UNIVERSITY COLLEGE LONDON

A Multi-criteria Design Framework for the Synthesis of Complex Pressure Swing Adsorption Cycles for CO_2 Capture

by

Giovanna Fiandaca

A Thesis Submitted for the
Degree of Doctor of Philosophy

of the
University College London

Declaration of Authorship

I, Giovanna Fiandaca, declare that this thesis titled, "A Multi-criteria Design Framework for the Synthesis of Complex Pressure Swing Adsorption Cycles for CO_2 Capture" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: ____

Date: ____

UNIVERSITY COLLEGE LONDON

Abstract

Department of Chemical Engineering

Doctor of Philosophy

by Giovanna Fiandaca

Pressure Swing Adsorption (PSA) is the most efficient option for middle scale separation processes. PSA is a cyclic process whose main steps are adsorption, at high pressure, and regeneration of the adsorbent, at low pressure. The design of PSA cycles is still mainly approached experimentally due to the computational challenges posed by the complexity of the simulation and by the need to detect the performance at cyclic steady state (CSS). Automated tools for the design of PSA processes are desirable to allow a better understanding of the the complex relationship between the performance and the design variables. Furthermore, the operation is characterised by trade-offs between conflicting criteria.

A multi-objective flowsheet design framework for complex PSA cycles is presented. A suite of evolutionary procedures, for the generation of alternative PSA configurations has been developed, including simple evolution, simulated annealing as well as a population based procedure. Within this evolutionary procedure the evaluation of each cycle configuration generated requires the solution of a multi-objective optimisation problem which considers the conflicting objectives of recovery and purity. For this embedded optimisation problem a multi-objective genetic algorithm (MOGA), with a targeted fitness function, is used to generate the approximation to the Pareto front. The evaluation of each alternative design makes use of a number of techniques to reduce the computational burden.

The case studies considered include the separation of air for N_2 production, a fast cycle operation which requires a detailed diffusion model, and the separation of CO_2 from flue gases, where complex cycles are needed to achieve a high purity product. The novel design framework is able to determine optimal configurations and operating conditions for PSA for these industrially relevant case studies. The results presented by the design framework can help an engineer to make informed design decisions.

Acknowledgements

Foremost I would like to thank my supervisor, Professor Eric Fraga, for supporting, inspiring and positively challenging me throughout the course of my PhD. I would also like to thank Professor Stefano Brandani, who introduced me to this project and advised me despite the geographical distance.

I would like to thank whole my family, especially my parents, for their immeasurable help, support and belief; thanks to my brother Roberto for his music, words and constant closeness. My special thanks to Issa, for without his love, understanding and encouragement I could have not completed this work.

Many thanks to the people with whom I so closely shared these three years: the colleagues who warmly welcomed me to the Department and made it all easier, especially Max, Andrea, Liban, Sarah, Karolina, Jia, Panos, Marcello and Luca. Thanks to the colleagues who joined me along the way and with whom I shared many chats, thoughts, coffees, beers and laughs: Eria, Alberto, Melanie, Shane, Chandni, Miguel, Olga, the "Chemeng Postgrad Eating Society" and all the other colleagues in the Department.

Thanks to Antonio, Hanie, Kisito, Samba, Anna, Vania, Luc and many other wonderful friends who made this experience so special.

I would also like to express my gratitude to all the staff of the Department of Chemical Engineering, especially Dr. Papageorgiou, Dr. Lettieri and Prof. Simons, for their encouragement and friendship. I would like to gratefully acknowledge the financial support received from the UCL Graduate School.

To my nephew Giovanni,
born while I was writing up this thesis

Contents

List of Figures

List of Tables

Chapter 1

Why Pressure Swing Adsorption

1.1 CO_2 Capture and Storage

Significant changes in the way energy is produced will have to be implemented and many different options will have to be pursued to face global warming. In 1998 a scientific intergovernmental body was set up by the World Meteorological Organisation (WMO) and by the United Nations Environment Programme (UNEP): the Intergovernmental Panel on Climate Change (IPCC). The scope of IPCC is to provide the decision-makers and others interested in climate change with an objective source of information by assessing the latest scientific, technical and socio-economic literature produced worldwide regarding the risk of human-induced climate change, its observed and projected impacts and options for adaptation and mitigation. In the mid-term, suggested actions include the improvement in the fuel efficiency (in power production, transportation and so on), while in the long term the successful strategy should be the use of alternate power sources such as nuclear, solar, renewable energy sources, enhancement of biological sinks [2]. At its 20th Session in 2003 in Paris, France, the IPCC committee agreed on the development of the Special Report on Carbon Dioxide Capture and Storage (CCS). CSS was presented as an option for the mitigation of climate change in the portfolio of actions for the stabilisation of atmospheric greenhouse gas concentrations. Since fossil fuel will remain the main energy source for a number of decades, development of CO_2 capture technology with subsequent storage in suitable geological structures is essential in the short term. According to the IPCC panel estimate by 2050 around 20-40% of global fossil fuel CO_2 emissions could be technically suitable for capture, including 40-60% of emissions from electricity generation and 30-40% of emissions from industry [1].

CCS is a process consisting of the separation of CO_2 from industrial and energy-related sources (e.g. power plants, steel mills, cement kilns and fermentation processes [19]), transport to a storage location and long-term isolation from the atmosphere.

Potential storage methods are [1]:

- storage in geological formations (e.g. oil and gas fields, unminable coal beds and deep saline formations),
- ocean storage (direct release into the ocean water column or onto the deep seafloor),
- industrial fixation of CO_2 into organic carbonates.

CO_2 may be reused for industrial purposes (e.g. in the food industry), but this is not going to contribute significantly to the reduction of CO_2. The capability of storage in geological formations has to be estimated yet, as well as the matches between large CO_2 source points and geological storage formation.

The net reduction of CO_2 emissions achievable by CCS depends many factors: the fraction of CO_2 captured, the increased CO_2 production due to the additional energy required for the capture, the transport and the fraction of CO_2 retained in storage over long term. The estimated result is that a power plant with CCS could reduce CO_2 emissions to the atmosphere by approximately 80-90% compared to a plant without CCS. In the least-cost portfolio of mitigation options, the economic potential of CCS would contribute 15-55% to the cumulative mitigation effort worldwide until 2100; the inclusion of CCS in the mitigation portfolio would reduce the costs of stabilising CO_2 concentrations by 30% or more [1]. CO_2 has already been captured in the oil, gas and chemical industry, as well as from power station flue gases; however, only a small fraction of the carbon dioxide produced is captured because the actual technology available for CO_2 capture would require an equipment 10 times larger than a typical power plant size to achieve a reduction of the emissions of the 75% [15].

1.1.1 Capture Technologies

There are different types of CO_2 capture systems:

- post-combustion (combustion + CO_2 separation),
- pre-combustion (gasification/reform + H_2 and CO_2),
- oxyfuel combustion (O_2 separation from air + combustion),

- industrial separation (industrial process + CO_2 separation).

The above methods are illustrated in Fig. 1.1

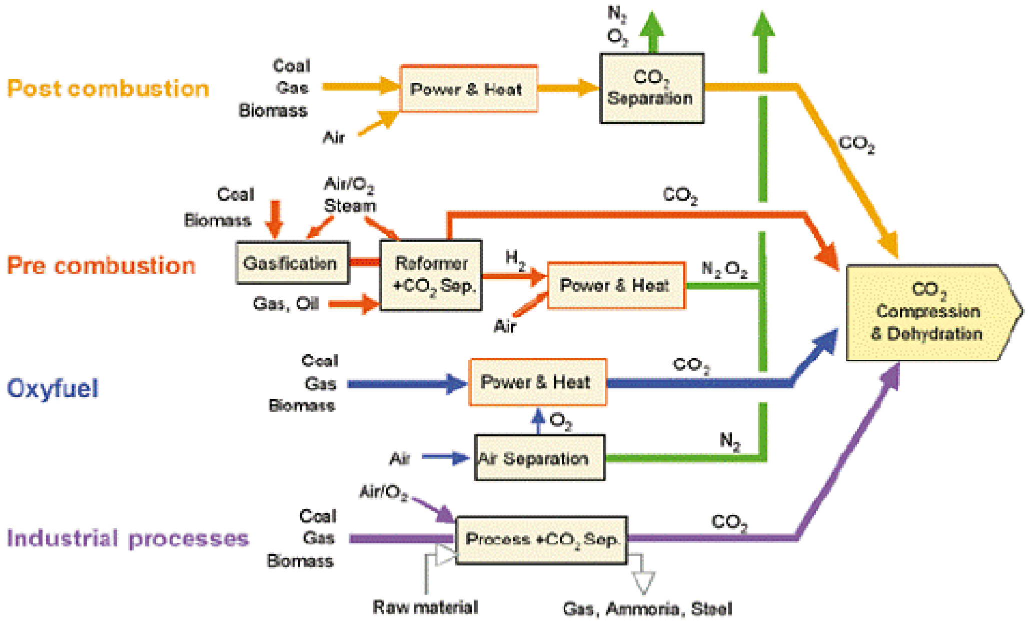


FIGURE 1.1: Overview of CO_2 capture processes [1]

The choice of the most appropriate capture system depends on the concentration of CO_2, the pressure and the temperature of the gas stream and the fuel type. Costs of the various components of a CCS system vary widely, depending on the reference plant and the wide range in CO_2 source, transport and storage situations. However, the relevance of the costs associated to the capture of CO_2 highlights the need of an improvement in capture technologies, aimed at higher energy efficiency. A high performance of the separation is needed since for the disposal of CO_2 to ocean and depleted oil field, it is necessary to concentrate the CO_2 up to 99% to reduce compression and transportation costs [92].

1.2 CO_2 separation: different options

Many different options are available for the separation of CO_2 and all of them can be more or less satisfactory depending on the conditions of the flue gas to be treated. The design of an efficient process for the separation of CO_2 should then be cost-effective in the most typical condition of pressure, temperature and composition of flue gases, as well as it should provide a high purity CO_2 streams, and show a good durability.

The review of the methods presented below follows the scheme proposed by Aaron and Tsouris [2].

