle defined as the projection of the N-dimensional Γ onto the hyper-rectangle indexed by $\mathbf{j}$. The ANOVA representation allows one to distinguish between first order effects, low-order interdependence, and high-order interaction. The summands $y_{\mathbf{j}}(\boldsymbol{Z}_{\mathbf{j}})$ can be calculated recursively as follows:

$$y_{\mathbf{0}} = \int_{\Gamma^N} y(\boldsymbol{Z}) \rho(\boldsymbol{Z}) \mathrm{d}\boldsymbol{Z}, \tag{6.24}$$

and

$$y_{\mathbf{j}}(\boldsymbol{Z}_{\mathbf{j}}) = \int_{\Gamma^{N-|\mathbf{j}|}} y(\boldsymbol{Z})\rho(\boldsymbol{Z}_{\mathcal{J}\setminus\mathbf{j}})\mathrm{d}\boldsymbol{Z}_{\mathcal{J}\setminus\mathbf{j}} - \sum_{\mathbf{k}\subset\mathbf{j}} y_{\mathbf{k}}(\boldsymbol{Z}_{\mathbf{k}}) - y_{\mathbf{0}}. \tag{6.25}$$

The measure $\rho(\boldsymbol{Z}_{\mathcal{J}\setminus\mathbf{j}})\mathrm{d}\boldsymbol{Z}_{\mathcal{J}\setminus\mathbf{j}}$ represents the integration over $\Gamma^{\mathcal{J}\setminus\mathbf{j}}$. The ANOVA expansion is an exact projection of $y(\cdot)$ with respect to the $L^2_\rho(\Gamma)$-inner product onto the mutually orthogonal $y_{\mathbf{j}}(\cdot), \mathbf{j} \subseteq \mathcal{J}$, that is,

$$\int_\Gamma y_{\mathbf{j}}(\boldsymbol{Z}_{\mathbf{j}})y_{\mathbf{k}}(\boldsymbol{Z}_{\mathbf{k}})\rho(\boldsymbol{Z})\mathrm{d}\boldsymbol{Z} = \delta_{\mathbf{jk}}, \tag{6.26}$$

in which $\delta_{\mathbf{jk}} \neq 0$ if $\mathbf{j} = \mathbf{k}$, else $\delta_{\mathbf{jk}} = 0$. Hence it holds that for all $|\mathbf{j}| > 0$,

$$\int_\Gamma y_{\mathbf{j}}(\boldsymbol{Z}_{\mathbf{j}})\rho(\boldsymbol{Z})\mathrm{d}\boldsymbol{Z} = 0. \tag{6.27}$$

6.5.2 Use of gPC for computing Sobol' indices

Sobol' [Sob93] proposed a variance based GSA method that extends decomposition (6.23) to a variance based representation where the summands can be interpreted as relative importance measures of the subsets of the input variables. Following Eqs. (6.23), (6.26-6.27) the Sobol' indices are given by:

$$S_{\mathbf{j}} = \frac{\mathbb{V}[y_{\mathbf{j}}(\boldsymbol{Z}_{\mathbf{j}})]}{\mathbb{V}[y(\boldsymbol{Z})]} = \frac{\int_\Gamma y^2_{\mathbf{j}}(\boldsymbol{Z}_{\mathbf{j}})\rho(\boldsymbol{Z})\mathrm{d}\boldsymbol{Z}}{\int_\Gamma y^2(\boldsymbol{Z})\rho(\boldsymbol{Z})\mathrm{d}\boldsymbol{Z} - y^2_{\mathbf{0}}}, \tag{6.28}$$

in which the variance of $y(\boldsymbol{Z})$ under the probability measure $\rho(\boldsymbol{Z})\mathrm{d}\boldsymbol{Z}$ is

$$\mathbb{V}[y(\boldsymbol{Z})] = \sum_{\mathbf{j}\subseteq\mathcal{J}} \mathbb{V}[y_{\mathbf{j}}(\boldsymbol{Z}_{\mathbf{j}})] = \sum_{\mathbf{j}\subseteq\mathcal{J}} \int_\Gamma y^2_{\mathbf{j}}(\boldsymbol{Z}_{\mathbf{j}})\rho(\boldsymbol{Z})\mathrm{d}\boldsymbol{Z}. \tag{6.29}$$

Accordingly, it holds that $\sum_{\mathbf{j}\subseteq\mathcal{J}} S_{\mathbf{j}} = 1$. It must be noted that Eq. (6.29) relies on the assumption of the mutual independence of $\{Z_n\}$. The Sobol' indices quantify the relative importance of their corresponding effects which provide valuable insight on the mixed effects. The total effect induced by each input variable Z_n can be defined by

$$S^T_n = \sum_{\mathbf{j}\subseteq\mathcal{J}:n\in\mathbf{j}} S_{\mathbf{j}}. \tag{6.30}$$

The total effects S^T_n are in practice easy to compute through their complement in $\mathcal{J}$, that is, $S_{\mathbf{j}}$ for which $n \notin \mathbf{j}$.

The use of global polynomial approximations is promising when the quantity of interest

$y(\boldsymbol{Z})$ is smooth with respect to the random input variables $\{Z_n\}$.

The tensor product structure of $L^2_\rho(\Gamma)$ allows one to introduce a polynomial subspace of $L^2_\rho(\Gamma)$ denoted by $\mathbb{P}(\Gamma)$ as well as $\rho(\boldsymbol{Z})\mathrm{d}\boldsymbol{Z}$-orthonormal basis

$$\Psi_{\mathbf{p}}(\boldsymbol{Z}) = \prod_{n=1}^{N} \Psi_{p_n}(y_n), \quad \mathbf{p} = (p_1, p_2, \dots, p_n, \dots, p_N) \in \mathbb{N}^N, \tag{6.31}$$

where $\Psi_{p_n}(y_n)$ denotes $\rho(y_n)\mathrm{d}y_n$-orthonormal polynomials on Γ_n. The goal is to project $y(\boldsymbol{Z})$ on $\mathbb{P}_{\Lambda(\omega)}(\Gamma) = span\{\Psi_{\mathbf{p}}(\boldsymbol{Z}),\ \mathbf{p} \in \Lambda(\omega)\}$ to obtain a global polynomial approximation

$$y_\omega(\boldsymbol{Z}) = \sum_{\mathbf{p}\in\Lambda(w)} \alpha_{\mathbf{p}} \Psi_{\mathbf{p}}(\boldsymbol{Z}), \tag{6.32}$$

for a suitable $\{\Psi_{\mathbf{p}}\}_{\mathbf{p}\subseteq\Lambda(\omega)}$, where $\Lambda(\omega), \omega \in \mathbb{N}$ are polynomial spaces of increasing index sets with respect to ω. This representation is known as the gPC expansion [GS91].

In a computational context the gPC expansion needs to be truncated, and here the general construction of $\Lambda(\omega)$ provides a polynomial space hierarchy with ω. The classical Tensor Product polynomial space $\Lambda(\omega) = span\{\mathbf{p} \in \mathbb{N}^N : \max_{n=1,2,\dots,N} p_n \leq \omega\}$ suffers from the curse of dimensionality since its dimension increases exponentially fast with the number N of random input variables. A more attractive option is the *sparse* Total Degree polynomial space $\Lambda(\omega) = span\{\mathbf{p} \in \mathbb{N}^N : \sum_{n=1}^{N} p_n \leq \omega\}$. The choice and construction of $\Lambda(\omega)$ are discussed in detail in [BNTT11].

When the randomness in $\boldsymbol{Z}$ is described by any of the common probability distributions, the Sobol' indices can be determined *exactly* from the coefficients of the gPC expansion of $y(\boldsymbol{Z})$. This is true when described by for example the uniform and normal distribution. When computing statistical moments such as the expected value and variance of $y(\boldsymbol{Z})$ the appropriate family of $\rho(\boldsymbol{Z})\mathrm{d}\boldsymbol{Z}$-orthonormal polynomials should be chosen with respect to the distribution measure of the input variables [XK02]. As the random variables in this study are uniformly distributed the normalised Legendre polynomials are employed. The first and second statistical moments of $y_\omega(\boldsymbol{Z})$ are then directly obtained from the gPC coefficients $\{\alpha_{\mathbf{p}}\}$:

$$E\left(y_\omega(\boldsymbol{Z})\right) = \alpha_{\mathbf{0}}, \text{ and } \mathbb{V}[y_\omega(\boldsymbol{Z})] = \sum_{\mathbf{p}\in\Lambda(\omega)} \alpha^2_{\mathbf{p}} - \alpha^2_{\mathbf{0}}. \tag{6.33}$$

There is a one-to-one correspondence between the analytical Sobol' indices and distinct subsets of gPC coefficients [Sud08]: the gPC expansion of $y(\boldsymbol{Z})$ may be recast as

$$\lim_{\omega\to\infty} y_\omega(\boldsymbol{Z}) = \alpha_{\mathbf{0}} + \sum_{\mathbf{j}\subseteq\mathcal{J}} \sum_{\substack{\mathbf{p}\in\mathbb{N}^N:\\ i_n>0\Leftrightarrow n\in\mathbf{j}}} \alpha_{\mathbf{p}}\Psi_{\mathbf{p}}(\boldsymbol{Z}), \tag{6.34}$$

and then given (6.23) and (6.34) the one-to-one correspondence is explicit, that is,

$$y_{\mathbf{j}}(\boldsymbol{Z}_{\mathbf{j}}) = \sum_{\substack{\mathbf{p}\in\mathbb{N}^N:\\ p_n>0\Leftrightarrow n\in\mathbf{j}}} \alpha_{\mathbf{p}}\Psi_{\mathbf{p}}(\boldsymbol{Z}_{\mathbf{j}}). \tag{6.35}$$

Insert Eq. (6.35) into the Sobol' index definition, (6.28), and exploit orthonormality of $\Psi_{\mathbf{p}}$:

$$S_{\mathbf{j}} = \frac{\sum_{\mathbf{p}\in\mathbb{N}^N:p_n>0\Leftrightarrow n\in\mathbf{j}} \alpha_{\mathbf{p}}^2}{\sum_{\mathbf{p}\in\mathbb{N}^N} \alpha_{\mathbf{p}}^2 - \alpha_{\mathbf{0}}^2}. \tag{6.36}$$

The gPC coefficients $\{\alpha_{\mathbf{p}}\}$ can in some cases be computed using Galerkin projection that involves a reformulation of the model equations [LMK10], but this is impractical for deterministic models of complex structure, such as those involving non-linear governing equations. To circumvent this, the non-intrusive stochastic sparse grid collocation method [BNT10] can be applied. Other non-intrusive approaches are described elsewhere [LMK10], for example those based on projection, that determine the coefficients by integration, and regression-based approaches, that rely on least squares.

Stochastic sparse grid collocation methods build upon a set of collocation points $\{\boldsymbol{Z}_k \in \Gamma\}$ with corresponding function responses $\{y(\boldsymbol{Z}_k)\}$, a global polynomial approximation $y_\omega : C^0(\Gamma) \to \mathbb{P}_{m(\mathbf{i})-1}(\Gamma)$:

$$y_{SG,\omega}(\boldsymbol{Z}) = \sum_{\mathbf{i}\in\mathbf{I}(\omega)} c(\mathbf{i}) \bigotimes_{n=1}^{N} U_n^{m(i_n)}[y](\boldsymbol{Z}),\ c(\mathbf{i}) = \sum_{\substack{\mathbf{j}=\{0,1\}^N:\\ \mathbf{i}+\mathbf{j}\in\mathbf{I}(\omega)}} (-1)^{|\mathbf{j}|}, \tag{6.37}$$

where $U_n^{m(i)} : C^0(\Gamma_n) \to \mathbb{P}_{m(i)-1}(\Gamma_n)$ denotes a mono-dimensional Lagrangian polynomial interpolant operator, $\bigotimes_{n=1}^N$ the Cartesian tensor product operator of the sets of collocation points in each direction n, $\mathbf{i} \in \mathbb{N}_+^N$ multi-indices, $\mathbf{I}(\omega)$ a sequence of increasing index sets and $m(i)$ the number of collocation points used to build the mono-dimensional interpolant at level i. The polynomial approximation (6.37) is known as sparse grid approximation and its construction is described elsewhere [BNTT11].

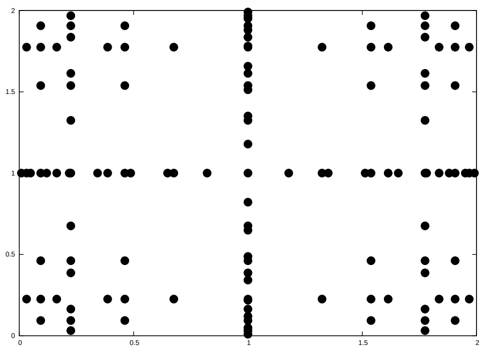


Figure 6.4: 2D example of a Smolyak sparse grid $\omega = 4$ using Gauss-Legendre points

The set of indices $\mathbf{I}(\omega)$ can be chosen so that the approximation belongs to a given polynomial space $\Lambda(\omega)$ [BNTT11]. This study will use the isotropic Smolyak sparse grid which is defined by:

$$\mathbf{I}_{\text{Smolyak}}(\omega) = \left\{ \mathbf{i} \in \mathbb{N}_+^N : \sum_{n=1}^{N} (i_n - 1) \le \omega \right\}, \tag{6.38}$$

and

$$m(i) = \begin{cases} 2^{i-1} + 1, & i > 1 \\ 1, & i = 1. \end{cases} \tag{6.39}$$

When using the Smolyak sparse grid many of the coefficients $c(\mathbf{i})$ in (6.37) may be zero, hence the name “sparse grid.” In this study the mono-dimensional Lagrangian interpolants use the non-nested Gauss-Legendre rule so that the gPC expansion is built upon tensor products of Legendre polynomials. The collocation points used in a Smolyak sparse grid with the Gauss-Legendre rule are shown in Figure 6.4. Keep in mind that generally the sparse grid approximation (6.37) is not interpolatory [BNR00].

Using direct sparse grid quadrature requires the evaluation of high-dimensional integrals to obtain the gPC coefficients, to circumvent this the key is to convert the sparse grid approximation into a Legendre gPC expansion without the need to evaluate any high-dimensional integrals [Tam12], and insert the coefficients in Eq. (6.36) to obtain the Sobol’ indices.