Absorption

The first method presented in [2] is absorption by a liquid solvent or a solid matrix. An appropriate solvent must dissolve CO_2, but not O_2, N_2 and other eventual impurities. The solvent needs to be stable to the contact with fly ashes, SO_x and NO_x present in the flue gas. The more the solvent is attracted to CO_2, the higher the loading, but the higher the cost of regeneration of the solvent as well.

One of the most common solvents for this operation is monoethanolamine (MEA), effective at low partial pressure of CO_2, while at high partial pressure of CO_2 solid solvents such as lithium hydroxide and lithium zirconate are preferable.

One of the advantages of absorption is that it is a well known process, both in term of mechanisms involved and of the thermodynamics of the operation. Main drawbacks of the absorption are the high costs of the regeneration of the solvent, corrosion of the steel facilities of the equipment [2, 92] due to high O_2 content in the flue gas, the need of a frequent addition of new solvent and other operating and maintenance costs [2]. Main improvements needed in this area are the development of new solvents and the optimisation of the regeneration step to increase energy efficiency.

Adsorption

While absorption involves the dissolution of the CO_2 in the solvent, adsorption is a heterogeneous process. The adsorption of CO_2 to the adsorbent particles can be manipulated by varying the conditions of pressure and temperature, because higher pressure and lower temperature promote adsorption of the most attracted species to the sorbent particle. This observation lead to the spread of two main methods for adsorption: Temperature Swing Adsorption (TSA) and Pressure Swing Adsorption (PSA). The latter has been proved to be less energy demanding and is capable to show higher regeneration rate (inertia to pressure swing is lower than inertia to temperature swing). One of the advantages of PSA over other separation options is that the PSA process can be operated at high temperatures, so that the flue gas does not need to be cooled before the removal of CO_2 [133]. PSA is attractive since it does not require a separate desorption step that needs heat input (compared to TSA), and because it runs continuously with automatic regeneration of the adsorbent [5].

Another option currently investigated is the electrical swing adsorption [2]: in this method the regeneration of the adsorbent is carried out by passing an electric current through it, which efficiently releases CO_2. However, this technology is still at a preliminary investigation step, and its feasibility has yet to be defined.

Cryogenic separation

The first step of cryogenic separation is the removal of all the components of the flue gas, except for the N_2 and the CO_2. Then, the binary mixture of N_2 and CO_2 is sent to a cryogenic chamber where the CO_2 is liquefied by appropriate manipulation of temperature and pressure. When the triple point of the CO_2 is reached (-56.6 ^0C and ~7.4 atm), CO_2 condenses while N_2 remains in gas phase.

The main advantages of the cryogenic method are that it provides liquid CO_2, ready for the transport in pipelines, and the high purity which can be reached (even over 99.95% [2]). On the other hand, the process is highly energy demanding both because of the effort needed to keep the system refrigerated, and because of the auxiliary step for the separation of the other components prior the refrigeration stage.

Membranes separation

In membrane separations the CO_2 selectively passes through the membrane and an enriched gas phase is obtained.

The main advantage of membranes is the simplicity of the equipment needed. However, the efficiency of the separation is often not satisfactory due to poor selectivity or permeability of the membrane towards CO_2. Another drawback is that membranes do not usually perform well at high temperatures, which are common for flue gases coming from a stack. As in the case of the absorption, also membranes can be degraded by the impurities. Mechanical and chemical stability are both an issue. The results achieved by the current research in the direction of more stable solid and liquid membranes (new metallic, ceramic and alumina membranes) will determine whether this technology can stand alone or can be used within a hybrid separation system.

Alternative options

There are three other technologies currently taken into account for the capture of carbon dioxide, here mentioned briefly since they are still at the stage of laboratory testing. These options are mentioned in the review by Aaron and Tsouris [2].

Hydrate formation and dissociation: this technology is based on the use of hydrates, ice-like structures, in which water forms a cage with cavities where small gases (such as CO_2) can be trapped. Under appropriate conditions, CO_2 might be the better "hydrate former", which means the most likely trapped species.

Redox technology: this option is based on the use of a redox active carrier which binds CO_2 at high pressure (reduction of the carrier), and releases it at low pressure (oxidation).

Ammonium carbonation: ammonium carbonation consists of the reaction between CO_2 and ammonia gas and water vapor in gas phase. The reaction proceeds according to the following scheme:

$$CO_2 + NH_3 + H_2O \rightarrow NH_4HCO_3$$

The product of the reaction is a stable solid, which can subsequently be used in agriculture as a soil fertilizer.

1.2.1 Comparison of separation options

Aaron and Tsouris [2] ranked the different option for CO_2 separation as follows, in order of increasing energy penalty: membrane diffusion, absorption, adsorption and cryogenic distillation. A more detailed investigation is carried out by Ruthven *et al.* [108], which takes into account both capital and operating costs. According to this study, at very large scale (30000-40000 SCFH) and very high product purity, cryogenic distillation is more efficient than all the other options. At medium scale (20000-30000 SCFH) PSA is always the more convenient, while at relatively small scales (2500-200000 SCFH) PSA is in direct competition with membranes. At smaller scales membrane separation is always preferable. This consideration is confirmed by Kostrosky and Wankat [67]. In fact, while for both membranes and adsorption costs scale linearly with the throughput of the operation, for cryogenic distillation costs increase less rapidly. In particular, membrane processes are preferred at very small scales, while PSA is the most economically attractive for relatively large scale operation. Furthermore, PSA requires simple reactors able to undergo relatively small pressure changes, while cryogenic separation units must withstand significant temperature changes, and the equipment of an absorber is exposed to the corrosive solutions formed by the solvent with the flue gas. This evaluation is confirmed in other studies: Biegler *et al.* [13] underlined that using PSA process brings many economical advantages, which justifies the actual efforts of research to optimise the process. For example, PSA processes operate at ambient temperature and require far less capital investment than cryogenic technologies, so that they are better suited to economies of scale and become especially attractive when lower production rates or lower gas purities are required.

Furthermore, recent advances on many aspects of the PSA operation such as configuration of the process, choice of the adsorbent material and operating conditions, lead to an improvement of the performance of PSA. The advantages offered by PSA motivate the research project presented in this thesis, which focuses on the development of automated tools for the design of PSA cycles. An introduction to PSA processes and an analysis of its performance is carried out in the remainder of the chapter to allow a better understanding of the research which has been carried out.

1.3 Basic features of Pressure Swing Adsorption

Pressure swing adsorption (PSA) is a cyclic separation process for gaseous mixtures. It consists of two main operations: during *adsorption*, a mixtures of gases (the *feed*) is passed at high pressure through a fixed bed filled with an adsorbent material. One or more species of the feed are preferentially adsorbed, while the others pass through the bed and are collected as a *raffinate* or *light* product (LP). During *desorption*, or *regeneration*, the pressure is decreased and the adsorbed species are removed from the bed as an *extract* or *heavy* product (HP). During the desorption step the bed is regenerated and prepared for the next cycle.

Usually, a purge stream is used during the desorption to enhance the regeneration of the bed. Hence, the heavy product collected is a mixture of the heavy components of the feed and the purge. On the other hand, the adsorption step is usually terminated before the more strongly adsorbed species break through the bed, to maximise the purity of the raffinate product.

The most known PSA configuration is a 4-step/ 2-bed process, called “Skarstrom cycle”, first proposed by Skarstrom [115]. A schematic of the process is shown in Fig. 1.2. Considering the bottom bed in the figure, the first step is pressurisation with high pressure feed. Then, the light product is withdrawn during the adsorption step. After adsorption, the bed undergoes the blowdown step, during which the bed is depressurised to atmospheric pressure. A fraction of the purified effluent from the second bed is passed through the first bed, countercurrent to the feed direction, to purge the bed at atmospheric pressure. At this point, the bed is ready to start the cycle again.

After a number of cycles, a cyclic steady state (CSS) is achieved at which the condition of the beds at the beginning of the adsorption step and at the end of the desorption step are identical. At CSS the profile of the concentration in the bed oscillate around a mean position.

The choice of the adsorbent is crucial for the separation to be viable. The role of the adsorbent is to provide the appropriate surface area to selectively adsorb the preferentially adsorbed species. However, the heavy component should not be adsorbed too strongly by the adsorbent, otherwise too much energy (high flow rate of the purge/lower pressure levels) would be required to regenerate the adsorbent. Other requirements for the adsorbent is a high capacity, as the higher the capacity the lower the volume of adsorbent needed and, hence, the costs.

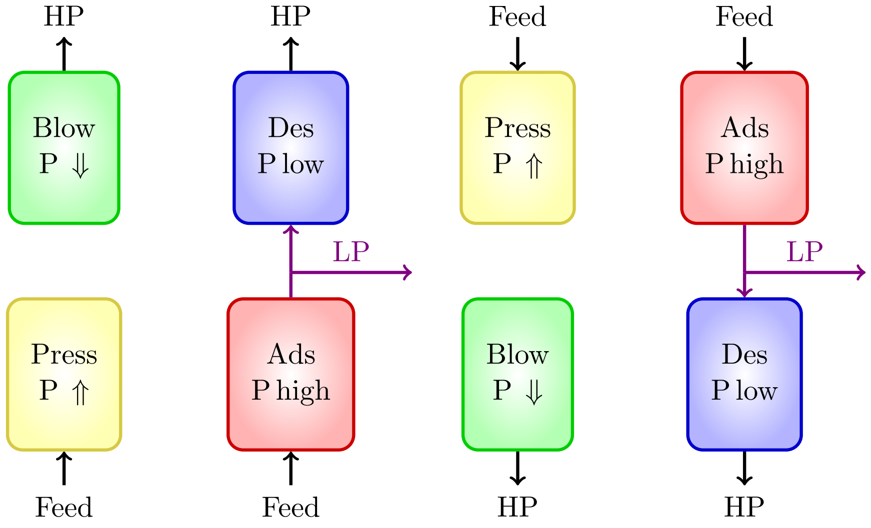


FIGURE 1.2: Full representation of a 2-bed/4-step Skarstrom Cycle.

The Skarstrom cycle has poor performance in many cases: improvements can be made to increase energy efficiency as well as the performance of the separation when a high purity of the heavy product is required (as in the case of CO_2 separation) or when the components to be separated have very similar properties (as in the case of air separation). There are two main approaches to improve the efficiency of the separation: an adsorbent with a higher selectivity can be developed and/or a more effective cycle configuration. In the next section some of the possible modifications to the Skarstrom cycle are discussed.