For a comparison of the performance using sparse grid based gPC and some other traditional methods (Monte Carlo, Quasi-Monte Carlo, EFAST), we will use the Ishigami test function is given in [SCS00]:

$$y(\boldsymbol{x}) = \sin(x_1) + 7\sin^2(x_2) + 0.1x_3^4 \sin(x_1) \tag{6.40}$$

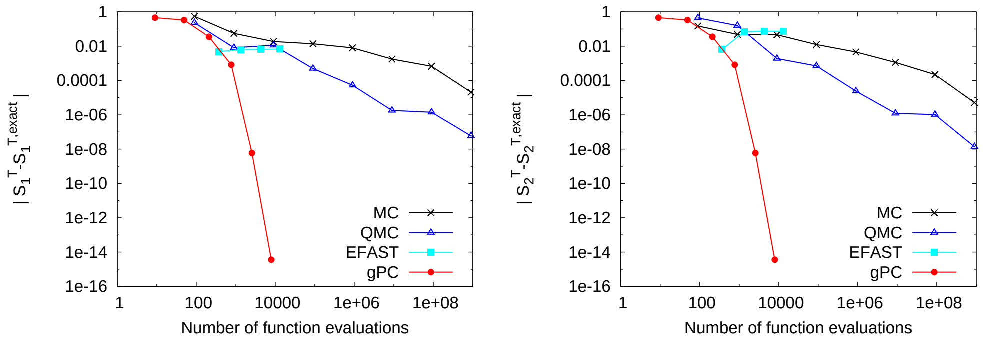


Figure 6.5: Error convergence for the total effects for x_1 and x_2 on the Ishigami test function, a typical benchmark in global sensitivity analysis. Left: S_1^T and Right: S_2^T.

where $x_n \in [-\pi, \pi]$. A fourth dummy input which is not used in the function evaluation is also used in the analysis. We only present the total effects for variable 1 and 2, see Figure 6.5. The Quasi-Monte Carlo (QMC) used the Sobol' sequence. This shows the great performance obtained using sparse grid based gPC.

6.5.3 gPC on Kriging using a small initial training data: GSA on PSA

Up to this point we have presented the construction of a "sparse" gPC expansion for representing the computational model's output $y(\boldsymbol{Z})$ over the design space $\Omega \subseteq \mathbb{R}^N$, given by our choice of design variables represented by $\boldsymbol{Z} = (Z_1, Z_2, \ldots, Z_N)^T \in \mathbb{R}^N$. For us, the sensitivity analysis should be applied on the PSA simulator's scalar output (e.g. product purity) considered over the design space. However, because the PSA simulator fails for some design points, and that the sparse-grid based gPC is built on a fixed sparse grid, this gPC construction is not possible if some design points of the sparse grid cannot be evaluated. To overcome this, we propose the use of the Kriging model on the original PSA simulator output, using some initial training data $D = \{(\boldsymbol{X}^D, \boldsymbol{Y}^D)\}$, and then apply the sparse grid gPC on this Kriging model. The Kriging predictor $\hat{y}(\cdot)$, given by Eq. (4.3), is replacing $y(\cdot)$ in Eq. (6.28) for calculating the Sobol' index for $\mathbf{j}$, that is,

$$S_{\mathbf{j}} = \frac{\int_\Gamma \hat{y}_{\mathbf{j}}^2(\boldsymbol{Z}_{\mathbf{j}})\rho(\boldsymbol{Z})\mathrm{d}\boldsymbol{Z}}{\int_\Gamma \hat{y}^2(\boldsymbol{Z})\rho(\boldsymbol{Z})\mathrm{d}\boldsymbol{Z} - \hat{y}_{\mathbf{0}}^2}. \tag{6.41}$$

Because the Kriging model is smooth, the gPC method is a good fit even for relatively low resolution, specified by ω. The gPC may only need a few thousand Kriging evaluations, which are fast-to-evaluate, to provide accurate estimates of the Sobol' indices (see Table 6.6), as opposed to Monte Carlo computations that most likely require at least 10^6 evaluations.

Global sensitivity analysis with gPC expansion on sparse grids is here applied on the Krig-

Table 6.6: The computation of the total effects of the design variables, see Eq. (6.30), on the Kriging model, with gPC expansion on sparse grids. The Kriging model is applied for the product purity, and used a training data set of size 128.

ω	**Kriging evaluations**	CV$_7$	F_{feed}	t_{feed}	p_{vac}	$CV_{3/6}$	T_{feed}
0	13	0.7316	0.0945	0.1648	0.0080	0.0008	0.0001
1	97	0.8172	0.1102	0.1494	0.0188	0.0098	0.0027
2	545	0.7737	0.0994	0.2003	0.0154	0.0139	0.0111
3	2561	0.7818	0.0921	0.1908	0.0131	0.0119	0.0077
4	10625	0.7811	0.0915	0.1897	0.0131	0.0121	0.0068

ing model with the same settings used in the previous section, where the correlation parameters are estimated with the Maximum Likelihood Estimation (MLE) method. The Smolyak sparse grid is used with level $\omega = 4$. See Figure 6.6 where the total effects S_n^T are shown for the product purity and recovery, calculated from the Sobol' indices using equation (6.30) for all the design variables, given as averages with min-to-max ranges.

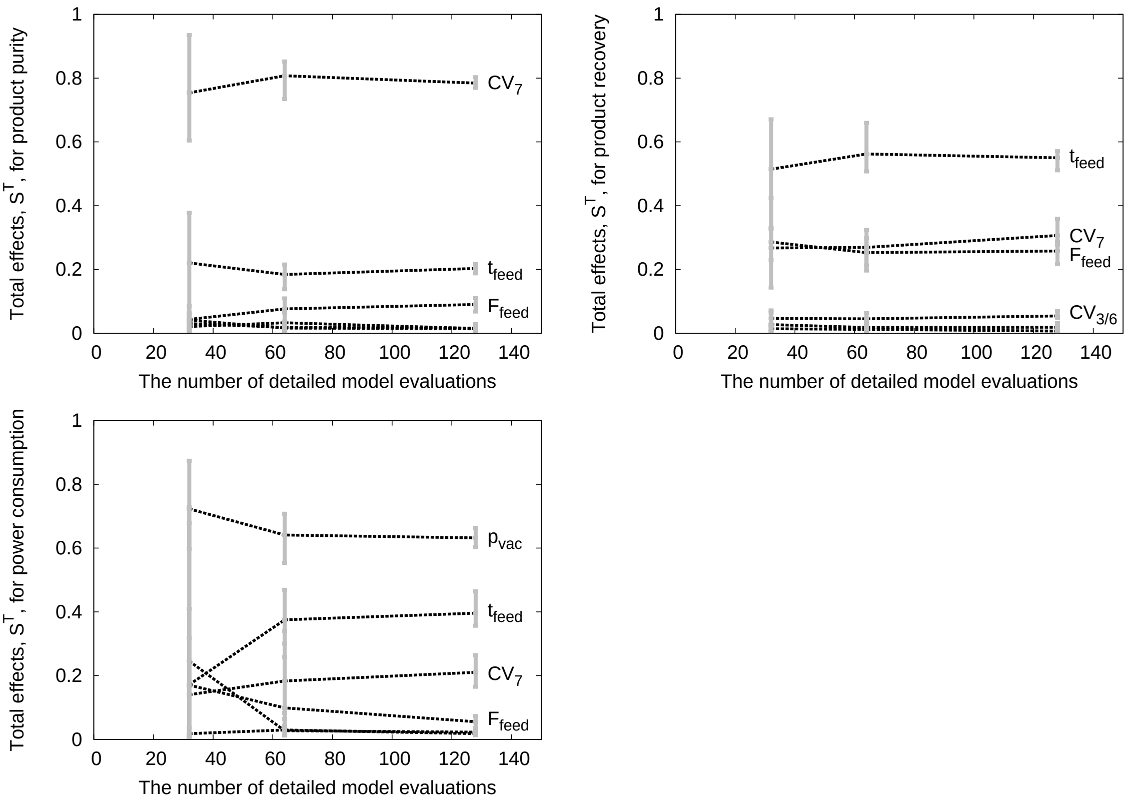


Figure 6.6: Total effects calculated with Sobol' indices for each of the six design variables (given as averages over 5 different training data of sizes 32, 64, and 128) on the quantities of interest via the Kriging model.

This is to our knowledge the first attempt to use a gPC expansion on top of the Kriging model that has been built for the original model's output. An interesting observation is that the Kriging model is adapted to the initial training data through the MLE-fit of the correlation parameters $\boldsymbol{\xi} = (\xi_1, \xi_2, \ldots, \xi_N)^T$, and the parameter choices are expected to greatly influence the estimation of the sensitivity measures. For example for the product purity we obtained $\boldsymbol{\xi} = (1.153, 4.693, 1.615, 15.0, 14.393, 15.000)^T$ for the product purity, which means the CV_7 is the most influencial, followed by t_{feed} and F_{feed}, the same order of importance identified by the total effect in Figure 6.6. On the other hand, for the product recovery, the ordering of the total effects are not the same, for example F_{feed} and CV_7 are switch places in the ordering.

That t_{feed} and F_{feed} are influencial across all the objectives is expected. Also, the vacuum pump is known to be the unit that consumes most power. We therefore expect that the power consumption would depend strongly on p_{vac} (Figure 6.6). It is the design variable affecting the energy performance of the system the most.

To understand the impact the correlation parameters on the sensitivity measure calculated on the Kriging model, the correlation parameters are set to the same value. Then, when the effect of using fitted correlation parameters is "removed" the effect of each design variable is still similar, see Figure 6.7. This reveals to us that the parameter estimation for the Kriging model is not dominating the sensitivity measures too much. However, we have noticed that with too short correlation lengths the sensitivity measures can be become too influenced by the non-linearity of the Kriging's response surface.

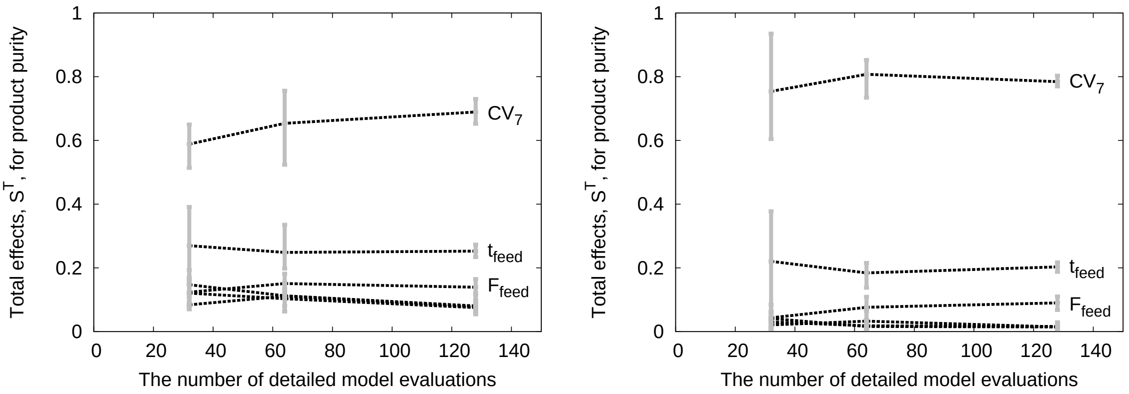


Figure 6.7: Total effects calculated with Sobol' indices (given as averages over 5 different training data of sizes 32, 64, and 128) via the Kriging model for the product purity, with all the correlation parameters ξ_j set to 2 (Left). This reveals that still these total sensitivity measures under isotropic correlation structure (Left) are similar to those obtained in the anisotropic case where the correlation parameters $\boldsymbol{\xi}$ were the MLE esimates (Right).

From these results we can conclude that CV$_7$, t_{feed}, and F_{feed} all have a strong influence on the product purity and recovery. The remaining ones seem to have little effect. For the power consumption, the p_{vac} is a very important design variable, and again t_{feed} and CV$_7$ are considered to have significant impact. This analysis using only a small initial training (of size 64) can be useful for model dimension reduction, for instance. In this case, perhaps $CV_{3/6}$ and T_{feed} could have been removed as design variables in the PSA design problem, and rather be chosen judiciously with some other approach, such as relying on expert knowledge.

6.6 SBO performance

In this section we are presenting the results obtained from performing SBO for a variety of optimisation problem formulations. The SBO performance is investigated for a few different surrogate based criteria, and the resulting design solutions are analysed at the end.

First a single-objective problem is considered, for which the product purity is maximised. Then the same optimisation problem is revisited with the exception that the design solutions of interest must exceed 90 $\%$ product recovery. After that, the product purity and recovery is optimised simultaneously in accordance with Pareto optimality. Finally, the system's power consumption is included as a third objective, which preferably should be as low as possible.

The SBO methods are compared to the real-coded GA or the NSGA-II, which both are using the following internal settings: population size 16, crossover rate 0.9, crossover index 5, mutation rate 1/6, mutation index 20, and the number of generations depend on the case example. The crossover and mutation operators used are the simulated binary crossover and polynomial mutation.

Let us assume we only have eight CPUs to our disposal at any given time. For all results presented, the SBO methods have utilised all the available CPUs for selecting eight new design solutions per SBO iteration. For the SBO methods that switch between two surrogate based criteria, e.g. SbGA-ALM and SbNSGAII-ALM, we are dividing the work (and the CPUs) needed for the multi-point selection equally between these criteria (that is, four points are selected by maximising the Kriging predictor, and four points selected by ALM).

6.6.1 Case I: Maximise CO$_2$ purity

To maximise the CO$_2$ purity, we applied the proposed SBO methods, SbGA, SbGA-ALM, and EGO, and compared the results to the real-coded GA. The comparison is presented in Figure 6.8, where five replications of all the methods were performed, using five different initial training data (of size 16). The real-coded GAs are initialised with the same training data. The variability in the results are caused by the randomness in the GA operators, as well as the generated initial training data, as they are the only sources of randomness.

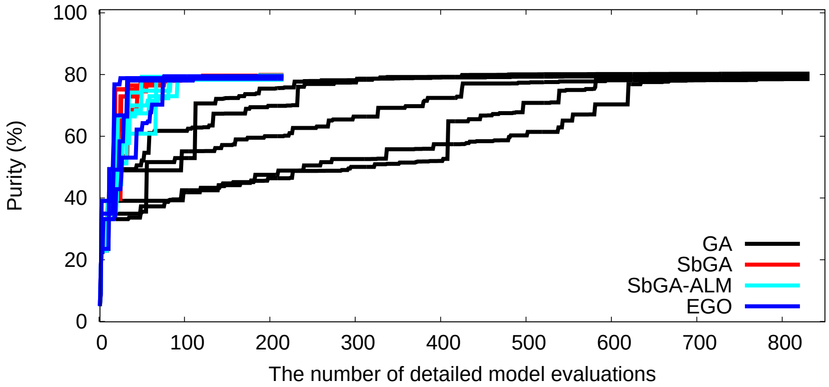


Figure 6.8: The use of the real-coded GA on CO_2 purity in the product, with and without the use of a Kriging model.

What can be noted from Figure 6.8 is that all the SBO methods outperform the stand-alone GA, and seem to converge to good design solutions within a budget of hundred twenty design solutions. The SbGA and EGO are displaying the best convergence rate, although one replication of the EGO method had an outlier with slighly worse efficiency. For this optimisation problem the ALM approach seems to be too conservative, and thus less efficient. Nonetheless, the performance with ALM is comparable to the others, which makes it advantageous as it also is the most robust of the three.

6.6.2 Case II: Maximise CO_2 purity with $> 90\%$ CO_2 recovery

This optimisation example is the same as for Case I but with an added design constraint, specifying that the product recovery needs to exceed 90%. As discussed in the introduction, for carbon capture at least a 90% CO_2 recovery is required. To enforce this constraint, the constraint handling method for GAs, develop by Deb [Deb00], is adopted. The use of Deb's constraint handling in SBO is described in Section 5.2.3. The performance of the constrained SBO method, SbGA-Ch-ALM, is again substantionally better than the conventional real-coded GA-Ch.