1.4 Possible improvements to the configuration of a PSA cycle

The PSA systems usually described in literature operate in a cyclic manner undergoing simple cycles (e.g. Skarstrom cycle). This is especially true for studies focusing on the optimisation of the PSA operations, since the effort required by more complicated process configurations implies a greater effort for the optimiser [13, 59]. This results in a gap between real PSA operations and theoretical studies of the simulation and optimisation of PSA.

Improvements to the performance of the Skarstrom cycle can be introduced by adding different steps: one of these is the *pressure equalisation step* [12, 79, 129] which is performed by connecting the beds so that the gas flows from high-pressure bed to low pressure bed. This allows significant saving in overall process energy consumption, since less mechanical energy is required to re-pressurise the bed [78]. Usually the recovery is also increased because less feed gas is necessary to re-pressurise the column. Another option, proposed by Ruthven *et al.* [108], is to augment the cycle with a *rinse step*, during which the bed is purged with feed rich in the adsorbed species in order to improve the purity of extracted product. Two modifications of the Skarstrom cycle have been proposed by Subramanian and Ritter [118] to improve the heavy product purity: the addition of a cocurrent depressurisation step or the addition of a high pressure rinse (i.e., heavy reflux) step. Kostrosky and Wankat [67] proposed a new configuration for PSA cycle whose advantage is providing high recovery maintaining the productivity, when lower purity of the product is acceptable. The productivity of a PSA cycle is defined as the amount of product produced per unit mass of the adsorbent per unit time. The configuration proposed is based on the combination of the conventional PSA with a "cyclic-zone pressure swing adsorption" (CZPSA) cycles. The simple Skarstrom cycle produces high purity product because the purge is done with product. However, recovery is usually low. In [67], the conventional PSA cycle is combined with a cycle operating in a limiting case, the CZPSA, which uses a continuous flow feed material, with adsorption and desorption occurring as a result of pressure change in the feed: it works like a Skarstrom cycle which uses low-pressure feed as purge gas. Such a process reaches lower purity but higher recovery.

Rapid Pressure Swing Adsorption

One particular technology, Rapid Pressure Swing Adsorption (RPSA), has shown recent advancements in process design and operation. It is of interest because the process plant size decreases as cycling becomes more rapid [127]. Cycle time refers to the time taken for the completion of one adsorption-desorption cycle, which in turn decides productivity of the process [10]. Hence, shorter cycle times translates into higher productivity: this lead to the development of RPSA processes [58].

The simple RPSA cycle consists of two steps: pressurisation with feed gas and countercurrent depressurisation with internal purging. The two steps have usually the same duration, normally in the range 1-5 seconds [90]. Fast cycles and a very small size of the adsorbent particles ($200 - 700\,\mu m$ in diameter [90]) characterise RPSA: they cause a steep and periodic variation of the pressure gradients in the bed which leads to an effective self-purging of the bed [90].

Kopaygorodsky *et al.* [66] proposed a new dynamic model for the study of the viability of ultra-rapid pressure swing adsorption (URPSA). The characteristics of URPSA are a small bed-size factor and large product recoveries, which imply improved efficiency of the separation per unit mass of adsorbent material. The bed-size factor is the ratio of the mass of adsorbent required per unit mass of oxygen produced per day. The product recovery achieved with URPSA is much higher than in conventional PSA, and it can achieve comparable oxygen purity with a bed-size factor more than two orders of magnitude smaller. Drawbacks of RPSA are low recovery and increased compressor duty [126].

Dual-Reflux Pressure Swing Adsorption

As mentioned earlier, one of the main drawback of conventional PSA is the low capacity of providing an extracted stream with a high concentration of the heavy (preferentially adsorbed) component. The Dual-Reflux Pressure Swing Adsorption (DR-PSA) is an alternative configuration which allows one to overcome this limit. Dual-reflux cycles [28] are characterised by reflux streams among the beds and sometimes by the introduction of the feed in intermediate points, thus reproducing the stripping and rectifying sections of distillation columns. This process shows to push the enrichment of the heavy component beyond the pressure ratio, which represents the thermodynamic limits of conventional PSA, where only a stripping section is used [28, 51]. A schematic of the dual-reflux configuration proposed by Diagne *et al.* [28] is shown in Fig. 1.3.

Reynolds *et al.* [103, 104] explained the difference between “stripping” PSA and “enriching” PSA: the word “stripping” is used to denote that the feed step is conducted at high pressure and that the adsorbent bed strips the heavy component from the gas phase, because of selective adsorption. We talk about “enriching” PSA when the feed step is performed at low pressure and the adsorbent bed enriches the gas phase of the heavy component due to desorption. Stripping PSA cycles have been used in order to produce a relatively highly pure light component, as in the case of the two-bed/four-step Skarstrom cycle in Fig. 1.2. Stripping cycles have a limitation in concentrating the heavy component in a feed stream, because the light-reflux step uses a part of the

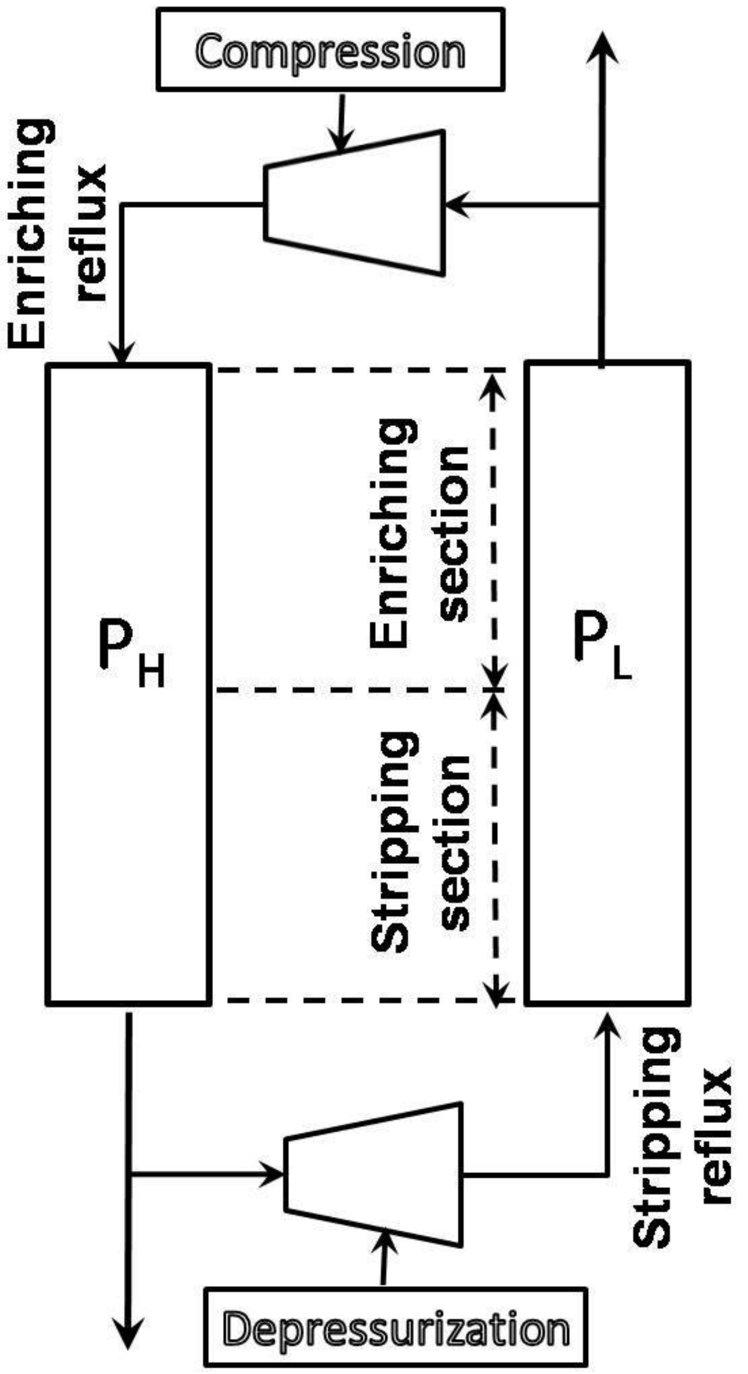


FIGURE 1.3: Conceptual diagram of a DR-PSA cycle with intermediate feed position [28]. In the figure, Q indicates the gas flow rate, R the recycle flow rate, X the molar fraction of a component. The subscript E refers to the enriched gas (HP), L to the lean gas (LP), and F to the feed. P_a is the high pressure during adsorption, while P_d is the low pressure during desorption. This cycle configuration is able to provide an almost pure heavy product ($X_E \to 1$) and a almost purified light product ($X_L \to 1$)

light-product gas for purge, which necessarily dilutes the heavy component in the heavy-product stream: the enrichment of the heavy component is limited by the pressure ratio for a stripping PSA with light reflux [118].

Intyre *et al.* [51] carried out an experimental study on a DR-PSA system, on the effects of intermediate feed position in the column and of the stripping and enriching ratios. Ebner *et al.* [37, 38] used a theoretical approach to DR-PSA. In [37, 38] an idealised analysis of such operation is carried out on the basis of the isothermal equilibrium theory. This ideal system was able to produce both components of a binary mixture at 100% of recovery, over a wide range of conditions (even with a low pressure ratio). Even if the analysis is carried out in a simplified and idealised case, it shows how the DR-PSA process can be a useful tool for the separation of a binary mixture into two relatively pure products, using a low pressure ratio, and thus with a low expenditure of energy.

1.5 CO_2 Capture by Pressure Swing Adsorption

An efficient separation strategy should concentrate the CO_2 up to 99% to reduce the costs of compression and transportation needed for the sequestration which follows the capture of the CO_2. Since CO_2 is usually the more strongly attracted species among the component of the flue gas stream, the simple Skarstrom configuration is not a proper option because it does not provide a high purity of the extracted phase.