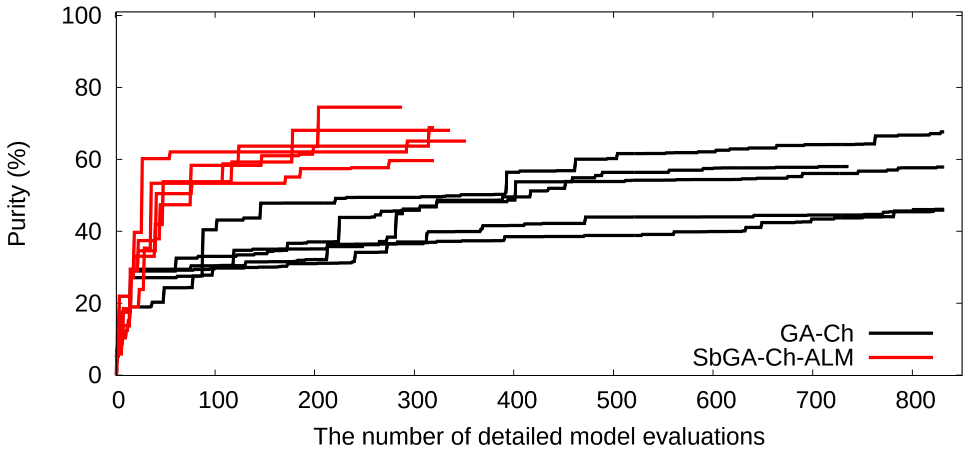


Figure 6.9: Comparison of real-coded GA-Ch and the SbGA-Ch-ALM, both using Deb's constraint handling.

6.6.3 Case III: Maximise CO_2 purity and recovery

In contrast to the previous case example, the goal is to maximise the CO_2 purity and recovery at the same time. For this, we are optimising the two objectives in a Pareto sense. Multi-objective optimisation approaches are presented in Section 3.2.3. The SbNSGAII and SbNSGAII-ALM up to 336 design solutions, and compared to the conventional NSGAII, for which we run up to 1600 design solutions. Figure 6.10 reveals the rapid convergence of the SBO methods, when compared to the NSGAII. See also Figure 6.13 for the Pareto front approximations obtained with the SBO methods and the NSGA-II, after 336 and 1600 PSA simulator evaluations, respectively. The original NSGA-II is performing very well for a large number of PSA simulator runs, in terms of richness and coverage of the Pareto front. Because of the huge number of simulation runs performed with various optimisation strategies, we are confident that the NSGA-II Pareto front, shown in Figure 6.13, is a good approximation of the true Pareto front.

The MOEGO has been found to be less robust than SbNSGA-ALM, see Figure 6.11. The MOEGO transforms the multi-objective problem into a single-objective one, in a statistically pleasing way, but also makes it heavily reliant on the Kriging being an accurate statistical representation (both for the predictive mean and variance), which is a rather strong assumption. As an example, consider the problem of minimising $\boldsymbol{y}(\boldsymbol{x}) = (y_1(\boldsymbol{x}), y_2(\boldsymbol{x}))^T = \left(\sqrt{(x_1-1)^2+(x_2-1)^2}, \sqrt{(x_1+1)^2+(x_2+1)^2}\right)^T$ over support $\boldsymbol{x} = (x_1, x_2)^T \in [-2,2]^2$, which was the numerical example used in [EDK11], the original work on MOEGO. By changing the correlation parameter choice between from the original setting $\boldsymbol{\xi} = (70.7, 70.7)^T$ to $\boldsymbol{\xi} = (1,1)^T$, the performance is deteriorating, see Figure 6.12.

Based on the results presented, SbNSGA-ALM has the best overall performance as a multi-objective optimisation method. The robustness of both the SbNSGAII and SbNSGAII-ALM is encouraging. For the Pareto front segments, for which both SbNSGAII and SbNSGAII-ALM were represented, the SbNSGAII most often displayed the best precision (see Figure 6.13). On the other hand, SbNSGAII-ALM seems to attain a more diverse set of Pareto solutions than SbNSGAII. This can be seen from that the solutions generated by SbNSGA-ALM (in Figure 6.10) amount to a notably larger portion of the Pareto front, compared to SbNSGAII (without ALM).

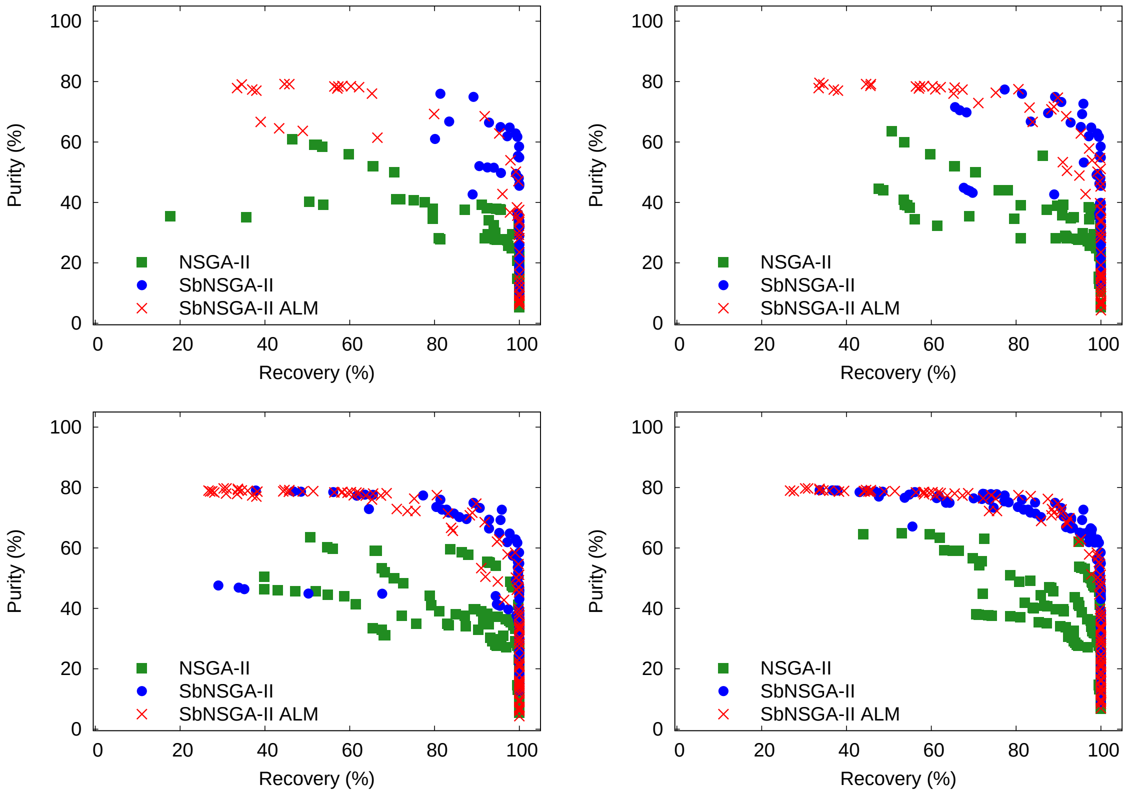


Figure 6.10: Comparison of different strategies for solving the CO_2 purity vs recovery problem: Top left: 64 points, right: 96. Bottom left: 176, right: 256. The results for each strategy is the collection of their Pareto fronts from five different initial training data.

The system's power consumption is also, as discussed earlier, a very important performance indicator, for assessing PSA for CO_2 capture. The power consumption for the different design solutions have been stored during the PSA simulator runs, and are available for

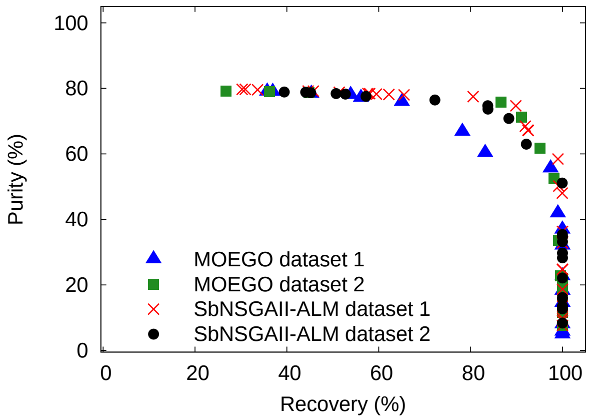


Figure 6.11: MOEGO and SbNSGA-ALM for the purity-recovery, that started with the same initial training data of size 16.

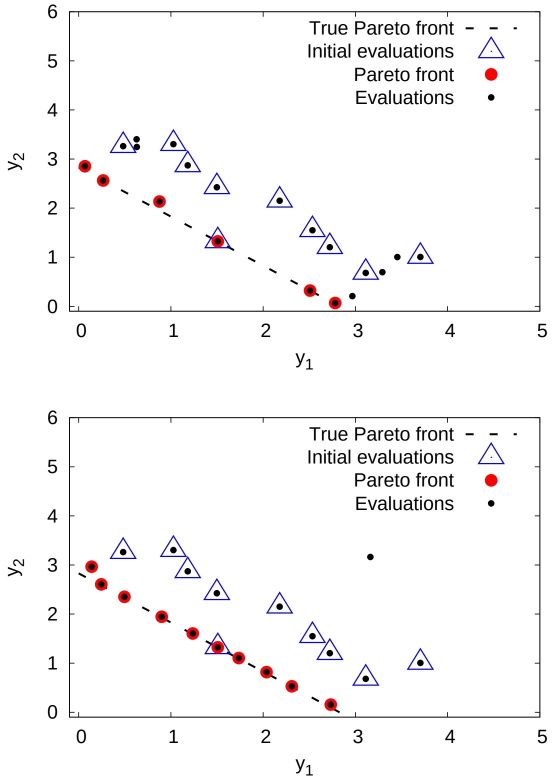


Figure 6.12: A simple optimisation problem (not the PSA problem) to illustrate the effect of the correlation parameter choice, $\boldsymbol{\xi} = (1, 1)^T$ (Top) and $\boldsymbol{\xi} = (70.7, 70.7)^T$ (Bottom), on the MOEGO performance. The intial training data consists of 10 initial evaluations (design solutions), and 10 additional selected sequentially with MOEGO.

analysis. Figure 6.15 suggests that roughly equal compromises between the purity and recovery is actually a low power consumption regime. There is a negative correlation between $\min\{y_{purity}(\boldsymbol{x}), y_{recovery}(\boldsymbol{x})\}$ and the Power consumption per mole (see Figure 6.14). That is, by attempting to achieve a moderate to high purity and recovery, simultaneously, the resulting design will be among the least demanding in terms of power usage.

The trend in Figure 6.14 could be because we have the pressure equalisation step, which increases the CO_2 purity, the most strongly adsorbed component, at the same time as the desorption is at a low pressure, which would require a relatively high compression ratio. The higher compression ratio, the lower $\mathrm{CO_2}$ purity. These opposing forces may cause the power consumption to either increase, or decrease, depending on the relative strength of these two effects.

Figure 6.16 shows an interesting relationship between segments of the Pareto front, and cluster patterns in the design space using Radial visualization (RadViz) [AKK96], see Figure 6.16. Courtesy to Antanas Zilinskas and Audrius Varoneckas at Vilnius University for identifying the clustering, but with multidimensional scaling for the same PSA data. RadViz is a widely used technique for visualising multi-dimensional data, typically to help detecting and reveal patterns hidden in the data.

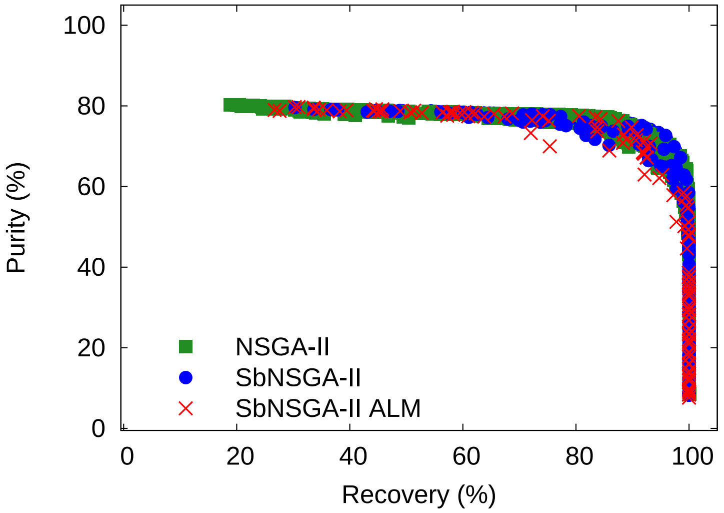


Figure 6.13: Purity vs Recovery: Comparison of NSGA-II, using 1600 design points, and SbNSGAII, as well as SbNSGAII-ALM, are using 336 design points. The results presented are the five Pareto fronts produced from using five different starting training data.

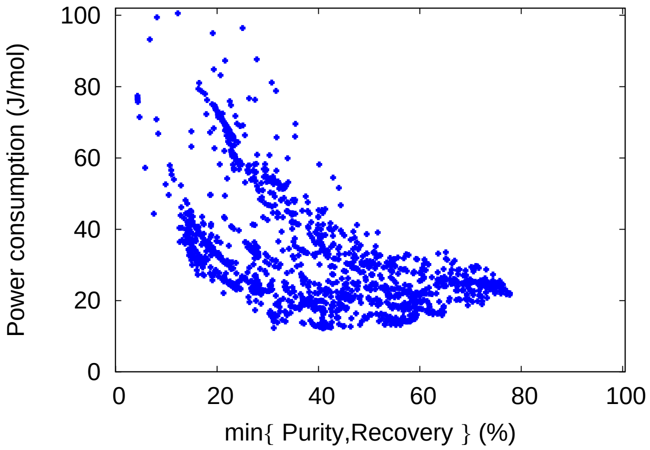


Figure 6.14: The correlation between the system's power consumption and $\min\{y_{purity}(\boldsymbol{x}), y_{recovery}(\boldsymbol{x})\}$. The design solutions are from a single NSGAII run (1600 design solutions).

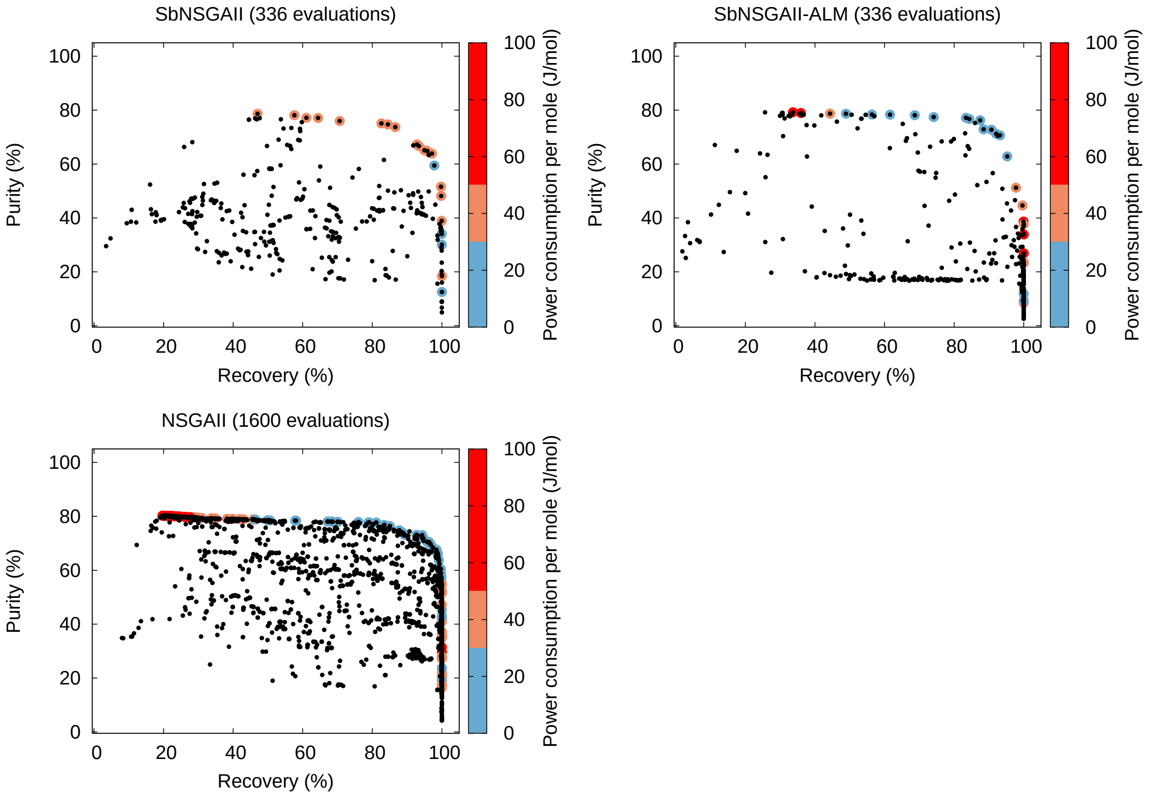


Figure 6.15: Purity vs Recovery: Comparison of NSGAII utilizing a budget of 1600 design points, and SbNSGAII, as well as SbNSGAII-ALM, a budget of 336 design points. The results presented are all the evaluations explored by the strategies. The power consumption per mole is also represented by the color of the Pareto front.

Another widely used to visualise multi-dimensional data is the parallel coordinates [Weg90]. Parallel coordinates is suitable for data with a large number of variables. In this representation each variable then corresponds to an axis. The axes are vertical (or horizontal) lines that are equidistantly placed. Each data point is represented by a polyline, crossing the axes according to its variables' values. But with more and more data points included, the lines become overlapping, causing crowding, making the visualisation difficult to view.

The use of both parallel coordinates and RadViz has been suggested in [BDS05]. This is motivated by the more complete visualisation achieved by combining the two. First, the RadViz visualisation is applied for cluster identification, then the parallel coordinates can be better used to visualise the design points belonging to the different clusters. For PSA design the cluster identification, related to some segment of the Pareto front approximation, can be useful for parametric studies, and to understand the reliability of the PSA simulation prediction (if similar performance is produced by nearby design points). With these visualisation techniques we have been able to identify clusters in the design space, here by dividing the Pareto front approximation into four distinct segments, see Figure 6.16. The design solutions displayed belong to the Pareto front calculated from all the solutions gathered from the five NSGAII runs performed in Case III.

The link between the different Pareto front segments highlighted with colours (cyan, red, green, and blue) and the design solutions are seen in Figure 6.16 to follow a pattern. This enables us to better understand the underlying process, and the impact the different design variables have on the outputs of interest. In the regime of high purity, we can see that no purge should be used in the light reflux step. This is reasonable as the $\mathrm{CO_2}$ is the heavy product taken out during the light reflux step, and to purge the bed with a gas mixture of high N_2 is likely to deteriorate the $\mathrm{CO_2}$ level of the outgoing product. Long cycle times promote high purity $\mathrm{CO_2}$ streams, but unlikely to lead to a high recovery system. Instead, we can observe that lowering the cycle time to an intermediate value, together with using a high flow rate, provides a trade-off where we expect good performance in purity and recovery simultaneously. The vacuum pressure is another factor, if the vacuum pressure is high, the $\mathrm{CO_2}$ will not be efficiently removed from the bed. The best performance in terms of $\mathrm{CO_2}$ purity is when the vacuum pressure is at its lowest point in the specified range. $\mathrm{CV}_{3/6}$ is mainly affecting the recovery; if the stem positions are more closed, the pressure change close to the vent will increase the CO_2 recovery of the system. As expected a higher temperature will improve the CO_2 uptake and in turn enrich the heavy product in CO_2. In conclusion, we can understand the PSA process much better from observing Figure 6.16, and the trade-off between purity and recovery can be explored in detail. If we design the PSA system to maximise the CO_2 concentration of the outgoing product, we can lower the amount of feed gas ($t_{feed} \times F_{feed}$) that enters the system, and increase slightly the vacuum pressure (p_{vac}) and the valve constant $CV_{3/6}$, in order to significantly improve the recovery without sacrificing too much in terms of purity. To increase CV_7 can improve the recovery, but interestingly enough we can see that it is not required for the system to achieve high purity and recovery.

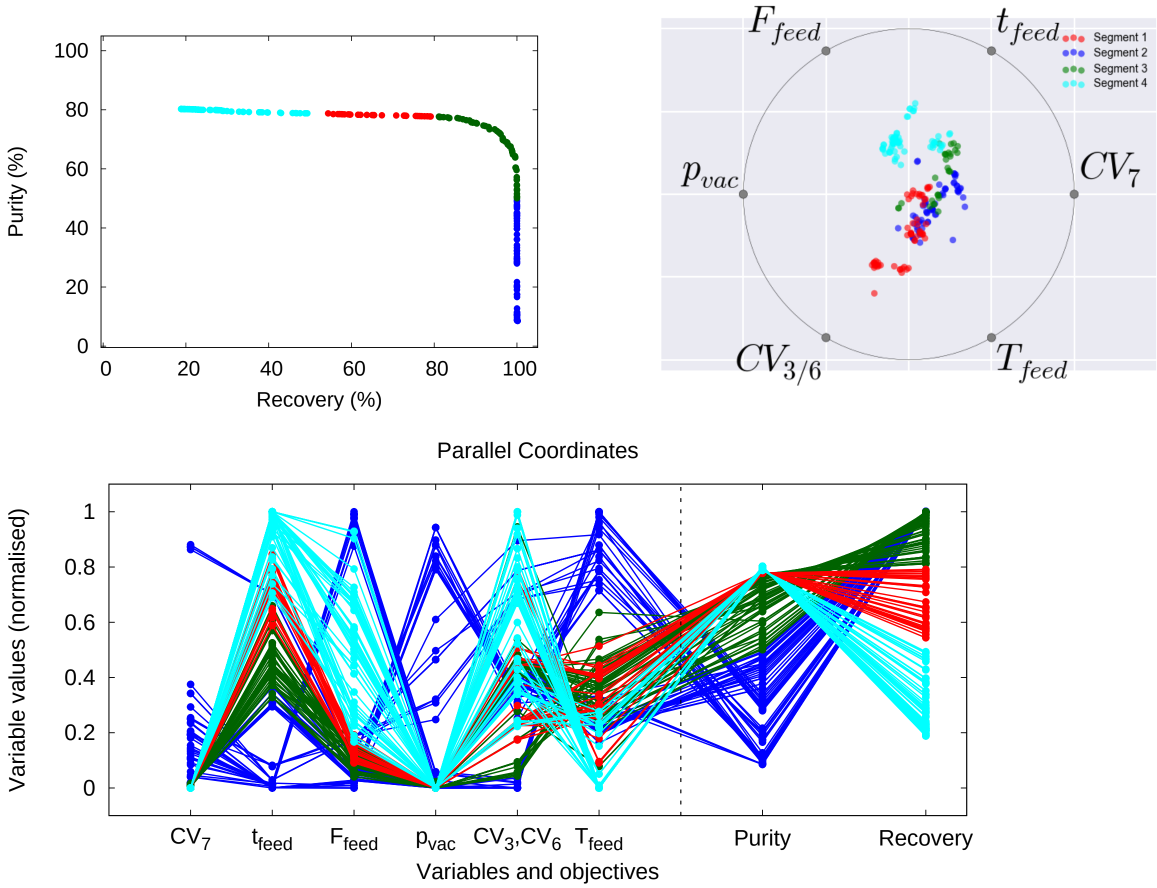


Figure 6.16: Pareto front divided into segments (Top Left) defined by different colours. RadViz (Top Right) and the parallel coordinates (Bottom) visualisation together show that the design solutions associated with different Pareto front segments are clustered.

6.6.4 Case IV: Maximise CO_2 purity and recovery, and minimise power consumption

The power consumption is now included as an objective to be minimised. Then the design problem is with three objectives: maximise the CO_2 purity and recovery, while minimising the system's power usage. The use of three objectives are usually prohibited because of the computational effort required. At most two objectives have been considered simultaneously for any PSA optimisation study in literature. The SBO method accommodates for the use of the Pareto approach in this case. The results obtained by using SbNSGAII-ALM, and NSGA-II is presented in Figure 6.18.

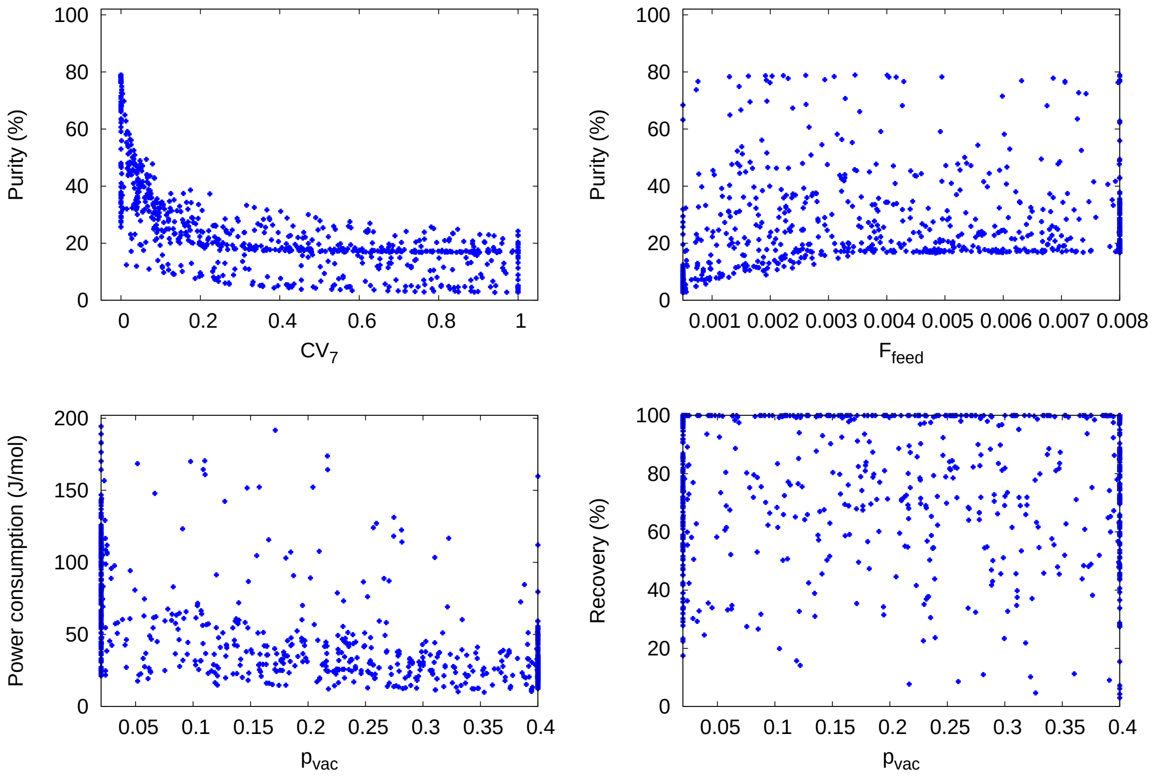


Figure 6.17: Sensitivity analysis using scatter plots for results obtained with SbNSGAII for three objectives, using 800 design solutions.

The SbNSGAII(-ALM) for purity-recovery-power achieved equally good Pareto front approximation as in Case III (purity-recovery) without the need to perform additional PSA simulations. SbNSGAII-ALM is shown to be scalable to more than two objectives. Also, the SbNSGAII-ALM exhibits a much faster convergence towards the true Pareto front than the NSGAII. Moreover, by using the results gathered using SbNSGA (800 design solutions), we can compare the design-to-output pattern of the design solutions given as scatter plots, against the global sensitivity analysis presented in Section 6.5 that used total effect based on Sobol' indices.