This problem has been addressed by researching the most appropriate adsorbent for the operation, and by exploring alternative configurations, such as dual reflux, piston driven rapid pressure swing and combined PSA and TSA systems [52]. A common configuration for CO_2 capture consists of two stages [92]: the first one concentrates CO_2 from 10-15% (usual concentration of CO_2 in flue gases) to 40-60%, while the second stage concentrates it up to 99% purity. This purity would be easily achieved by a one stage PSA if the concentration of CO_2 in the feed were above 25%; unfortunately this is not the case when the feed is a flue gas coming from a coal power plant. Typical composition of flue gases is N_2 and O_2 (75%), CO_2 (15%), H_2O (10%) and minor impurities such as SO_x and NO_x [101]. The latter ones, together with the water, are usually removed before the separation process, so that the fuel gas fed into the PSA unit can be considered as a mixture of N_2, CO_2 and O_2.

Adsorbent

One of the main aspects of an adsorption operation is the choice of the adsorbent. In the case of CO_2 capture, the adsorbent must be selective through CO_2 and show a good stability during the operation over a long period.

Historically, the first adsorbent used for CO_2 capture has been activated carbon [62]. However, successive studies showed that zeolite 13X is more suitable [19]. In fact, even if zeolite 13X shows a higher heat of adsorption than activated carbon, it has higher working capacity, lower purge requirement and higher equilibrium selectivity. These advantages are not affected by the higher temperature developed and, furthermore, the high temperature eases the purge steps by requiring a lower amount of purge [19]. A limit of the zeolites, however, is their affinity towards water: a hydrophilic zeolite adsorbs the water vapor contained in the stack gas, thus reducing the capacity of the zeolite to adsorb the CO_2 [122].

More recently another class of adsorbent has attracted attention as a valuable option for the capture of carbon dioxide: hydrotalcite-like compounds (HTlcs), which are expected to have a high selectivity towards CO_2, high capacity, favorable kinetics, stability to cyclic mechanical stress and showed to work better at high temperatures compared to other adsorbents [32, 102, 132, 133]. HTlcs belong to the group of anionic and basic clays, also known as mixed-metal layered hydroxides and layered double hydroxides, consisting of positively charged layers of metal oxide, or hydroxide, with inner layers of anions, such as carbonate [32]. Equilibria and kinetics of the adsorption of carbon dioxide on such adsorbents are investigated by Diagne and Alpay [32]. As an extension of this work, a cyclic and multibed process for the continuous and efficient recovery of CO_2 using potassium promoted hydrotalcite adsorbent was proposed [33].

Configuration

A multitude of PSA cycles has been developed to produce a highly pure heavy product from feedstocks with low CO_2 concentration.

Kikkinides *et al.* [62] suggested a 4-bed 4-step process with activated carbon which could recover 68% of CO_2 at 99.997% purity. A Pressure and Temperature Swing Adsorption cycle, PTSA, has been proposed by Pugsley *et al.* [96]. The process achieved 90% CO_2 purity together with a 70% recovery. Chue *et al.* [19] demonstrated that zeolite 13X is better than activated carbon for CO_2 recovery on a 3-bed 7-step process. However, they could only achieve low CO_2 recovery due to the lack of a countercurrent step in the cycle. Diagne *et al.* [28–30] proposed an improvement to the process of carbon enrichment and removal by using an intermediate feed inlet in a dual-reflux piston driven system. A model for the operation was proposed [31]. The process is able to enrich simultaneously the weak and the strongly adsorbed species. However, a high power consumption was required.

The benefits of fast cycles were investigated by Suzuki *et al.* [122], where a Piston-Driven Rapid Pressure Swing Adsorption using hydrophobic zeolite was proposed. The process was able to reach a productivity 10 times higher than conventional PSA processes, but the purity of CO_2 in the product stream was low (25 %).

An interesting analysis of the power consumption linked to $\mathrm{CO_2}$ separation from flue gases using PSA was carried out by Park *et al.* [92]. The authors focused on the first stage of a pressure swing adsorption process, where $\mathrm{CO_2}$ is concentrated from 10-15% to 40-60%. This diminished the complexity of the whole problem, which arises since the two stages are coupled: the effluent of the second stage is recycled to the first stage. Furthermore, the first stage is more interesting since it is the more consuming stage. It was found that in this process most of the power is consumed in the vacuum pump and blower. Three different configurations were considered: simple Skarstrom cycle, Skarstrom cycle with pressure equalisation step and Skarstrom cycle with rinse step. The equalisation step allows to gain a higher $\mathrm{CO_2}$ purity without any significant increase of the specific power consumption. The rinse step allows a high purity of the $\mathrm{CO_2}$, but it requires a higher amount of gas to be pumped to produce a given amount of $\mathrm{CO_2}$, thus increasing the specific power consumption for a given recovery.

Choi *et al.* [17] presented the development of an optimal operation system for the PSA process for $\mathrm{CO_2}$ separation and recovery. The problem was addressed both experimentally and theoretically. The study aims at overcoming difficulties in $\mathrm{CO_2}$ recovery when the concentration of $\mathrm{CO_2}$ in the feed is low, as in the case of flue gases. To separate and recover the more strongly adsorbed species ($\mathrm{CO_2}$), a pressure equalisation step and rinse step are added to the Skarstrom cycle. At the rinse step, part of the product flow is recycled to the beds to increase product purity. A MATLAB function based on sequential quadratic programming was used to solve the nonlinear optimisation problem. The authors reported more than 70% $\mathrm{CO_2}$ recovery at more than 90% purity for a modified 3-bed 7-step cycle. However, they solved a small two variable optimisation problem, thus being a specialised case.

Chou *et al.* [18] investigated the performance of Vacuum Swing Adsorption (VSA) for the recovery of $\mathrm{CO_2}$ on zeolite 13X through simulation. A VSA cycle is a PSA cycle where the low pressure achieved during the blowdown step is subatmospheric. The high pressure is just above atmospheric, so that the added cost of the evacuation step is balanced by savings in the pressurisation phase. Evacuation to a very low absolute pressure allows a deeper regeneration of the adsorbent when the isotherm of the adsorbed species is highly favorable. VSA promotes the desorption of the $\mathrm{CO_2}$ in the desorption step, allowing for higher recovery.

Zhang *et al.* [136] have given justifications for using a specific cyclic component step in the adsorption cycle in the context of CO_2 capture by using a simplistic mathematical model for the PSA process. The study investigated the effect of process and operating parameters (feed gas temperature, evacuation pressure and feed concentration) of a CO_2-VSA cycle, especially looking at power consumption. They recalculated the cost of PSA to be around US$ 67/tonne of CO_2 produced compared to that in the IEA report of US$ 97/tonne. This new cost for PSA, as verified through the IEA, compared much more favorably with amine scrubbing technology at US$ 60/tonne. The error in the estimate of the costs was due to the fact that the PSA cycles analysed by the IEA were not designed for heavy component recovery.

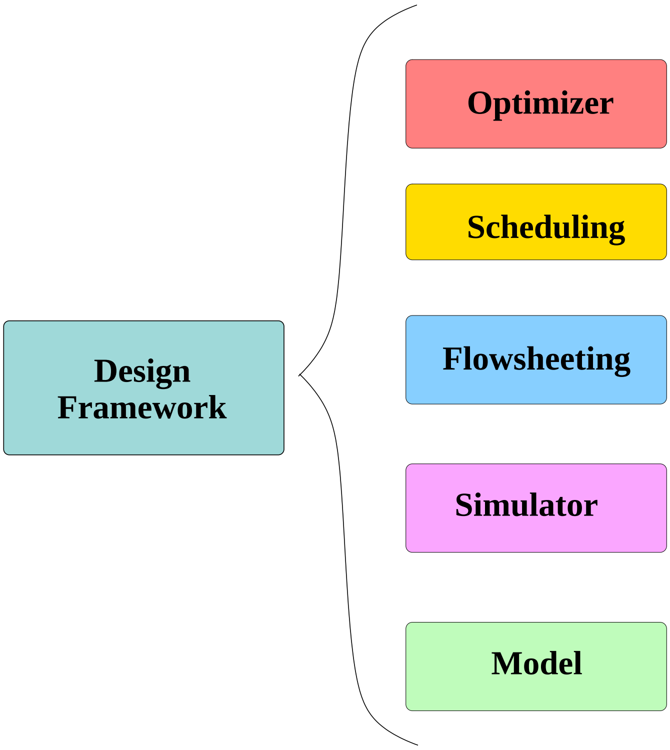
Since 2002, Ritter and co-workers have investigated the effect of PSA cycle configurations for the recovery on the heavy component [35–38, 51, 81, 101, 102, 105]. In particular, they investigated the performance of dual-reflux PSA cycles for the capture of CO_2 from flue gases using HTlcs as adsorbents. In their latest publication [105], nine stripping PSA cycle configurations have been investigated, all with a heavy reflux step, some with a light reflux step, and some with a recovery or feed plus recycle step. The best cycle based on overall performance was a 5-bed 5-step stripping PSA cycle with LR and HR from countercurrent depressurisation: 98.7% CO_2 purity, 98.7% CO_2 recovery. It was shown that any PSA with a heavy reflux step outperformed any PSA process with only light reflux step. The performance of the cycles proposed in this study represents the best trade-offs achieved in literature between purity and recovery of CO_2 so far.

1.6 Aim of the project and structure of the thesis

The development of CO_2 capture technology is necessary to decrease emissions needed to face global warming. Improvement in the efficiency of separation processes needs to be achieved. Pressure Swing Adsorption (PSA) has recently attracted attention as an efficient separation technology for CO_2 capture, since it is the most economic and effective alternative for middle scale operations. PSA has originally been designed to provide a high purity stream of the less strongly adsorbed component. However, CO_2 is the most preferably adsorbed species on the majority of adsorbents used for the operation. For this reason, PSA was initially considered an anti-economical operation as the estimation was made on the basis of PSA cycles designed for the recovery of light components. Advances in adsorbent materials and development of new and more complex cycle configurations have proven PSA to be efficient and economically viable for CO_2 capture [101, 135]. However, complex multi-bed/multi-steps PSA cycles are necessary to obtain high purity CO_2 efficiently. The development of automated tools for the design of complex cycles has been hindered by the complexity of the simulation and the need to evaluate the performance at cyclic steady state. Furthermore, PSA cycles are characterised by conflicting measures of their performance: the lower the costs, the lower the efficacy of the separation; the higher the purity of the desired species in the product, the lower its recovery. Only few previous studies [63, 109] implemented design frameworks taking into account the multi-criteria nature of PSA cycles.