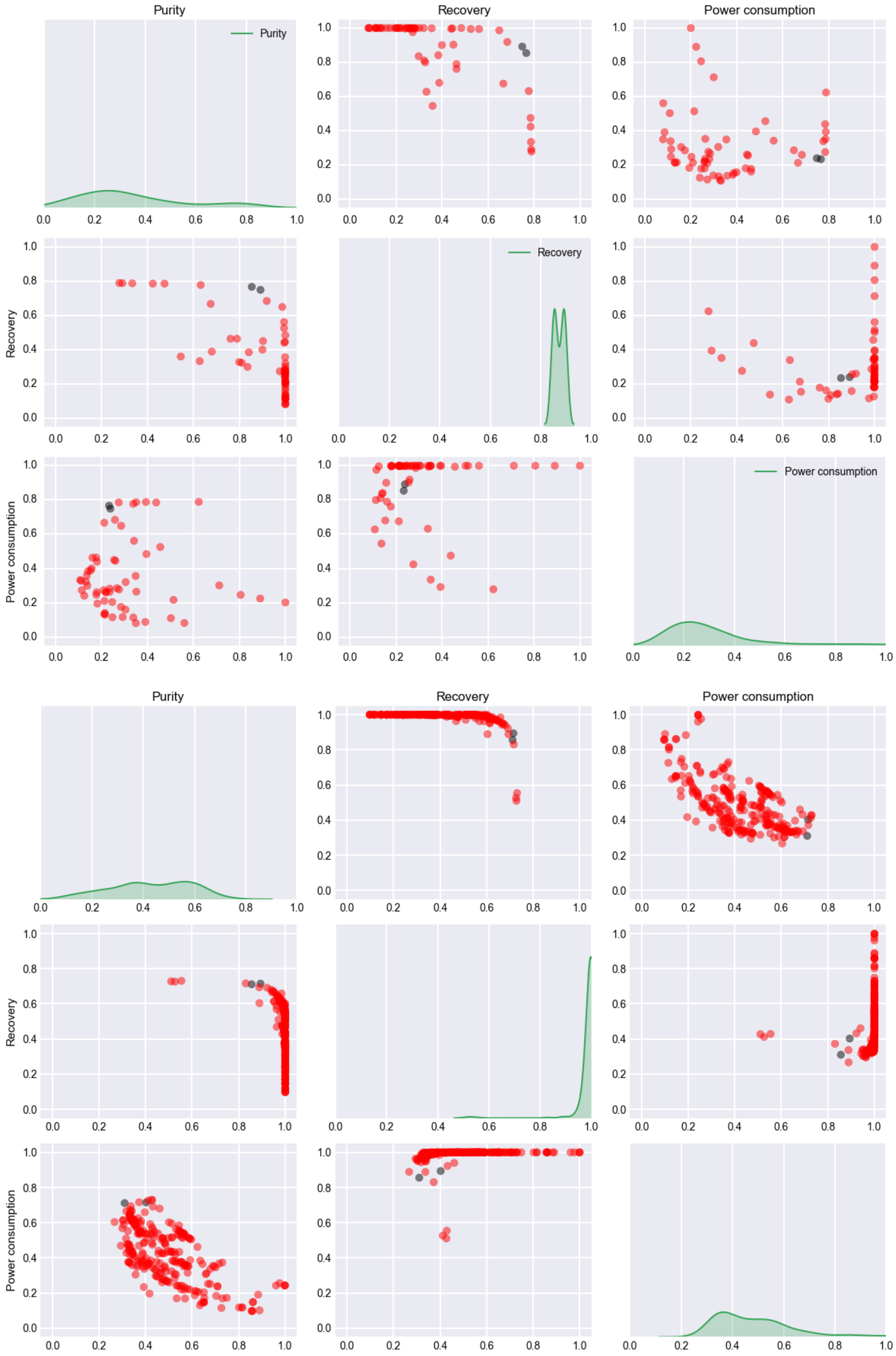


Figure 6.18: Optimised with the Pareto approach for three objectives: product purity, recovery, and power consumption. The results are the Pareto front approximation presented as a scatter plot for SbNSGAII-ALM (Top) with 336 points, and original NSGA-II (Bottom) with 1600 points. The dark grey points are the solutions satisfying product purity $> 70\%$, and recovery $> 85\%$.

For this, we have selected a few representative scatter plots for the input-output relationships of the solutions obtained with SbNSGAII. The scatter plots given in Figure 6.17 strengthen our belief in the result obtained through our preliminary global sensitivity analysis, using the Kriging on only 64 PSA simulation runs. The pressure vacuum p_{vac} has a signficant effect on the power consumption, but little effect on the product recovery, whereas we can see the huge impact purge-to-feed ratio, CV_7, has on the product purity. The feed, F_{feed}, has some affect on the purity of product. With the SbNSGAII-ALM we had 24 PSA simulation failures out of the total of 800 simulation runs, and NSGAII had 32. These are representative numbers for all our runs, that is, roughly 5% of all simulation runs lead to some internal failure. Based on these numbers, using the procedure presented in Section 5.4 for preventing revisits to the design points that already caused failure, is worth implementing.

From the parallel coordinate system, showed in Figure 6.16, for the different design solutions, we made the following observations: the recovery of the product increases with the temperature, and with the vacuum pressure. The purity is increasing with the cycle time, in contrast to the recovery, and is high when purge-to-feed valve is closed. The best region of performance in terms of both product purity and recovery, would be with no purge to feed, intermediate cycle length, low feed flow rate, low vaccum pressure, little throughput on CV_3 and CV_6, while at temperature at the high end of the range.

6.6.5 Comparison in terms of computation effort

When using Kriging surrogate models, the computational cost of fitting a Kriging model is the most expensive operation, it is proportional to m^3, where m is the number of design points available to build the surrogate. The algorithm cost between the SBO method for single objective problems, and for multiple objectives, typically differs by a factor of the number of objectives. This is because for each objective function, we build a separate Kriging model. The construction of the Kriging is the most demanding part of the SBO method. Also to consider is that the single objective problem requires the use of for example a real-coded GA, whereas we used NSGA-II when more than one objective were used. The latter is more costly for large population sizes as the Pareto front has to be determined many times. The computational cost to run the PSA simulator should is important when deciding on what algorithm to use. If the PSA simulator is fast, the algorithm cost for SBO has to be small otherwise the method is not efficient. This is because the algorithm cost of the optimisation procedure should be much smaller than the cost to calculate the objective functions, and constraints.

In this case eaxmple, the PSA simulator is moderately expensive to run, between 10 min to an hour. A simple cost analysis reveals that the algorithm cost for all the SBO methods used in this investigation is low enough to make the number of simulator runs to be the dominant important factor in determining the computational efficiency of performing the optimisation. Because the MLE fit operation is only performed until the design set is of size 80, we exclude the MLE fit operation cost in this analysis. If it would have been considered, this cost would be proportional to the number of "expensive" objective and constraint functions that are modelled with Kriging.

Table 6.7: CPU time for the SBO algorithms at the iteration with 200 design points.

Single objective:	$\sim$ 20s
Single objective with constraint:	$\sim$ 100s
Bi-objective:	$\sim$ 100s
Tri-objective:	$\sim$ 200s

We are comparing the SBO algorithm costs when using 200 design points, given the settings used in this case example. The SBO algorithm costs are presented in Table 6.7. This shows that a single-objective SBO method is computationally cheaper. Moreover, the computational effort of a single objective formulation with a single constraint, which requires two Kriging models, is of the same order as the effort needed by the bi-objective SBO. The SBO for three objectives are only twice as expensive as the bi-objective case. As the PSA simulator in this case example takes on average about 30 min to evaluate, the selection in each iteration mounts to about 5% of the total cost. We advice that computational cost to run the PSA simulation is evaluated, and after that decide what approach that is most advantageous not only in terms of the number of PSA simulation runs needed to find a desired solution, but also considered the computational effort required when using the Kriging model.

6.7 Concluding remarks

We have shown that the SBO procedure developed for multi-objective optimisation works for a real, challenging, case study on PSA for carbon capture. We have applied the same procedure for different optimisation formulations, and we showed that for all cases this new procedure is robust and much faster in finding good design solutions than the genetic algorithms. It can be an order of magnitude faster in terms of number of PSA simulation runs required. The optimisation problem has been specified with both computationally expensive objectives and constraints. Based on the results obtained for the different cases, it seems like multi-objective optimisation of the purity and recovery, is preferred to a single objective optimisation of the purity, with recovery constraint. Similarly, the three-objective SBO performs as well as the bi-objective SBO. The approximation quality in the beginning of the optimisation for the product recovery and power consumption, is above 10% (we use 16 points to initialise the optimisation), which is not particularly accurate for being a model, but the important aspect is that the accuracy improves as the search progresses. Also, the optimisation is robust against any issue caused by low accuracy in the beginning, since the surrogate models are only used to assist the search, not fully replace the objectives (or constraints). We presented a novel computationally-efficient global sensitivity analysis with gPC on Kriging instead of Monte Carlo with Sobol' sequences. This sensitivity analysis can be useful for selecting the design variables to be considered during the optimisation. The visualisation procedure to analyse the design solutions is new, that is, the link between segments of the Pareto front representation in the objective space and the corresponding design solutions in the parallel coordinate system. This visualisation procedure can be very helpful even for other applications.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

PSA offers a broad range of design possibilities influencing the system behaviour. To understand the capability of PSA for different tasks, we wish to assess its performance for different design configurations, typically via PSA computer simulation of some rigorous mathematical description of PSA processes. To perform this assessment, there is a need for efficient optimisation, since the computer simulations are computationally expensive to evaluate.

Over the past decade, much attention has been devoted towards PSA optimisation for design, in particular of complex multi-bed/multi-step PSA cycles. More recently, there has been some new design frameworks, such as the 2-bed superstructure [ABZ10b], for optimal design. However, there has been a need to develop efficient optimisation methods for this end. Some of these efforts are the use of a trust-region approach with a reduced-order model [ABZ09], and a simple surrogate-based optimisation strategy for VSA design [HKF$^+$11]. However, a more efficient optimisation approach is desirable for large-scale PSA optimisation. One such application is the investigation of PSA as a means for carbon capture. Most optimisation methods for PSA design, such as the ones above, are not equipped for multi-criteria design in the Pareto optimal sense. The ones that are suitable for multi-criteria design are computationally inefficient, e.g. straight-forward use of multi-objective evolutionary algorithms.

The focus of this thesis is on the development of efficient optimisation that can be helpful in the long term for the Carbon Capture and Storage (CCS) effort in the choice of gas separation technique. Our work has shown that surrogate modelling can be used to improve already existing optimisation techniques, by reducing the number of PSA simulation runs required to find good design solutions. By using the SBO procedure proposed in Chapter 4, which was further improved in Chapter 5, robust single and multi-objective optimisation can be performed more efficiently as fewer simulation runs are needed. This procedure, when used with genetic algorithms, can attain the nice features of those algorithms, especially for multi-objective optimisation.

Few studies have attempted optimisation of PSA systems for multi-criteria design. The two most used design criteria, the product purity and recovery, are known to be conflicting, which motivates multi-criteria design. Because PSA is currently investigated as a gas separation technique in the CCS process, a Pareto front approximation can be useful in the decision making process. For the challenging multi-criteria design problem of a 2-bed/6-step PSA system, given in Chapter 6, even three design criteria were optimised simultaneously, which is considered to be a huge challenge for optimisers when the objectives are computationally expensive. SbNSGAII-ALM applied to the 2-bed/6-step PSA system for various design formulations, has been demonstrated to be efficient and robust. The SBO procedure can be used together with any type of PSA simulation approach. In this work we used a black-box optimisation framework, together with the successive substitution as the simulation strategy, both highly stable, but often rejected because of being computationally intractable for large-scale optimisation.

Chapter 5 presented some novel improvements of the SBO method. For instance, the transformation of the Kriging for the product purity and recovery is novel, and to our knowledge not used before. This transformation has been shown to be important for multi-objective optimisation with surrogate models, for engineering problems where we have physical constraints on the outputs. Also, the improvement made by introducing the knowledge about the PSA simulation failures, in the Kriging variance, is an essential modification that promotes global search. The use of ALM gives an advantage, as some surrogate based optimisation methods are overconfident in the predictive mean of the Kriging, which can be misleading. In a multi-criteria setting, SBO with ALM tends to generate a higher diversity in the Pareto solutions.

In Chapter 6 we performed global sensitivity analysis with the Sobol' method. The Sobol' method is known to be computationally demanding as it requires many multi-dimensional integrations, for this typically Monte Carlo (MC) sampling is performed. Instead we proposed a novel approach, that uses a novel sparse grid based polynomial chaos method [BBMF13] on the Kriging, which enables the calculation of the total sensitivity effects on the objectives used in the 2-bed/6-step PSA case study.

To conclude, the main contribution of this thesis has been the development of a SBO procedure suitable for process design under multiple objectives. This should be a helpful step to begin to address PSA design for large-scale industrial applications, which are very challenging computationally. For example, when the PSA model is used as a process unit in a power plant model for carbon capture. The PSA model is then likely to be the most computationally expensive sub-model, and the use of surrogate models can in this setting be adventurous.

7.2 Future work

The results obtained showed the use of surrogate models benefits optimisation for PSA design. The SBO procedure has been shown to be robust and efficient, see results in Chapter 4 and 6. Kriging, which is the fast-to-evaluate surrogate model, has been proven to be a valuable aid in optimisation and sensitivity analysis, and there are many other interesting research directions to explore. Here follows some suggestions on possible research directions.

- The SBO method SbNSGAII-ALM, demonstrated in Chapter 6, has been shown to be efficient means of achieving fast and robust optimisation of PSA systems for design. This efficient optimisation method should be possible to incorporate directly in one of the new flowsheet design frameworks developed in [FFB09a, ABZ10b] for assessing PSA for carbon capture in a large-scale effort, ranging over a wide range of cycle schedules, design configurations, and operating conditions. Also if combined with e.g. mixed-integer programming the adsorbent materials can be included as a decision variable in the design problem.
- The design of a PSA system should ideally be assessed in the setting it should be used, which allows more relevant design criteria for the CCS process. This means that if the PSA model is embedded in the carbon capture post-combustion unit of a steady-state power plant model, the performance can be assessed in terms of the performance of the carbon capture unit, and the power plant's power generation efficiency with and without the carbon capture integration.
- In Chapter 6 we have showed that multi-objective optimisation can be performed more efficiently with the use of surrogate models for two and three objectives. An extension of this work could be to investigate when to consider the multi-objective Pareto approach over the single-objective optimisation for the problem of minimising the power consumption with constraints on the product purity and recovery.

Even though the SBO procedure has been shown to be robust and efficient, when applied as in SbGA-ALM and SbNSGAII-ALM, there are some possible improvements to the SBO procedure that could be interesting:

- One could combine a targeted optimisation approach suited for process systems design [WSF04] with SBO to make what would be “Efficient Targeted Optimisation”, ETO. The issue of designing a surrogate model to be more accurate in targeted regions has been address before, however not with the surrogate models working in conjunction with the optimisation algorithm to target regions of mutual interest. In this procedure the “targeted” region is considered to be the “feasible” region. In this framework a design point is labelled feasible or infeasible depending on the model’s response, e.g. whether or not the model fails, or the response values do not satisfy certain criteria. The choice of feasible region criteria is critical, the failure to remove the ”badly-behaved” regions can cause the surrogate model performance to deteriorate. The feasible region criteria can be chosen based on expert knowledge or through the use of data analysis. The feasible region is often not known beforehand, and from point-wise computation of the original model before and during the optimisation, a feasbile region approximation can be established. To identify the feasible region is not a trivial task, as the region may be irregular and non-convex. However, the use of data analysis and visualisation techniques has been shown to be very helpful for feasible region approximation, e.g., convex hull, search cones, and scan circles [BWF07]. For this the surrogate model can be used to more efficiently identify the feasible region. The feasible region approximation attemps to include as many feasible points as possible; although in this process, infeasible points may also become covered by the hull. The feasible region can be approximated by linear constraint cuts. The linear constraint cuts are determined in an iterative and automated fashion using the Scan Circle Algorithm (see, e.g., [WSF03]), a cluster identification technique with parallel coordinates. The optimisation method considered can be e.g. the real-coded GA, or NSGA-II for multiple objectives, with operators (crossover, mutation, etc.) tailored to produce the majority of offsprings in the targeted regions, and the surrogate model, a Kriging model, is updated using the “best” solution found, if feasible. At the end of each iteration the feasible region appproximation should be refined.
- Chapter 6 showed that the Kriging can be useful when computing the Sobol’ method for global sensitivity analysis. The use of global sensitivity analysis for screening the design variables before optimisation can be benefical when you have design problems with many design variables (> 10).

Bibliography

[AB05] H. Ahn and S. Brandani. A new numerical method for accurate simulation of fast cyclic adsorption processes. *Adsorption*, 11:113–122, 2005.

[ABW99] B.K. Arumugam, J.F. Banks, and P.C. Wankat. Pressure effects in adsorption systems. *Adsorption*, 5:261–278, 1999.

[ABZ09] A. Agarwal, L.T. Biegler, and S.E. Zitney. Simulation and optimization of pressure swing adsorption systems using reduced-order modeling. *Industrial & Engineering Chemistry Research*, 48(5):2327–2343, 2009.

[ABZ10a] A. Agarwal, L.T. Biegler, and S.E. Zitney. Superstructure-based optimal synthesis of pressure swing adsorption cycles for precombustion CO_2 capture. *Industrial & Engineering Chemistry Research*, 49(11):5066–5079, 2010.

[ABZ10b] A. Agarwal, L.T. Biegler, and S.E. Zitney. A superstructure-based optimal synthesis of PSA cycles for post-combustion CO_2 capture. *AIChE Journal*, 56(7):1813–1828, 2010.

[AC12] I. Andrianakis and P.G. Challenor. The effect of the nugget on gaussian process emulators of computer models. *Computational Statistics & Data Analysis*, 56(12):4215 – 4228, 2012.

[Adm10] U.S. Energy Information Administration. International energy outlook 2010. Technical Report EIA-0484, U.S. Department of Energy, 2010.

[AFR02] R. Arvind, S. Farooq, and D.M. Ruthven. Analysis of a piston PSA process for air separation. *Chemical Engineering Science*, 57(3):419 – 433, 2002.

[Aga10] A. Agarwal. *Advanced Strategies for Optimal Design and Operation of Pressure Swing Adsorption Processes*. PhD thesis, PhD Thesis. Carnegie Mellon University, 2010.

[AJ05] A. Abraham and L. Jain. *Evolutionary multiobjective optimization*. Springer, 2005.

[AKK96] M. Ankerst, D.A. Keim, and H-P Kriegel. Circle Segments: A technique for visually exploring large multidimensional data sets. In *Proc. Visualization 96, Hot Topic Session, San Francisco, CA, 1996*, volume 1501, page 1, 1996.

[ALY06] M. Atiquzzaman, S-Y. Liong, and X. Yu. Alternative decision making in water distribution network with NSGA-II. *Journal of water resources planning and management*, 132(2):122–126, 2006.

[AS92] E. Alpay and D.M. Scott. The linear driving force model for fast-cycle adsorption and desorption in a spherical particle. *Chemical Engineering Science*, 47(2):499–502, 1992.

[AT05] D. Aaron and C. Tsouris. Separation of CO_2 from flue gas: A review. *Separation Science and Technology*, 40(1-3):321–348, 2005.

[ATL11] S. Ayoub, D. Tondeur, and M.A. Latifi. New formulation of optimization-based simulation of a pressure swing adsorption process: Hybrid dynamic optimization. *AIChE Journal*, 57(5):1367–1373, 2011.

[BB13] S. Bandyopadhyay and R. Bhattacharya. Applying modified NSGA-II for bi-objective supply chain problem. *Journal of Intelligent Manufacturing*, 24(4):707–716, 2013.

[BBLP05] T. Bartz-Beielstein, C.W.G. Lasarczyk, and M. Preuß. Sequential parameter optimization. In *Evolutionary Computation*, volume 1, pages 773–780. IEEE, 2005.

[BBMF13] S. Brown, J. Beck, H. Mahgerefteh, and E.S. Fraga. Global sensitivity analysis of the impact of impurities on CO2 pipeline failure. *Reliability Engineering & System Safety*, 115:43–54, 2013.

[BCP89] Kathryn Eleda Brenan, Stephen La Vern Campbell, and Linda Ruth Petzold. *Numerical solution of initial-value problems in differential-algebraic equations*, volume 14. Siam, 1989.

[BDMS08] J. Branke, K. Deb, K. Miettinen, and R. Slowinski. *Multiobjective optimization: Interactive and evolutionary approaches*, volume 5252. Springer, 2008.

[BDS05] E. Bertini, L. Dell'Aquila, and G. Santucci. SpringView: cooperation of RadViz and parallel coordinates for view optimization and clutter reduction. In *Proc. 3rd International Conference on Coordinated and Multiple Views in Exploratory Visualization*, pages 22–29. IEEE, 2005.

[Ber66] N.H. Berlin. Technical Report U.S. Patent 3,280,536., 1966.

[BFB$^+$12] J. Beck, D. Friedrich, S. Brandani, S. Guillas, and E.S. Fraga. Surrogate based optimisation for design of pressure swing adsorption systems. *Proceedings of the 22nd European Symposium on Computer Aided Process Engineering*, (June):1217–1221, 2012.

[BG04] H-J. Bungartz and M. Griebel. Sparse grids. *Acta numerica*, 13:147–269, 2004.

[BGHvBW03] L.T. Biegler, O. Ghattas, M. Heinkenschloss, and B. van Bloemen Waanders. *Large-scale PDE-constrained optimization: an introduction*. Springer, 2003.

[BGW85] L.T. Biegler, I.E. Grossmann, and A.W. Westerberg. A note on approximation techniques used for process optimization. *Computers & Chemical Engineering*, 9(2):201 – 206, 1985.

[BJ11] S. Ba and R.V. Joseph. Multi-layer designs for computer experiments. *Journal of the American Statistical Association*, 106(495), 2011.

[BJF05] L.T. Biegler, L. Jiang, and G.V. Fox. Recent advances in simulation and optimal design of pressure swing adsorption systems. *Separation & Purification Reviews*, 33(1):1–39, 2005.

[BM10] P.M. Budd and N.B. McKeown. Highly permeable polymers for gas separation membranes. *Polymer Chemistry*, 1:63–68, 2010.

[BNE07] N. Beume, B. Naujoks, and M. Emmerich. SMS-EMOA: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, 181(3):1653–1669, 2007.

[BNR00] V. Barthelmann, E. Novak, and K. Ritter. High dimensional polynomial interpolation on sparse grids. *Advances in Computational Mathematics*, 12(4):273–288, 2000.

[BNS95] L.T. Biegler, J. Nocedal, and C. Schmid. A reduced hessian method for large-scale constrained optimization. *SIAM Journal on Optimization*, 5(2):314–347, 1995.

[BNT07] I. Babuška, F. Nobile, and R. Tempone. Reliability of computational science. *Numerical methods for partial differential equations*, 23(4):753–784, 2007.

[BNT10] I. Babuška, F. Nobile, and R. Tempone. A stochastic collocation method for elliptic partial differential equations with random input data. *SIAM Review*, 52(2):317–355, 2010.

[BNTT11] J. Bäck, F. Nobile, L. Tamellini, and R. Tempone. Stochastic spectral Galerkin and collocation methods for PDEs with random coefficients: a numerical comparison. In *Spectral and High Order Methods for Partial Differential Equations*, pages 43–62. Springer, 2011.

[BPS99] F.P. Bernardo, E.N. Pistikopoulos, and P.M. Saraiva. Integration and computational issues in stochastic design and planning optimization problems. *Industrial and Engineering Chemistry Research*, 38(8):3056–3068, 1999.

[Bro65] C.G. Broyden. A class of methods for solving nonlinear simultaneous equations. *Mathematics of Computation*, 19(92):577–593, 1965.

[BSMD08] S. Bandyopadhyay, S. Saha, U. Maulik, and K. Deb. A simulated annealing-based multiobjective optimization algorithm: AMOSA. *Evolutionary Computation*, 12(3):269–283, 2008.

[Bur76] R.L. Burwell. Manual of symbols and terminology for physicochemical quantities and units - appendix II. definitions, terminology and symbols in colloid and surface chemistry. part II: Heterogeneous catalysis. *Pure and Applied Chemistry*, 46(1):71–90, 1976.

[BWF07] R.A. Bates, H.P. Wynn, and E.S. Fraga. Feasible region approximation: a comparison of search cone and convex hull methods. *Engineering Optimization*, 39(5):513–527, 2007.

[CCL02] C.A. Coello Coello and M.S. Lechuga. MOPSO: A proposal for multiple objective particle swarm optimization. In *Evolutionary Computation*, volume 2, pages 1051–1056. IEEE, 2002.

[CG08] J.A. Caballero and I.E. Grossmann. An algorithm for the use of surrogate models in modular flowsheet optimization. *AIChE Journal*, 54(10):2633–2650, 2008.

[CH83] V. Chankong and Y.Y. Haimes. *Multiobjective Decision Making Theory and Methodology*. Elsevier Science, New York, 1983.

[Chi88] A.S.T. Chiang. Arithmetic of PSA process scheduling. *AIChE Journal*, 34(11):1910–1912, 1988.

[CJ97] D.D. Cox and S. John. SDO: A statistical method for global optimization. *Multidisciplinary Design Optimization: State of the Art*, pages 315–329, 1997.

[CKY$^+$95] K.T. Chue, J.N. Kim, Y.J. Yoo, S.H. Cho, and R.T. Yang. Comparison of activated carbon and zeolite 13X for CO2 recovery from flue gas by pressure swing adsorption. *Industrial & Engineering Chemistry Research*, 34(2):591–598, 1995.

[CKY$^+$03] W-K. Choi, T-I. Kwon, Y-K. Yeo, H. Lee, H.K. Song, and B-K. Na. Optimal operation of the pressure swing adsorption (PSA) process for CO2 recovery. *Korean Journal of Chemical Engineering*, 20(4):617–623, 2003.

[CL94] D.T. Croft and M.D. LeVan. Periodic states of adsorption cycles I. direct determination and stability. *Chemical Engineering Science*, 49(11):1821 – 1829, 1994.

[CLVV07] C.A. Coello, G.B. Lamont, and D.A. Van Veldhuisen. *Evolutionary algorithms for solving multi-objective problems*. Springer, 2007.

[CMM05] P. Cruz, F.D. Magalhaes, and A. Mendes. On the optimization of cyclic adsorption separation processes. *AIChE journal*, 51(5):1377–1395, 2005.

[CMM06] P. Cruz, F.D. Magalhes, and A. Mendes. Generalized linear driving force approximation for adsorption of multicomponent mixtures. *Chemical Engineering Science*, 61(11):3519 – 3531, 2006.

[Coh96] D.A. Cohn. Neural network exploration using optimal experiment design. *Neural Networks*, 9(6):1071 – 1083, 1996.

[Cre89] N. Cressie. Geostatistics. *The American Statistician*, 43(4):197–202, 1989.

[CS98] M.B. Cutlip and M. Shacham. The numerical method of lines for partial differential equations. *CACHE News*, 47:18–21, 1998.

[CSMM03] P. Cruz, J.C. Santos, F.D. Magalhaes, and A. Mendes. Cyclic adsorption separation processes: analysis strategy and optimization procedure. *Chemical Engineering Science*, 58(14):3143 – 3158, 2003.

[CSMM05] P. Cruz, J.C. Santos, F.D. Magalhaes, and A. Mendes. Simulation of separation processes using finite volume method. *Computers & chemical engineering*, 30(1):83–98, 2005.

[CSV09] A.R Conn, K. Scheinberg, and L.N. Vicente. *Introduction to derivative-free optimization*, volume 8. SIAM, 2009.

[CT06] T.M. Cover and J.A. Thomas. *Elements of Information Theory*. Wiley Series in Telecommunications and Signal Processing. Wiley-Interscience, 2 edition, July 2006.

[DA00] Y. Ding and E. Alpay. Equilibria and kinetics of CO_2 adsorption on hydrotalcite adsorbent. *Chemical Engineering Science*, 55(17):3461–3474, 2000.

[DAPM00] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. *Lecture Notes in Computer Science*, 1917:849–858, 2000.

[Das97] I. Das. *Nonlinear Multicriteria Optimization and Robust Optimality*. PhD thesis, Rice University, Houston, Texas, 1997.

[Dav13] M.E. Davis. *Numerical methods and modeling for chemical engineers*. Courier Corporation, 2013.

[DCL02] Y.Q. Ding, D.T. Croft, and M.D. LeVan. Periodic states of adsorption cycles IV. direct optimization. *Chemical Engineering Science*, 57(21):4521–4531, 2002.

[Deb00] K. Deb. An efficient constraint handling method for genetic algorithms. *Computer Methods in Applied Mechanics and Engineering*, 186(2):311–338, 2000.

[Deb01] K. Deb. *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons, Inc., New York, NY, USA, 2001.

[DFB13] W. Dang, D. Friedrich, and S. Brandani. Characterisation of an automated dual piston pressure swing adsorption (DP-PSA) system. *Energy Procedia*, 37:57–64, 2013.

[DG96] K. Deb and M. Goyal. A combined genetic adaptive search (GeneAS) for engineering design. *Computer Science and Informatics*, 26(4):30–45, 1996.

[DGH95] D. Diagne, M. Goto, and T. Hirose. Parametric studies on CO_2 separation and recovery by a dual reflux PSA process consisting of both rectifying and stripping sections. *Industrial & Engineering Chemistry Research*, 34(9):3083–3089, 1995.

[DL01] Y. Ding and D.M. LeVan. Periodic states of adsorption cycles III. convergence acceleration for direct determination. *Chemical Engineering Science*, 56(17):5217 – 5230, 2001.

[DPAM02] K. Deb, A. Pratap, S. Agarwal, and T.A.M.T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *Evolutionary Computation*, 6(2):182–197, 2002.

[DR07] P. Diggle and P.J. Ribeiro. *Model-based geostatistics*. Springer, 2007.

[DS96] J.E. Dennis and R.B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations (Classics in Applied Mathematics, 16)*. Society for Industrial & Applied Mathematics, 1996.

[DVB12] A.W. Dowling, S.R.R. Vetukuri, and L.T. Biegler. Large-scale optimization strategies for pressure swing adsorption cycle synthesis. *AIChE Journal*, 58(12):3777–3791, 2012.

[EDK11] M.T.M. Emmerich, A.H. Deutz, and J.W. Klinkenberg. Hypervolume-based expected improvement: Monotonicity properties and exact computation. In *Evolutionary Computation*, pages 2147–2154. IEEE, 2011.

[EEHJ96] K. Eriksson, D. Estep, P. Hansbo, and C. Johnson. *Computational Differential Equations*. Press Syndicate of the University of Cambridge, 1996.

[EGN06] M.T.M. Emmerich, K.C. Giannakoglou, and B. Naujoks. Single-and multiobjective evolutionary optimization assisted by Gaussian random field metamodels. *Evolutionary Computation*, 10(4):421–439, 2006.

[EN04] M. Emmerich and B. Naujoks. Metamodel assisted multiobjective optimisation strategies and their application in airfoil design. In *Adaptive computing in design and manufacture VI*, pages 249–260. Springer, 2004.

[ER09] A. Ebner and J. Ritter. State-of-the-art adsorption and membrane separation processes for carbon dioxide production from carbon dioxide emitting industries. *Separation Science and Technology*, 44:1273–1421, 2009.

[Fau99] L.V. Fausett. *Applied Numerical Analysis Using MATLAB*. Pearson Prentice Hall, 1999.

[fBISB07] Department for Business Innovation & Skills (BEER). Competition for a carbon dioxide capture and storage demonstration, 2007.