The aim of the project is to establish the feasibility of developing a multi-criteria design framework capable of proposing novel cycle configurations which could outperform the cycles developed so far. As this is an investigatory work, it does not seek to develop definitive design models or high performance search and optimisation algorithms. The methodology at this initial stage is to develop satisfactory approaches that will prove the concept: this in itself moves the field significantly forward as no such framework currently exists. The experimental nature of this project necessitates the investigation of approaches, methods and technologies that have not previously been combined in this field.

A design framework for PSA consists of many necessary elements:



First, the PSA process must be modeled. Different models are available, depending on the characteristics of the adsorption process exploited for the separation (e.g. kinetic or equilibrium driven separation). In general, PSA process are described by a set of coupled PDAEs whose solution is not trivial. An appropriate numerical method and solution scheme must be detected. The model developed must be embedded in a simulation tool to allow the evaluation of the performance of the cycle. One of the most requiring aspect of the simulation is the evaluation of CSS. Again, a viable method to detect CSS should be adopted efficiently and reliably. The simulation tool should be flexible enough to take into account any cycle configuration. A cycle configuration is described by a flowsheet which specifies the number of beds used in the cycle, and the sequence of interconnections which are implemented among the beds during the cycle. Since the performance of the separation depends on the flowsheet of the cycle used, the framework should be able to design the appropriate flowsheet for the separation of interest. Similarly, the time allocated to each step of the cycle, or "schedule" of the cycle, affects the performance of the separation, and should be optimally designed by the framework. The optimisation of the cycle configuration is a highly combinatorial problem conceptually difficult to tackle.

For the framework to propose the optimal cycle configuration (flowsheet and schedule of the cycle) and operating conditions, an appropriate mathematical tool should be included: the optimiser. The choice of the optimiser should be driven by the characteristics of the problem. The optimiser should be able to cope with the cumbersome simulation needed to evaluate the process performance. Furthermore, it should consider the multi-criteria nature of the performance of PSA operations.

The development of each element of the framework poses a challenge. All the elements of the framework interact with each other, and should be developed consistently with each other. In the remainder of the thesis a flowsheet design framework for PSA cycles is proposed. All the elements of the framework have been taken into account.

In Chapter 2, the phenomena governing a PSA operation and their mathematical models are discussed. Chapter 3 deals with the choice and solution of the models for two gas separation processes: separation of air for N_2 production, and CO_2 capture from flue gases. A simplified, yet reliable, simulation tool is developed. The simplifications introduced have been necessary to allow a time effective evaluation of the performance of a PSA cycle.

Each of the case studies poses a specific challenge to the development of the design framework: in the case of N_2 production, a detailed diffusion model needs to be adopted to capture the dynamics of this kinetically controlled separation. On the other hand, CO_2 capture is an equilibrium driven separation which requires the development of PSA cycles capable of providing a high purity of the strongly adsorbed species.

In Chapter 4, the single objective optimisation of a simple 2-bed/4-step Skarstrom cycle is carried out. A preliminary investigation is carried out to characterise the objective function and determine the most appropriate class of optimisers to be used.

The multicriteria nature of the design problem is confronted in Chapter 5. A multi-objective genetic algorithm (MOGA) has been implemented to generate approximations of the Pareto front for both the design problems.

In Chapter 6 a flowsheet multi-objective design framework is introduced, able to synthesise optimal cycle configurations for CO_2 capture via PSA. The performance of three evolutionary algorithms has been investigated. The challenging, and most novel, element of this The framework proposed constitutes a valuable tool to support the design and implementation of new cycles, addressing the needs of the many industries which are currently building new PSA plants.

Chapter 2

Modeling and Simulation of PSA processes

The development of models to simulate PSA processes requires the knowledge of the phenomena that take place in an adsorption bed and their mathematical modelling. Different combinations of adsorbent-sorbate are controlled by different mechanisms (kinetic or equilibrium based separations) and described by different mathematical models. Operating conditions also affect the phenomena involved with adsorption (e.g. contact time between the two phases). An understanding of the relationship between an adsorption system and its mathematical description is crucial to correctly model and simulate its behaviour.

A theoretical background about adsorption processes and their modelling is provided in this chapter. An overview of how PSA models have been solved in literature can be found in section 2.3.

2.1 Principles of adsorption

The knowledge of the basic principles of adsorption is a necessary requirement to a good understanding of PSA operations. The introduction to adsorption provided here is mainly based on the books *Principles of Adsorption and Adsorption Processes* [107] and *Pressure Swing Adsorption* [108] by Ruthven *et al.*

Gas molecules experience a reduction of their potential energy as a result of the interactions with the atoms, or molecules, of a solid. As a result, the gas molecules concentrate near the solid surface, where the density of the gas phase will be higher than in the free-gas phase. The forces linking the gas to the solid depend on the nature of both. We talk about "*physical adsorption*" or "*physisorption*" when the forces linking the two phases are weak, involving van der Waals interaction or electrostatic forces. Physisorption occurs when polar or quadrupolar species are involved. If a real chemical bond takes place, involving transfer of electrons between the adsorbent and the sorbate, the phenomenon is called "*chemisorption*". It is conventionally accepted that if the binding energy per atom or molecule is below 0.5 eV physisorption is taking place; if the binding energy is higher chemisorption is occurring [107]. In chemisorption, the sorbate forms only a monolayer on the solid surface, while in physisorption multiple sorbate layers can form. This implies a higher capacity of physisorption processes, which are hence preferred in practical industrial applications.

Since the strength of the interaction between adsorbent and sorbate depends on both, different substances are adsorbed on the same surface with different affinities. Separation processes based on adsorption exploit this selectivity to separate the components of gas mixtures. A high selectivity of the adsorbent towards the components of the mixture is a primary requirement for the separation process to be economically viable. The **selectivity** in an adsorption process comes from differences in either adsorption equilibrium (equilibrium-controlled separation) or adsorption rate (kinetic-controlled separation) between the components to be separated. In some cases, one of the species is completely excluded from the adsorbent, which acts as a size-selective sieve, and a very efficient separation is achieved. When the adsorption process depends on equilibrium selectivity, the separation factor of a binary gas mixture of components A and B is defined as [107]:

$$\alpha_{AB} = \frac{q_A/q_B}{y_A/y_B}$$

where q_i and y_i are the mole fractions of component $i \in A, B$ in adsorbed and fluid phase at equilibrium, respectively. The separation factor can be thought as the analogue of the relative volatility, which expresses the ease with which the components might be separated by distillation. However, while for a given binary mixture the relative volatility is fixed, the separation factor strongly depends on the adsorbent used for the separation. As a consequence, the choice of the appropriate adsorbent is of primary importance for the design of an adsorption process. The kinetic selectivity is measured by the ratio of the micropore or intracrystalline diffusivities of the two components [107], which will be defined in section 2.1.2.2.

Adsorption processes are invariantly exothermic, as shown by a simple argument proposed by Ruthven [107]: a molecule loses rotational freedom when passing from the free gas phase to the adsorbed phase. This corresponds to a negative entropy change: $\Delta S = (S_{ads} - S_{gas}) < 0$. For adsorption to take place spontaneously, the free energy change ΔG of the process must be negative, $\Delta G = (\Delta H - T\Delta S) < 0$, where ΔH is the enthalpy change. Since $\Delta S < 0$, it is necessary that ΔH is negative, or equivalently that the adsorption is exothermic, to satisfy the condition on ΔG. As all the exothermic processes, adsorption is favoured by low temperatures.

The amount of sorbate in equilibrium with the gas phase increases with the concentration of the solid phase or, equivalently, with the pressure. The relationship between amount adsorbed and pressure is exploited by PSA.

2.1.1 Equilibrium isotherms

The amount of adsorbate of a species at equilibrium with the concentration/pressure of the same species in the gas phase is described by equilibrium isotherms. In absence of mass transfer resistances, the amount of adsorbate would be the quantity at equilibrium with the gas phase. Knowing the equilibrium concentration of the sorbate is necessary to determine the actual adsorbed amount.

At low concentrations, the equilibrium relationship can be expressed by Henry's law (eq. 2.1), which describes a linear relation between the amount adsorbed (q, in $mol\,m^{-3}$) and the concentration of the sorbate in the fluid phase (c, in $mol\,m^{-3}$), or equivalently the pressure (p).

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

where the proportionality constant K or K' is called Henry constant [107].

At higher concentrations, the equilibrium isotherms deviate from linearity. Deviations from linearity are due to the fact that the interactions between free and sorbate molecules, as well as between sorbate-sorbate molecules, become important as the concentration increases. Isotherms have been divided into five categories, according to the classification by S. Brauner, L. S. Deming, W. E. Deming and E. Teller in 1940 [14]. A description of the five types of isotherms is provided by Ruthven [107]: type I isotherms characterise microporous adsorbents. Type II and III describe adsorption on macroporous adsorbents with strong and weak adsorbate-adsorbent interactions, respectively. Types IV and V represent adsorption isotherms with hysteresis. A sixth isotherm has been later introduced by Sing, characteristic of adsorbent with two pore size distributions. The six isotherms, which together constitute the official IUPAC classification of adsorption isotherms, are illustrated in Fig. 2.1.