[FF93] C.M. Fonseca and P.J. Fleming. Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In *Proceedings of the 5th International Conference on Genetic Algorithms*, pages 416–423. Morgan Kaufmann Publishers Inc., 1993.

[FFB09a] G. Fiandaca, E.S. Fraga, and S. Brandani. Development of a flowsheet design framework of multi-step PSA cycles for CO_2 capture. In *10th International Symposium on Process Systems Engineering: Part A*, volume 27 of *Computer Aided Chemical Engineering*, pages 849–854. Elsevier, 2009.

[FFB09b] G. Fiandaca, E.S. Fraga, and S. Brandani. A multi-objective genetic algorithm for the design of pressure swing adsorption. *Engineering Optimization*, 41(9):833–854, 2009.

[FFB13] D. Friedrich, M-C. Ferrari, and S. Brandani. Efficient simulation and acceleration of convergence for a dual piston pressure swing adsorption system. *Industrial & Engineering Chemistry Research*, 2013.

[FFP$^+$08] J.D. Figueroa, T. Fout, S. Plasynski, H. McIlvried, and R.D. Srivastava. Advances in CO_2 capture technology–the U.S. Department of Energy's carbon sequestration program. *International Journal of Greenhouse Gas Control*, 2(1):9 – 20, 2008.

[FGI$^+$13] L. Formaggia, A. Guadagnini, I. Imperiali, V. Lever, G. Porta, M. Riva, A. Scotti, and L. Tamellini. Global sensitivity analysis through polynomial chaos expansion of a basin-scale geochemical compaction model. *Computational Geosciences*, 17(1):25–42, 2013.

[Fia10] G. Fiandaca. *A multi-criteria design framework for the synthesis of complex pressure swing adsorption cycles for CO2 capture*. PhD thesis, PhD Thesis. UCL (University College London), 2010.

[FK09] A.I.J. Forrester and A.J. Keane. Recent advances in surrogate-based optimization. *Progress in Aerospace Sciences*, 45(1):50–79, 2009.

[FSK08] A. Forrester, A. Sobester, and A.J. Keane. *Engineering Design Via Surrogate Modelling: A Practical Guide*. Wiley Blackwell, 2008.

[FTR98] S. Farooq, C. Thaeron, and D.M. Ruthven. Numerical simulation of a parallel-passage piston-driven PSA unit. *Separation and Purification Technology*, 13(3):181 – 193, 1998.

[GC47] E. Glueckauf and J. I. Coates. 241. theory of chromatography. part IV. the influence of incomplete equilibrium on the front boundary of chromatograms and on the effectiveness of separation. *Journal of the Chemical Society*, pages 1315–1321, 1947.

[GL09] R.B. Gramacy and H.K.H. Lee. Adaptive design and analysis of supercomputer experiments. *Technometrics*, 51(2):130–145, 2009.

[GLRC10] D. Ginsbourger, R. Le Riche, and L. Carraro. Kriging is well-suited to parallelize optimization. In *Computational Intelligence in Expensive Optimization Problems*, pages 131–162. Springer, 2010.

[Gol62] A.S. Goldberger. Best linear unbiased prediction in the generalized linear regression model. *Journal of the American Statistical Association*, 57(298):369–375, 1962.

[Gri86] A. Griewank. *Rates of convergence for secant methods on nonlinear problems in Hilbert space*. Springer, 1986.

[GS91] R.G. Ghanem and P.D. Spanos. *Stochastic finite elements: a spectral approach*. Springer-Verlag, New York, USA, 1991.

[HAW08] M.T. Ho, G.W. Allinson, and D.E. Wiley. Reducing the cost of CO_2 capture from flue gases using pressure swing adsorption. *Industrial & Engineering Chemistry Research*, 47(14):4883–4890, 2008.

[HBEF12] F.M.M. Hasan, R.C. Baliban, J.A. Elia, and C.A. Floudas. Modeling, simulation, and optimization of postcombustion CO2 capture for variable feed concentration and flow rate. 2. pressure swing adsorption and vacuum swing adsorption processes. *Industrial & Engineering Chemistry Research*, 51(48):15665–15682, 2012.

[HBG$^+$05] A. C. Hindmarsh, P. N. Brown, K. E. Grant, S. L. Lee, R. Serban, D. E. Shumaker, and C. S. Woodward. Sundials: Suite of nonlinear and differential/algebraic equation solvers. *ACM Transactions on Mathematical Software*, 31(3):363–396, September 2005.

[HJ98] M.P. Hansen and A. Jaszkiewicz. *Evaluating the quality of approximations to the non-dominated set*. Department of Mathematical Modelling, Technical University of Denmark, 1998.

[HKF$^+$11] F.M.M. Hasan, I.A. Karimi, S. Farooq, A. Rajendran, and M. Amanullah. Surrogate-based VSA process optimization for post-combustion CO2 capture.

In *21st European Symposium on Computer Aided Process Engineering*, volume 29, page 402. Elsevier, 2011.

[HLAW06] M.T. Ho, G. Leamon, G.W. Allinson, and D.E. Wiley. Economics of CO2 and mixed gas geosequestration of flue gas using gas separation membranes. *Industrial & Engineering Chemistry Research*, 45(8):2546–2552, 2006.

[HM11] C.A. Henao and C.T. Maravelias. Surrogate-based superstructure optimization framework. *AIChE Journal*, 57(5):1216–1232, 2011.

[HMN$^+$13] R. Haghpanah, A. Majumder, R. Nilam, A. Rajendran, S. Farooq, I.A. Karimi, and M. Amanullah. Multiobjective optimization of a four-step adsorption process for postcombustion CO2 capture via finite volume simulation. *Industrial & Engineering Chemistry Research*, 52(11):4249–4265, 2013.

[HNG94] J. Horn, N. Nafpliotis, and D.E. Goldberg. A niched Pareto genetic algorithm for multiobjective optimization. In *Evolutionary Computation*, pages 82–87. IEEE, 1994.

[HNW91] E. Hairer, S.P. Nørsett, and G. Wanner. *Solving ordinary differential equations*, volume 2. Springer, 1991.

[HS93] M.S. Handcock and M.L. Stein. A Bayesian analysis of Kriging. *Technometrics*, 35(4):403–410, 1993.

[IOA$^+$96] M. Ishibashi, H. Ota, N. Akutsu, S. Umeda, M. Tajika, J. Izumi, A. Yasutake, T. Kabata, and Y. Kageyama. Technology for removing carbon dioxide from power plant flue gas by the physical adsorption method. *Energy Conversion and Management*, 37(6):929–933, 1996.

[IPC05] Summary for policymakers - a special report of working group III of the intergovernmental panel on climate change. Technical report, IPCC, 2005.

[JBF03] L. Jiang, L.T. Biegler, and V.G. Fox. Simulation and optimization of pressure-swing adsorption systems for air separation. *AIChE Journal*, 49(5):1140–1157, 2003.

[JBF05] L. Jiang, L.T. Biegler, and G.V. Fox. Design and optimization of pressure swing adsorption systems with parallel implementation. *Computers & Chemical Engineering*, 29(2):393 – 399, 2005.

[JFB04] L. Jiang, V.G. Fox, and L.T. Biegler. Simulation and optimal design of multiple-bed pressure swing adsorption systems. *AIChE Journal*, 50(11):2904–2917, 2004.

[Jin05] Y. Jin. A comprehensive survey of fitness approximation in evolutionary computation. *Soft computing*, 9(1):3–12, 2005.

[JLRGG12] J. Janusevskis, R. Le Riche, D. Ginsbourger, and R. Girdziusas. Expected improvements for the asynchronous parallel global optimization of expensive functions: Potentials and challenges. In *Learning and Intelligent Optimization*, pages 413–418. Springer, 2012.

[JMY90] M.E. Johnson, L.M. Moore, and D. Ylvisaker. Minimax and maximin distance designs. *Journal of Statistical Planning and Inference*, 26(2):131–148, 1990.

[JO05] S. Jeong and S. Obayashi. Efficient global optimization (EGO) for multi-objective problem and data mining. In *Evolutionary Computation*, volume 3, pages 2138–2145. IEEE, 2005.

[Jon01] D.R. Jones. A taxonomy of global optimization methods based on response surfaces. *Journal of Global Optimization*, 21(4):345–383, 2001.

[JSW98] D.R. Jones, M. Schonlau, and W.J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13:455–492, 1998.

[KC02] J. Knowles and D. Corne. On metrics for comparing nondominated sets. In *Evolutionary Computation*, volume 1, pages 711–716. IEEE, 2002.

[Kel99] C.T. Kelley. Detection and remediation of stagnation in the nelde - mead algorithm using a sufficient decrease condition. *SIAM Journal on Optimization*, 10(1):43–55, 1999.

[KFH$^+$94] R. Kumar, V.G. Fox, D.G. Hartzog, R.E. Larson, Y.C. Chen, P.A. Houghton, and T. Naheiri. A versatile process simulator for adsorptive separations. *Chemical Engineering Science*, 49(18):3115–3125, 1994.

[KK82] G.E. Keller and C.H.A. Kuo. Enhanced gas separation by selective adsorption. Technical Report US Patent 4,354,859, Assigned to Union Carbide Corporation, 1982.

[Kle09] Kleijnen, J.P.C. Kriging metamodeling in simulation: A review. *European Journal of Operational Research*, 192(3):707 – 716, 2009.

[KM02] D. Ko and I.L. Moon. Multiobjective optimization of cyclic adsorption processes. *Industrial & Engineering Chemistry Research*, 41(1):93–104, 2002.

[KN08] J. Knowles and H. Nakayama. Meta-modeling in multiobjective optimization. In *Multiobjective Optimization*, pages 245–284. Springer, 2008.

[Kno06] J. Knowles. ParEGO: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *Evolutionary Computation*, 10(1):50–66, 2006.

[Kri51] D.G. Krige. A statistical approach to some basic mine valuation problems on the witwatersrand. *Journal of the Chemical, Metallurgical and Mining Society of South Africa*, 52(6):119–139, 1951.

[KSB03] D. Ko, R. Siriwardane, and L.T. Biegler. Optimization of a pressure-swing adsorption process using zeolite 13X for CO_2 sequestration. *Industrial & Engineering Chemistry Research*, 42(2):339–348, 2003.

[KSB05] D. Ko, R. Siriwardane, and L.T. Biegler. Optimization of pressure swing adsorption and fractionated vacuum pressure swing adsorption processes for CO2 capture. *Industrial & Engineering Chemistry Research*, 44(21):8084–8094, 2005.

[KSG08] A. Krause, A. Singh, and C. Guestrin. Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies. *The Journal of Machine Learning Research*, 9:235–284, 2008.

[Kva97] Kvamsdal, H.M. and Hertzberg, T. Optimization of PSA systems - studies on cyclic steady state convergence. *Computers & Chemical Engineering*, 21(8):819 – 832, 1997.

[KVB04] J.P.C. Kleijnen and W.C.M. Van Beers. Application-driven sequential designs for simulation experiments: Kriging metamodelling. *Journal of the Operational Research Society*, 55(8):876–883, 2004.

[KW05] A.R. Kovscek and Y. Wang. Geologic storage of carbon dioxide and enhanced oil recover. I. Uncertainty quantification employing a streamline based proxy for reservoir flow simulation. *Energy Conversion and Management*, 46:1920–1940, 2005.

[Lan98] C.B Laney. *Computational gasdynamics*. Cambridge University Press, 1998.

[LeV02] R.J. LeVeque. *Finite-Volume Methods for Hyperbolic Problems*. Cambridge University Press, 2002.

[LGL$^+$11] Z. Liu, C.A. Grande, P. Li, J. Yu, and A.E. Rodrigues. Multi-bed vacuum pressure swing adsorption for carbon dioxide capture from flue gas. *Separation and Purification Technology*, 81(3):307–317, 2011.

[LHR99] Y.J. Liu, C.E. Holland, and J.A. Ritter. Solvent vapor recovery by pressure swing adsorption. III. comparison of simulation with experiment for the butane-activated carbon system. *Separation Science and Technology*, 34(8):1545–1576, 1999.

[LKL$^+$09] W. Liu, D. King, J. Liu, B. Johnson, Y. Wang, and Z. Yang. Critical material and process issues for CO_2 separation from coal-powered plants. *OM Journal of the Minerals, Metals and Materials Society*, 61(4):36–44, 2009.

[LMK10] O.P. Le Maître and O.M. Knio. *Spectral methods for uncertainty quantification: with applications to computational fluid dynamics*. Scientific Computation. Springer, New York, 2010.

[LNS02] S.N. Lophaven, H.B. Nielsen, and J. Søndergaard. DACE - a Matlab Kriging toolbox, version 2.0. Technical report, 2002.

[Loc97] M. Locatelli. Bayesian algorithms for one-dimensional global optimization. *Journal of Global Optimization*, 10(1):57–76, 1997.

[LS13] Y. Liu and F. Sun. Parameter estimation of a pressure swing adsorption model for air separation using multi-objective optimisation and support vector regression model. *Expert Systems with Applications*, 40(11):4496 – 4502, 2013.

[LST08] M.A. Latifi, D. Salhi, and D. Tondeur. Optimisation-based simulation of a pressure swing adsorption process. *Adsorption*, 14(4-5):567–573, 2008.

[LSW09] J.L. Loeppky, J. Sacks, and W.J. Welch. Choosing the sample size of a computer experiment: A practical guide. *Technometrics*, 51(4), 2009.

[LTZ06] M. Laumanns, L. Thiele, and E. Zitzler. An efficient, adaptive parameter variation scheme for metaheuristics based on the epsilon-constraint method. *European Journal of Operational Research*, 169(3):932–942, 2006.

[LZT$^+$07] W. Liu, Q. Zhang, E. Tsang, C. Liu, and B. Virginas. On the performance of metamodel assisted MOEA/D. In *Advances in Computation and Intelligence*, pages 547–557. Springer, 2007.

[MA10] R. Marler and J. Arora. The weighted sum method for multi-objective optimization: new insights. *Structural and Multidisciplinary Optimization*, 41:853–862, 2010.

[Mac92] D.J.C. MacKay. Information-based objective functions for active data selection. *Neural Computation*, 4(4):590–604, 1992.

[Mat63] G. Matheron. Principles of geostatistics. *Economic Geology*, 58:1246–1266, 1963.

[MBC79] M.D. McKay, R.J. Beckman, and W.J. Conover. Comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21(2):239–245, 1979.

[MBS$^+$01] M. Meckesheimer, R.R. Barton, T. Simpson, F. Limayem, B. Yannou, and G.V.D. Vignes. Metamodeling of combined discrete/continuous responses. *AIAA Journal American Institute of Aeronautics and Astronautics*, 39(10):1950–1959, 2001.

[MDC98] T. Mehrdad, J. Dylan, and R. Carlos. Goal programming for decision making: An overview of the current state-of-the-art. *European Journal of Operational Research*, 111(3):569 – 581, 1998.

[MER10] A. Mehrotra, A. Ebner, and J. Ritter. Arithmetic approach for complex PSA cycle scheduling. *Adsorption*, 16:113–126, 2010.

[MER11] A. Mehrotra, A. Ebner, and J. Ritter. Simplified graphical approach for complex PSA cycle scheduling. *Adsorption*, 17:337–345, 2011.

[MILR09] A. Marrel, B. Iooss, B. Laurent, and O. Roustant. Calculations of Sobol indices for the Gaussian process metamodel. *Reliability Engineering & System Safety*, 94(3):742–751, 2009.

[MIT07] The future of coal - an interdisciplinary mit study. Technical report, Massachusetts Institute of Technology (MIT), 2007.

[MLWB10] T.C. Merkel, H. Lin, X. Wei, and R. Baker. Power plant post-combustion carbon dioxide capture: An opportunity for membranes. *Journal of Membrane Science*, 359(1-2):126 – 139, 2010.

[MM95] M.D. Morris and T.J. Mitchell. Exploratory designs for computational experiments. *Journal of Statistical Planning and Inference*, 43(3):381–402, 1995.

[Moc94] J. Mockus. Application of Bayesian approach to numerical methods of global and stochastic optimization. *Journal of Global Optimization*, 4(4):347–365, 1994.

[Mon06] D.C. Montgomery. *Design and Analysis of Experiments*. John Wiley & Sons, 2006.

[MS14] W.K. Mashwani and A. Salhi. Multiobjective memetic algorithm based on decomposition. *Applied Soft Computing*, 21:221–243, 2014.

[MTS82] G.J. McRae, J.W. Tilden, and J.H. Seinfeld. Global sensitivity analysis a computational implementation of the fourier amplitude sensitivity test (FAST). *Computers & Chemical Engineering*, 6(1):15–25, 1982.

[NGGK08] D. Nikolic, A. Giovanoglou, M.C. Georgiadis, and E.S. Kikkinides. Generic Modeling Framework for Gas Separations Using Multibed Pressure Swing Adsorption Processes. *Industrial & Engineering Chemistry Research*, 47(9):3156–3169, May 2008.

[NKG09] D. Nikolic, E.S. Kikkinides, and M.C. Georgiadis. Optimization of multibed pressure swing adsorption processes. *Industrial & Engineering Chemistry Research*, 48(11):5388–5398, 2009.

[NP98] S. Nilchan and C.C. Pantelides. On the optimisation of periodic adsorption processes. *Adsorption*, 4(2):113–147, 1998.

[NS83] S. Nakao and M. Suzuki. Mass-transfer coefficient in cyclic adsorption and desorption. *Journal of Chemical Engineering of Japan*, 16(2):114–119, 1983.

[NS96] S.G. Nash and A. Sofer. *Linear and nonlinear programming*, volume 692. McGraw-Hill, New York, 1996.

[oFEoS99] Office of Fossil Energy and Office of Science. Carbon sequestration - research and development. Technical report, U.S. Department of Energy, 1999.

[ONK03] Y.S. Ong, P.B. Nair, and A.J. Keane. Evolutionary optimization of computationally expensive problems via surrogate modeling. *AIAA Journal*, 41(4):687–696, 2003.

[OO04] Jeremy E Oakley and Anthony O'Hagan. Probabilistic sensitivity analysis of complex models: a Bayesian approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 66(3):751–769, 2004.

[PAPDP08] A. Ponsich, C. Azzaro-Pantel, S. Domenech, and L. Pibouleau. Constraint handling strategies in genetic algorithms application to optimal batch plant design. *Chemical Engineering and Processing: Process Intensification*, 47(3):420–434, 2008.

[PBKC02] J-H. Park, H-T. Beum, J-N. Kim, and S-H. Cho. Numerical analysis on the power consumption of the PSA process for recovering CO2 from flue gas. *Industrial & Engineering Chemistry Research*, 41(16):4122–4131, 2002.

[PFKH12] J.M. Parr, A.I.J. Forrester, A.J. Keane, and C.M.E. Holden. Enhancing infill sampling criteria for surrogate-based constrained optimization. *Journal of Computational Methods in Science and Engineering*, 12(1):25–45, 2012.

[PG97] R.H. Perry and D.W. Green. *Perry's Chemical Engineers' Handbook (7th Edition).* McGraw-Hill, 1997.

[PHFK10] J. Parr, C.M.E. Holden, A.I.J. Forrester, and A.J. Keane. Review of efficient surrogate infill sampling criteria with constraint handling. In *2nd International Conference on Engineering Optimization*, pages 1–10, 2010.

[Pis95] E.N. Pistikopoulos. Uncertainty in process design and operations. *Computers & Chemical Engineering*, 19:S553–S563, 1995.

[PLCS06] L. Petzold, S. Li, Y. Cao, and R. Serban. Sensitivity analysis of differential-algebraic equations and partial differential equations. *Computers & Chemical Engineering*, 30(10-12):1553 – 1559, 2006.

[Pow06] M.J.D. Powell. The NEWUOA software for unconstrained optimization without derivatives. In *Large-scale nonlinear optimization*, pages 255–297. Springer, 2006.

[Pow09] M.J.D. Powell. The BOBYQA algorithm for bound constrained optimization without derivatives. 2009.

[RCM04] C.W. Rowley, T. Colonius, and R.M. Murray. Model reduction for compressible flows using POD and Galerkin projection. *Physica D: Nonlinear Phenomena*, 189(1):115–129, 2004.

[RF11] J. Rohmer and E. Foerster. Global sensitivity analysis of large-scale numerical landslide models based on Gaussian-process meta-modeling. *Computers & Geosciences*, 37(7):917–927, 2011.

[RFK93] D.M. Ruthven, S. Faroog, and K.S. Knaebel. *Pressure Swing Adsorption.* Wiley-VCH, 1993.

[RHK11] P. Ranjan, R. Haynes, and R. Karsten. A computationally stable approach to Gaussian process interpolation of deterministic computer simulation data. *Technometrics*, 53(4):366–378, 2011.

[RHR86] N.S. Raghavan, M.M. Hassan, and D.M. Ruthven. Numerical simulation of a PSA system using a pore diffusion model. *Chemical Engineering Science*, 41(11):2787 – 2793, 1986.

[RMER08] S.P. Reynolds, A. Mehrotra, A.D. Ebner, and J.A. Ritter. Heavy reflux PSA cycles for CO_2 recovery from flue gas: Part I. performance evaluation. *Adsorption-Journal of the International Adsorption Society*, 14(2-3):399–413, 2008.

[Rut84] D.M. Ruthven. *Principles of Adsorption and Adsorption Processes*. Wiley-Interscience, 1984.

[RW06] C. Rasmussen and C. Williams. *Gaussian processes for machine learning*. MIT Press, 2006.

[Sch85] J.D. Schaffer. Some experiments in machine learning using vector evaluated genetic algorithms. Technical report, PhD Thesis. Vanderbilt University, 1985.

[Sch98] M. Schonlau. *Computer experiments and global optimization*. PhD Thesis. University of Waterloo, 1998.

[SCS00] A. Saltelli, K. Chan, and E. Scott. *Sensitivity Analysis*. John Wiley & Sons, 2000.

[SD94] N. Srinivas and K. Deb. Muiltiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary computation*, 2(3):221–248, 1994.

[SG07] B. Sankararao and S.K. Gupta. Multi-objective optimization of pressure swing adsorbers for air separation. *Industrial & Engineering Chemistry Research*, 46(11):3751–3765, 2007.

[SGK05] K. Sastry, D. Goldberg, and G. Kendall. Genetic algorithms. In *Search Methodologies*, pages 97–125. Springer, 2005.

[Sir06] S. Sircar. Basic research needs for design of adsorptive gas separation processes. *Industrial & Engineering Chemistry Research*, 45(16):5435–5448, 2006.

[SIW91] O.J. Smith IV and A.W. Westerberg. The optimal design of pressure swing adsorption systems. *Chemical Engineering Science*, 46(12):2967 – 2976, 1991.

[SJ97] K. Singh and J. Jones. Numerical simulation of air separation by piston-driven pressure swing adsorption. *Chemical Engineering Science*, 52(18):3133–3145, 1997.

[Ska60] C.W. Skarstrom. Technical Report U.S. Patent 2,944,627., 1960.

[SLC01] T.W. Simpson, D.K.J. Lin, and W. Chen. Sampling strategies for computer experiments: design and analysis. *International Journal of Reliability and Applications*, 2(3):209–240, 2001.

[SMM03] D. Sarkar and J. M Modak. Optimisation of fed-batch bioreactors using genetic algorithms. *Chemical Engineering Science*, 58(11):2283–2296, 2003.

[Sob93] I.M Sobol'. Sensitivity estimates for nonlinear mathematical models. *Mathematical Modeling & Computational Experiment*, 1(4):407 – 414, 1993.

[Sob01] I.M. Sobol'. Global sensitivity indices for nonlinear mathematical models and their monte carlo estimates. *Mathematics and Computers in Simulation*, 55(1-3):271–280, 2001.

[SSPC00] A. Suppapitnarm, K.A. Seffen, G.T. Parks, and P.J. Clarkson. A simulated annealing algorithm for multiobjective optimization. *Engineering Optimization*, 33(1):59–85, 2000.

[SSW$^+$11] T. Schaul, Y. Sun, D. Wierstra, F. Gomez, and J. Schmidhuber. Curiosity-driven optimization. In *Evolutionary Computation*, pages 1343–1349. IEEE, 2011.

[Ste99] M.L. Stein. *Interpolation of spatial data: some theory for Kriging*. Springer Verlag, 1999.

[Sud08] B. Sudret. Global sensitivity analysis using polynomial chaos expansions. *Reliability Engineering and System Safety*, 93(7):964–979, July 2008.

[SW88] N. Sundaram and P.C. Wankat. Pressure drop effects in the pressurization and blowdown steps of pressure swing adsorption. *Chemical Engineering Science*, 43(1):123–129, 1988.

[SW92] O.J. Smith and A.W. Westerberg. Acceleration of cyclic steady state convergence for pressure swing adsorption models. *Industrial & Engineering Chemistry Research*, 31(6):1569–1573, 1992.

[SWGO00] S. Seo, M. Wallat, T. Graepel, and K. Obermayer. Gaussian process regression: active data selection and test point rejection. In *Proceedings of the International Joint Conference on Neural Networks 2000*, volume 3, pages 241–246. IEEE, 2000.

[SWMW89] J. Sacks, W.J. Welch, T.J. Mitchell, and H.P. Wynn. Design and analysis of computer experiments. *Statistical Science*, 4(4):409–423, 1989.

[Tam12] L. Tamellini. *Polynomial approximation of PDEs with stochastic coefficients*. PhD thesis, PhD Thesis. Politecnico di Milano, 2012.

[TB98] D.J. Ternet and L.T. Biegler. Recent improvements to a multiplier-free reduced hessian successive quadratic programming algorithm. *Computers & Chemical Engineering*, 22(7-8):963–978, 1998.

[TE99] K.G. Teague and T.F. Edgar. Predictive dynamic model of a small pressure swing adsorption air separation unit. *Industrial & Engineering Chemistry Research*, 38(10):3761–3775, 1999.

[TFMW03] R.S. Todd, G.B. Ferraris, D. Manca, and P.A. Webley. Improved ODE integrator and mass transfer approach for simulating a cyclic adsorption process. *Computers & Chemical Engineering*, 27(6):883–899, 2003.

[THW$^+$01] R.S. Todd, J. He, P.A. Webley, C. Beh, S. Wilson, and M.A. Lloyd. Fast finite-volume method for PSA/VSA cycle simulation experimental validation. *Industrial & Engineering Chemistry Research*, 40(14):3217–3224, 2001.

[TK71] P.H. Turnock and R.H. Kadlec. Separation of nitrogen and methane via periodic adsorption. *AIChE Journal*, 17(2):335–342, 1971.

[TW06] R. Todd and P. Webley. Mass-transfer models for rapid pressure swing adsorption simulation. *AIChE Journal*, 52(9):3126–3145, 2006.

[TZ89] A. Torn and A. Zilinskas. *Global optimization*. Springer-Verlag, New York, Inc., 1989.

[VK10] I. Voutchkov and A. Keane. Multi-objective optimization using surrogates. In *Computational Intelligence in Optimization*, pages 155–175. Springer, 2010.

[Vol08] S. Volkwein. Model reduction using proper orthogonal decomposition. *Lecture Notes, Institute of Mathematics and Scientific Computing*, 2008.

[VSBT14] F.A.C. Viana, T.W. Simpson, V. Balabanov, and V. Toropov. Metamodeling in multidisciplinary design optimization: How far have we really come? *AIAA Journal*, pages 1–21, 2014.

[Weg90] E.J. Wegman. Hyperdimensional data analysis using parallel coordinates. *Journal of the American Statistical Association*, 85(411):664–675, 1990.

[WH00] P. A. Webley and J. He. Fast solution-adaptive finite volume method for PSA/VSA cycle simulation; 1 single step simulation. *Computers & Chemical Engineering*, 23(11-12):1701–1712, January 2000.

[WHBH06] L. While, P. Hingston, L. Barone, and S. Huband. A faster algorithm for calculating hypervolume. *Evolutionary Computation*, 10(1):29–38, 2006.

[Wil82] P.H. Wilson. Inverted pressure swing adsorption process, Nov 1982. US Patent 4,359,328.

[WL03] K.S. Walton and D.M. LeVan. Consistency of energy and material balances for bidisperse particles in fixed-bed adsorption and related applications. *Industrial & Engineering Chemistry Research*, 42(26):6938–6948, 2003.

[WS00] W.E. Waldron and S. Sircar. Parametric study of a pressure swing adsorption process. *Adsorption*, 6(2):179–188, 2000.

[WSF03] K. Wang, A. Salhi, and E.S. Fraga. Cluster analysis and visualisation enhanced genetic algorithm. *Computer Aided Chemical Engineering*, 15:642–647, 2003.

[WSF04] K. Wang, A. Salhi, and E.S. Fraga. Process design optimisation using embedded hybrid visualisation and data analysis techniques within a genetic algorithm optimisation framework. *Chemical Engineering and Processing: Process Intensification*, 43(5):657 – 669, 2004.

[XK02] D. Xiu and G.E. Karniadakis. The Wiener-Askey polynomial chaos for stochastic differential equations. *SIAM Journal on Scientific Computing*, 24(2):619–644, 2002.

[XZW$^+$08] P. Xiao, J. Zhang, P. Webley, G. Li, R. Singh, and R. Todd. Capture of CO2 from flue gas streams with zeolite 13x by vacuum-pressure swing adsorption. *Adsorption*, 14(4-5):575–582, 2008.

[YXF$^+$08] H.Q. Yang, Z.H. Xu, M.H. Fan, R. Gupta, R.B. Slimane, A.E. Bland, and I. Wright. Progress in carbon dioxide separation and capture: A review. *Journal of Environmental Sciences-China*, 20(1):14–27, 2008.

[ZT99] E. Zitzler and L. Thiele. Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach. *Evolutionary Computation*, 3(4):257–271, 1999.

[ZWX08] J. Zhang, P.A. Webley, and P. Xiao. Effect of process parameters on power requirements of vacuum swing adsorption technology for CO2 capture from flue gas. *Energy Conversion and Management*, 49(2):346–356, 2008.