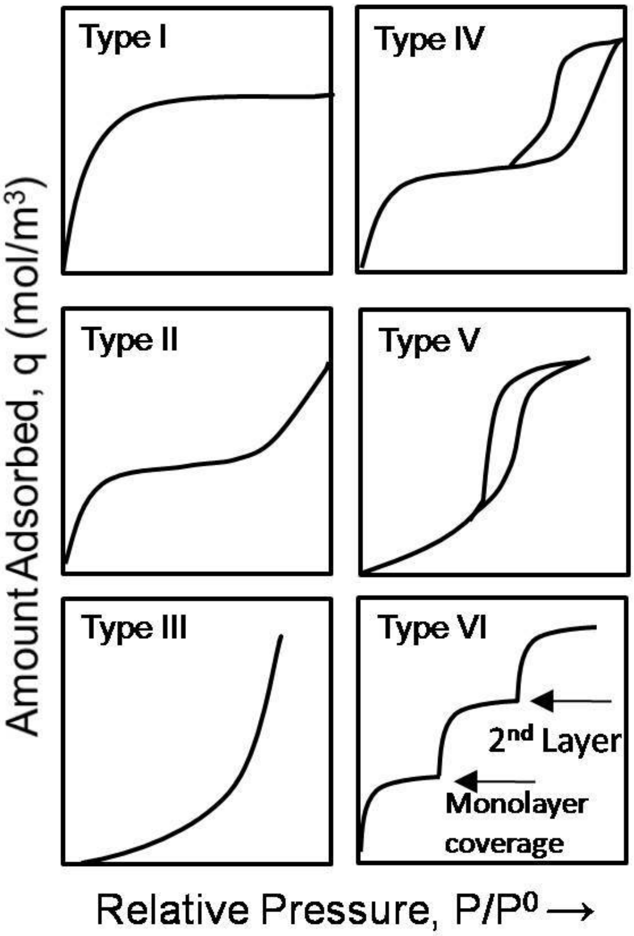


FIGURE 2.1: The IUPAC classification of adsorption isotherms [40].

The simplest model for monolayer adsorption is due to Langmuir :

$$\frac{q}{q_s} = \frac{bc}{1+bc} \tag{2.2}$$

where q_s is the saturation limit and b is an equilibrium constant. According to eq. 2.2, at low concentrations ($bc \lll 1$), the concentration q approaches the value $q_s bc$: $q \rightarrow q_s bc$. This asymptotic behaviour agrees with the linear relationship expressed by Henry's law, and allows to state the following relationship between b and the Henry constant: $K = bq_s$. At high concentrations ($bc \ggg 1$), the value of q reaches a plateau, whose value is given by the saturation limit, q_s. This model generally provides a good representation of most type I isotherms, which are typical of systems where the adsorbent is microporous.

For multicomponent systems, the species in the gaseous phase compete for the adsorption sites. The most used model to describe multicomponent equilibria is the extended

Langmuir model [108]. For a mixture of N components, the extended Langmuir model is expressed by equation 2.3

$$\frac{q_i}{q_s} = \frac{b_i p_i}{1 + \sum_{i=1}^{N} b_i p_i} \tag{2.3}$$

where p_i is the partial pressure of species i in the mixture. Equation 2.3 correctly predicts the behaviour of adsorption systems where the loading is relatively low [108]. Alternative models are described by Ruthven *et al.* [108].

It is often difficult to get reliable equilibrium data for a given system over the pressure range of interest, especially for multicomponent equilibria. In the latter case data are extrapolated from single component equilibrium data.

Reliable models for the description of the equilibrium isotherms are necessary to accurately take into account the deviation of the system from a simple linear equilibrium curve and estimate the correct loading of the adsorbate during a PSA cycle.

Since a PSA cycle usually involves adsorption and desorption steps at the same temperature, we would like the isotherm not to deviate greatly from linearity in order to have a good reversibility: any hysteresis will lead to an unacceptable build-up of the residual concentration in the adsorbed phase. This criterion can be used to choose the appropriate adsorbent to perform the separation of a given mixture.

2.1.2 Mass transfer

A mass transfer between the gaseous and the solid phase is necessary for the equilibrium to be reached. A correct description of the mass transfer occurring between the fluid and the porous adsorbent particles is needed to accurately model an adsorption operation. The comprehension of the mass transfer phenomena that take place in the adsorbent material require an understanding of the structure of the adsorbents themselves.

Two classes of adsorbents can be considered [108]: homogeneous and composites. Homogeneous adsorbents are those whose pore structure persists throughout the entire solid. Since the size of the pores is determined by the chemical structure of the adsorbent, in homogeneous adsorbents the pore size distribution is unimodal. Examples of homogeneous adsorbents are silica gel, activated alumina, activated carbon and homogeneous ion exchange resins.

Composite adsorbents are obtained by the aggregation of small microporous microparticles, held together with the help of a clay binder. Such adsorbents are obtained as composite pellets [107], whose structure is illustrated in Fig. 2.2, together with the three main mass transfer resistance mechanisms offered by the pellet: external film resistance, microporous or intracrystalline resistance (within the microporous crystals) and macroporous or intercrystalline resistance (within the paths described by the binder). Examples of composite adsorbents are carbon molecular sieves, pelleted zeolites and macroreticular ion exchange resins. Microporous adsorbents are usually employed in practical applications, as the higher surface area per unit of volume implies a higher capacity.

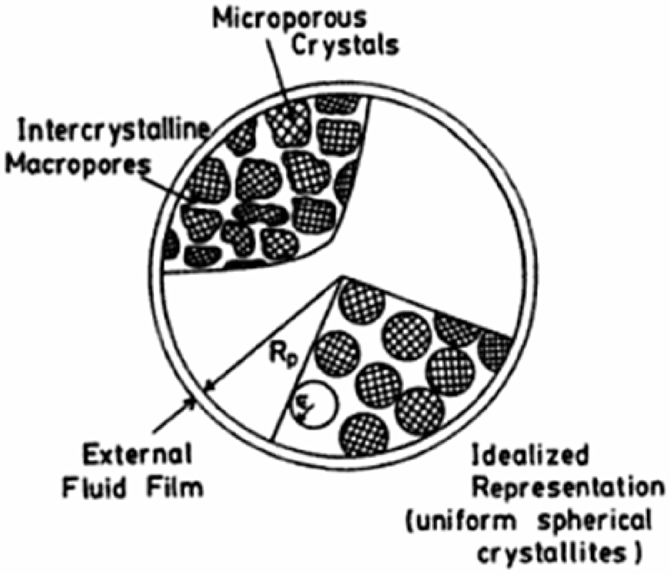


Figure 2.2: Schematic diagram of a composite adsorbent pellet showing the three principal resistances to mass transfer. Figure from [107].

According to the IUPAC classification, three categories of pores exist: micropores (< 20 Å), mesopores (20-500 Å) and macropores (> 500 Å). However, the real distinction between micro- and macro- pores depends on the ratio of the pore diameter to the particle size. When the pore diameter and the dimension of the guest molecule are comparable, the molecule never escapes the force field of the surface of the pore, and it can be considered as adsorbed even if it is at the centre of the pore. In this case, the diffusing molecules jump between adsorption sites, without escaping the force field of the adsorbent surface. This kind of diffusion is called intracrystalline or microporous. On the other hand, a molecule in the centre of a meso- or macro- pore can escape the force field of its walls. In this case, two phases can be distinguished within the pore, an adsorbed phase and a free-gas phase. Due to the small surface area with respect to the pore volume, macropores do not contribute to the overall capacity of the adsorbent, but facilitate the diffusion of the gas molecule to the interior of the adsorbent particle. When diffusion occurs in macrpores the diffusion is called "macroporous".

When the selectivity relies on kinetic control, micropore diffusion is the dominant resistance to mass transfer [108]. In an equilibrium controlled adsorption process the major resistance is usually due to macropore diffusion.

A low resistance to mass transfer is usually desirable. In a composite adsorbent, this can be achieved by reducing the size of the microporous crystals and, hence, minimising the intracrystalline resistance. However, reducing the size of the crystals would imply a decrease of the size of the macroporous path which separate them. The consequence would be a reduction of the macroporous diffusivity which could hinder the diffusion of the gas molecules towards the adsorbent micropores. This drawback could be overcome by reducing the overall size of the pellet particle, but this would bring higher pressure drop. It is then necessary to properly choose both crystal size and pellet size.

2.1.2.1 Micropore Diffusion

As mentioned earlier, micropore diffusion takes place when the dimensions of the diffusing particle and of the pore are comparable, so that the diffusing molecule never escapes the force field of the solid surface [108]. It is an activated process where the activation energy is usually strongly dependent on the relative size of the molecule with respect to the radius of the pore.

The chemical potential of a thermodynamic system is the amount by which the energy of the system would change if an additional particle were introduced, with the entropy and volume held fixed. The true driving force to any diffusive process is the gradient of chemical potential μ:

$$J = -Bc\frac{d\mu}{dr} \tag{2.4}$$

where J, expressed in $mol\, m^{-2}s^{-1}$, is the material flux, B is the mobility coefficient and r is the coordinate along the adsorbent particle radius. The chemical potential is usually expressed as a function of the activity a, as:

$$\mu = \mu^0 + R_g T \ln a \tag{2.5}$$

.

where R_g the universal gas constant and T the temperature. If we consider the system as ideal, the activity corresponds to the partial pressure p. If we substitute this expression of μ in eq. 2.4, and we differentiate taking into account that the partial pressure p and the sorbate concentration q are related by an equilibrium relationship, the expression of the diffusing flux can be expressed as:

$$J = -D\frac{dq}{dr} \tag{2.6}$$

where

$$D = D_0\frac{d\ln p}{d\ln q}; \quad D_0 = BRT \tag{2.7}$$

For systems described by Henry's law, a linear relationship holds between the pressure and the concentration, so that $\frac{d\ln p}{d\ln q} \rightarrow 1$ and $D \simeq D_0$. In this case, the flux is independent of the concentration. For microporous system, isotherms are often described by type I isotherms. This can be mathematically described by the Langmuir isotherm (eq. 2.2). According to this equations

$$\frac{d\ln p}{d\ln q} = \frac{1}{1-q/q_s}; \quad D = \frac{D_0}{1-q/q_s} \tag{2.8}$$

Eq. 2.8 shows that as q approaches the saturation limit q_s, the dependence of the diffusivity on the concentration increases.

It has been experimentally observed that D_0 is independent of the concentration [108], but no theoretical explanation can support this evidence. However, since microporous diffusion is an activated process, D_0 is strongly dependent on the temperature, according to a Arrhenius law:

$$D_0 = D_\infty e^{-E/RT} \tag{2.9}$$

where E is the activation energy and D_∞ is the pre-exponential factor, representing the value of D_0 for $T \rightarrow \infty$. In most microporous adsorbent, such as zeolites and carbon molecular sieves, the energy barrier E is mainly dependent on the repulsive interactions due the molecule experiences to pass though constrictions in the pores [108].

2.1.2.2 Mass transport mechanisms in macropores

The underlying assumption in the analysis of macropores diffusion is that the transport occurs through the pores rather than through the solid [107]. Macroporous transport can be described as a diffusive process, and the Fick's law can be used to model it:

$$J = -D_p \frac{\partial q}{\partial r} \tag{2.10}$$

where J, expressed in $mol\, m^{-2} s^{-1}$, is the material flux, D_p, expressed in $m^2 s^{-1}$, is the diffusivity in the macropores and r is the coordinate along the adsorbent particle radius. In eq. 2.10, the diffusivity D_p is expressed as a function of the concentration q, but not of its gradient $\partial q/\partial r$. However, the diffusivity D_p is independent of q only in thermodynamically ideal systems, i.e. highly diluted system, for which the adsorption equilibrium is described by the Henry's law. When the behaviour deviates from linearity, the dependence of D_p on the concentration must be taken into account.

The geometry of the pores affects the value of the diffusivity: D_p is smaller in real pores than it would be in ideal straight cylindrical pores because of the random orientation of the pores and of the variation in pore diameter. The random orientation of the pores implies a longer diffusion path and a reduced concentration gradient in the direction of the flow [107]. The geometrical effect linked to the porosity of the adsorbent are accounted for by the "tortuosity factor", τ:

$$D_p = \frac{D}{\tau} \tag{2.11}$$

where D is the diffusivity under the same conditions in a straight cylindrical pore. The tortuosity τ is a function of the geometry, depending on the adsorbent and not of the temperature or of the diffusing species. The value of τ is usually determined experimentally [107].

Four transport mechanisms can occur in the macropores of an adsorbent particle [107]: molecular diffusion, Knudsen diffusion, Poiseuille flow and surface diffusion. These mechanisms are discussed below.

Molecular diffusion

Such regime is characterised by the predominance of the collisions molecule-molecule, which results in an exchange of momentum between the particles. For a binary mixture of species A and B, the molecular diffusivity is given by [107]:

$$D_p = \frac{D_m}{\tau} \quad \text{where} \quad D_m = \frac{0.00158 T^{3/2} (1/M_{w,A} + 1/M_{w,B})^{1/2}}{P \sigma_{AB}^2 \Omega(\varepsilon / k_b T)} \tag{2.12}$$

where $M_{w,A}$ and $M_{w,B}$ are the molecular weights, $\sigma_{AB} = \frac{1}{2}(\sigma_A + \sigma_B)$ is the collision parameter from the Lennard-Jones potential, and Ω is a function of ϵ/k_bT where $\epsilon = \sqrt{\epsilon_A\epsilon_B}$ is the Lennard-Jones force constant and k_b is the Boltzmann constant [107]. Molecular diffusion regime occurs at high concentration and in large pores.

Knudsen diffusion

The transport mechanism described by Knudsen diffusion is characterised by the preponderance of molecular collisions against the pore walls. This regime is likely to occur at low concentration and in small diameter pores: in such condition the mean free path of diffusing molecules is smaller than the pore diameter, so that collisions molecule-pore wall occur more frequently than collision molecule-molecule [107]. Knudsen diffusivity can be expressed by

$$D_k = 9700\,\mathfrak{r}\left(\frac{T}{M_w}\right)^{1/2} \tag{2.13}$$

According to eq. 2.13, D_k is proportional to the mean radius of the pores, $\mathfrak{r}$, and to the square of the ratio between temperature T and molecular weight M_w of the diffusing species. Knudsen diffusivity is independent of the concentration, and each species diffuses independently of the other.

Intermediate regime between molecular and Knudsen diffusion

At a given pressure, it is likely that both molecular diffusion (in macropores) and Knudsen diffusion (in mesopores) take place. Since the mean free path of the molecule strongly depends on pressure, for any couple adsorbent/adsorbate there will be a transition from the predominance of molecular diffusion (at high pressure) to predominance of Knudsen diffusion (at low pressure). The combination of this two regimes can be described by two different models, one based on the Fick's law, and the other based on the Stefan-Maxwell relation, from which the dusty-gas model is derived:

- Model based on Fick's law: the driving force to mass transfer is supposed to be the species concentration gradient and the characteristic parameter is the effective pore diffusion coefficient, D_p;

According to this model, for the i-th species in the mixture, the flux J_i can be expressed as:

$$J_i = -D_p \frac{\partial q_i}{\partial r}; \tag{2.14}$$

$$\frac{1}{D_{p,i}} = \frac{1}{D_{k,i}} + \frac{1}{D_{m,i}} \tag{2.15}$$

- Dusty-gas model: it directly derives from the application of the Stefan-Maxwell relations to the diffusive transport in the pores. The model considers the pore wall as a pseudo-species in the mixture, whose concentration is uniformly distributed in the space, and whose molecular weight is infinite. According to the model, the diffusive fluxes of each species, J_i, in a mixture of N components, must obey the following relation [23]:

$$\frac{\partial q_i}{\partial r} = -\left(\sum_{j=1, j\neq i}^{N} \frac{y_j J_i - y_i J_j}{D_{i,j}} + \frac{J_i}{D_{k,i}} \right) \tag{2.16}$$

where y_i) is the molar fraction of $i - th$ species.

Poiseuille flow

The Poiseuille flow is due to the difference in total pressure across the adsorbent particle, resulting in a laminar flow through macropores which contributed to the overall adsorption flux [107]. This effect is usually negligible in large packed beds, as the pressure drop across the particle is small, but it might become significant in small laboratory rigs [107]. The equivalent Poiseuille diffusivity is expressed as [108]:

$$D_{\mathrm{P}} = Pr^2/8\mu \tag{2.17}$$

where P is the pressure, r is the radius of the pore and μ is the viscosity. The contribution from the Poiseuille flow is directly addictive to the combined diffusivity given by the molecular and Knudsen mechanisms (eq. 2.14).

Surface Diffusion

The surface diffusion is linked to the mobility of a species in the adsorbed phase present at the pore wall. It occurs when the diffusing molecule escapes from the force field of the adsorbent surface. The main interactions are sorbate-adsorbate. Even if the mobility of the diffusing species is much smaller than in the gas phase, its concentration can be much higher so that the contribution of the surface flow can be important if the thickness of the adsorbate layer is significant. This is more likely to happen at low temperature, where adsorption is promoted. It is an "activated process, characterised by a succession of jumps between adjacent equilibrium sites" [23, 107]. The surface diffusion flux is due to a gradient of chemical potential of the adsorbed species, and it can be approached, once again, either using the Fick's law, or on the basis of the Stefan-Maxwell relation.

2.2 Modeling PSA operation

Modelling the operation of a bed undergoing adsorption requires the following balances and equations to be carried out:

- a gas phase mass balance for each component;
- an energy balance for the gas, the solid and the adsorbed phase;
- an equation relating pressure gradient and velocity of the fluid phase;
- an equation to describe the rate of transfer for each component between the gas and the adsorbed phase;
- a gas-solid equilibrium relationship for each component, along with an equation of state for the gas phase.

These equations give birth to a coupled set of partial differential and algebraic equations (PDAEs) with associated initial and boundary conditions.

The models used to describe PSA processes in literature differ under different aspects. According with the specifics of the system taken into account, the following choices can be made:

1. the fluid flow pattern (generally plug flow or axially dispersed plug flow);
2. constant or variable fluid velocity;
3. the form of the equilibrium relationships;
4. the form of the kinetic rate expressions;
5. the inclusion of heat effects (isothermal/nonisothermal process).

In the following sections some details of the above balances are provided.

2.2.1 Mass balance of the gas phase

The mass balance for the i-th component over a differential volume element of an adsorption column is [94, 107]:

$$-D_L \frac{\partial}{\partial z}\left(c\frac{\partial y_i}{\partial z}\right) + \frac{\partial}{\partial z}(cv_i) + \frac{\partial c_i}{\partial t} + \frac{(1-\varepsilon)}{\varepsilon}\frac{\partial \overline{q_i}}{\partial t} = 0 \tag{2.18}$$

where

- $\overline{q_i}(t)$, mol/m^3, average solid concentration;
- v_i, m/s, interstitial gas velocity;
- ε, bed voidage;
- D_L, m/s axial dispersion coefficient;

The first term of eq. 2.18 takes into account the axial dispersion, and D_L is a lumped coefficient which takes into account all the mechanisms of axial mixing. The second term is the convective term. The third and forth terms describe the build up in the fluid and solid phase, respectively.

Generally radial dispersion is not so important to be taken into account within the model. The effect of axial dispersion may be neglected when it is small compared to the effect of the mass transfer resistance, and the plug flow model can be used (it is obtained from eq. 2.18 by dropping the term of axial dispersion $-D_L \frac{\partial}{\partial z}\left(c\frac{\partial y_i}{\partial z}\right)$). When we want to solve numerically the equations, anyway, the dispersed flow model results advantageous since the inclusion of the dispersive terms eliminates the discontinuities in the slope of the concentration profile.

A further assumption can be added when modeling trace systems, where the adsorbable species is at low concentration in an excess of inert carrier (for example in purification processes such as air drying and hydrogen purification): the change in the gas velocity through the bed due to adsorption/desorption can be neglected (e.g. $\frac{\partial v_i}{\partial z} = 0$).

The opposite case occurs when the mole fraction of the adsorbable component in the feed is large, and we can no longer consider the velocity to be constant along the bed. We are then in the case of the *bulk separation* which requires a detailed analysis of the pattern of the velocity through the adsorbent bed [108].

2.2.2 Mass balance in the particle

For the evaluation of the build up in the solid phase, $\partial\overline{q_i}/\partial t$, it is necessary to carry out a balance around the solid particle. Cruz *et al.* [23] carried out this balance, assuming local equilibrium between fluid and adsorbed phase: the rate of change in the gas and in the adsorbate phase equals the sum of the fluxes in the pores and in the surface. Under certain assumptions, the the intra-particle mass balance, in terms of solid concentration, is [23]:

$$\frac{\partial q_i}{\partial t} = \frac{1}{r^s}\frac{\partial}{\partial r}\left(r^s D^e_{M,i}\frac{\partial q_i}{\partial r}\right) = \frac{1}{r^s}\frac{\partial}{\partial r}\left(r^s J_i\right) \tag{2.19}$$

where s is the particle geometry factor, which is 0 for a plane sheet, 1 for a cylinder and 2 for a sphere. $D^e_{M,i}$ is the effective homogeneous diffusion coefficient, being equation 2.19 referred to as the *homogeneous diffusion equation.* Depending on the transport mechanisms taking place, different models of the material flux J_i need to be used, as discussed in section 2.1.2. Eq. 2.19 applies when the isotherm is linear and the viscous (Poiseuille) flux is negligible. These assumptions imply some limitations of this model when simulating cyclic adsorption processes, since it is valid for long time cycles and low total pressure [23].

For a more detailed description of the solid mass balance, Fickian and dusty-gas model could be adopted to express the flux in the pores. The two models are equivalent in two situations:

1. in dilute systems, where all species but the solvent have low concentration ;
2. when Knudsen diffusion is predominant, i.e. when the pore diameter is lower than the mean free path of the diffusing molecules.

An extended comparison between the Fickian and the Maxwell-Stefan approach is carried out in [68]. Here, it is underlined the potential generality of the Maxwell-Stefan approach, which is able to correctly take into account thermodynamic non-idealities and the influence of external forces (such as electrostatic and centrifugal fields).

2.2.2.1 The linear driving force approximation

The most common simplification of the homogeneous diffusion equation (eq. 2.19) is the linear driving force (LDF) approximation, which allows to significantly simplify the bulky computations involved in the solution of the adsorption processes. According to the LDF approximation, the PDE describing the mass balance in the particle (eq. 2.19) is replaced by a simpler ordinary differential equation, which states that the uptake rate of a species is proportional to the difference between the surface concentration (q^*) and the average concentration ($\overline{q}$) within the particle.

In the LDF model the mass transfer rate is expressed as [108]:

$$\frac{d\overline{q}}{dt} = k(q^* - \overline{q}) \qquad \text{LDF Approximation} \tag{2.20}$$

where k is the overall LDF mass transfer coefficient based on adsorbent phase concentration. The average concentration in the particle is given by:

$$\overline{q} = \frac{\int 4\pi r^2 q dr}{\frac{4}{3}\pi R_p^3} \tag{2.21}$$

The general expression of k for micropore control is [107]

$$k = \Omega \frac{D}{R_P^2} \tag{2.22}$$

where D is the diffusivity, R_p is is the radius of the particle and Ω is a parameter which has been expressed in different ways, according to the operation taken into account.

It can be demonstrated that between the diffusion equation and the LDF approximation, there is a substantial equivalence when the concentration profile within the particle can be considered parabolic. The demonstration is carried out below.

According to the diffusion equation 2.19, the average uptake rate of the solid phase in the assumption of spherical adsorbent particle and constant D is

$$\frac{d\overline{q}}{dt} = \frac{3}{R_p} D \left(\frac{\partial q}{\partial r}\right)_{R_p} \qquad \text{Diffusion equation} \tag{2.23}$$

where the solid concentration $q(t, r)$ is a function of time and space, and R_p is the radius of the particle. If we assume a parabolic concentration profile within the particle, $q(r) = a + cr + br^2$, the coefficient c needs to have a 0 value to satisfy the “symmetry condition”: $(dq/dr)_{r=0} = 0$. This condition implies the symmetry of the concentration profile with respect to the center of the particle (r=0), or in other words a zero-flux at the center of the particle. This holds because of the symmetry of the flux of matter from all the directions towards (in the case of adsorption) and from (in the case of desorption) the center of the particle. Accordingly, the term $\left(\frac{\partial q}{\partial r}\right)_{R_p}$ of eq. 2.23 is expressed by:

$$\left(\frac{\partial q}{\partial r}\right)_{R_p} = 2bR_p \tag{2.24}$$

Substituting eq. 2.24 into eq. 2.23, we obtain:

$$\frac{d\overline{q}}{dt} = 6Db \tag{2.25}$$

which represent the diffusion equation according to the assumptions made.

According to the definition of average concentration in the particle (eq. 2.21), assuming the parabolic concentration profile, the expression of the average concentration is:

$$\overline{q} = a + \frac{3}{5}bR_p^2 \tag{2.26}$$

The concentration at the surface is given by $q^* = a + bR_p^2$. Therefore, if we substitute eq. 2.25 and eq. 2.26 into eq. 2.20, we find the following equivalence criterion between the LDF and the diffusion model, in the system of assumptions made:

$$k = 15\frac{D}{R_p^2} \tag{2.27}$$

Comparing eq. 2.27 and 2.22, we obtain that the value of Ω that satisfies the equivalence between diffusion and LDF criteria is 15. The same value of Ω was indicated in the earlier formulation of the LDF approximation by Glueckauf [41] both for micropore and macropore diffusion. Successively Nakao and Suzuki [87] showed the dependency of Ω on the frequency of adsorption and desorption steps by comparing the diffusion and LDF model for a single particle subject to cyclic adsorption/desorption steps, and suggested a correlation to calculate the LDF constant for any given cycle time.

Alpay and Scott [7] used the penetration theory to find the best equivalence criterion between diffusion model and LDF approximation. According to the penetration theory [7], the adsorbent particle is large enough that the concentration at the center of the particle is constant since it is not affected by the boundary conditions at the surface, even when these are changing periodically (as in a PSA process). The results they got by comparison between the LDF and the diffusion expression yielded $\Omega = 5.14/\theta_c$, which is close to the expression by Nakao and Suzuki [87] over the range $10^{-3} < \theta_c < 10^{-1}$, where θ_c is the dimensionless adsorption/desorption time, expressed as $\theta_c = Dt/R^2$.

Studying the case of a heatless dryer, Raghavan *et al.* [97] confirmed the dependence of Ω on the cycle time; however they tested their equivalence criteria between pore diffusion and LDF not on a single particle, as in [87], but on a packed beds, finding a slightly different correlation. The same result has been showed in [127]: the linear driving force model and a rigorous pellet model (the viscous flow plus dusty gas model intrapellet flux equation) were compared for the simulation of a rapid pressure swing adsorption operation. The simplified LDF models appears to fail, while the detailed one maintains a good agreement with experimental results. The minimum value of the dimensionless cycle time below which LDF model fails is $\theta_c = 1.0$, thus suggesting that conclusions from studies from a single pellets [8] may not apply rigorously for to packed beds.

A limitation of the LDF model is in the ability to fully describe the system when the dependency of the diffusivity on the sorbate concentration is pronounced [108]: the LDF approximation does not usually have limitations in the description of equilibrium controlled separations (macropores diffusion occurs), while problems arise in kinetically controlled systems. Such dependency of the diffusivity on the concentration is observed in the main microporous adsorbent such as zeolites and carbon molecular sieves. The effect is more pronounced in binary systems, being the binary diffusion more sensitive to the concentration profiles within the adsorbent particle.

2.2.3 Energy Balance

The energy balance is usually evaluated as the rate of change of the internal energy of a differential unit of volume with respect to time.

The conservation of energy is retained within the model formulation to account for temperature variations due to the adsorption of a bulk component from the gas stream. Non-isothermal effects become important when temperature variations from adsorption/desorption affect the loading and, hence, the performance of the separation. The temperature effects may be ignored in case of adsorption of a trace component that is weakly adsorbed [127]. Entropy and enthalpy changes can be calculated, allowing the evaluation of non-reversibility and energy losses, which can be used to characterize the efficiency of PSA cycles [11].

2.2.4 Momentum Balance

The momentum balance relates the system pressure gradient to the gas velocity. Analysis for differential momentum balance shows that the time constant for this equation is much smaller than the time constants for either mass or heat transfer, so that Biegler *et al.* [13] chose the steady state Ergun equation (eq. 2.28).

$$-\frac{\partial P}{\partial z} = 150\,\frac{\mu v(1-\varepsilon)^2}{d_p^2\,\varepsilon^3} + 1.75\frac{\rho M_w v^2(1-\varepsilon)}{d_p\varepsilon^3} \tag{2.28}$$

where d_p is the particle diameter, ρ the gas density and μ its viscosity. Another option widely used to model momentum balance is the simplified version of the above, Darcy's law [9, 90]. Darcy's law is given by eq. 2.29

$$-\frac{\partial P}{\partial z} = 150\,\frac{K_1\mu(1-\varepsilon)^2}{d_P^2\varepsilon^3}(\varepsilon v) \tag{2.29}$$

where K_1 is a correction factor determined experimentally [9]. Often the pressure drop within the bed is considered negligible [16, 21, 62].

2.3 Simulation strategies in literature

The main difficulties in the simulation of adsorption processes are the high CPU time requirements posed by the simulation and the lack of flowsheet flexibility to simulate different process configurations [69]. The most common models for the simulation of PSA cycles are able to describe the experimental trends of the operations, but fail to describe actual industrial processes because they consider limited process configurations [69]. The usual drawback of sophisticated simulation is the computational effort required, which might make the simulation until cyclic steady state slower than the real time of the operation. These limits translate into the need of relying on experimental data for the development of new processes, while the prospect of building large and expensive experimental units for the development of new ideas is unattractive [69].

Even after many years, these points remain the bottleneck towards the development of an efficient design tool for adsorption processes. Since the development of an automated flowsheet design framework for PSA configurations is the goal of this project, it is necessary to develop a simplified and reliable simulation tool, which could allow the reduction of the computational time and the simulation multi-bed/multi-step cycle configurations. A review of simulation strategies proposed in literature is presented in this section, while the simulation tools developed in the present study are presented later.

Biegler *et al.* [13] detected two different approaches used in literature to solve the system of PDAEs involved in the simulation of PSA operations: complete discretisation and the method of lines.

A complete discretisation consists of a simultaneous discretisation in both space and time domain of all the PDAEs. The resulting set of nonlinear algebraic equations can be solved by large-scale Newton-based equation solvers. This method has been used by Nilchan and Pantelides [90]: the numerical discretisation of the set of PDAEs is directly discretised by gPROMS and reduced to a set of DAEs. The method of orthogonal collocation on finite elements has been used for the spatial discretisation, while the second order finite difference method has been used for the temporal discretisation.

Complete discretisation is efficient for simple models, and allows a straightforward addition of the cyclic steady state (CSS) conditions to the discretised bed models, thus leading to simultaneous convergence of the entire cycle. However, failure of the Newton solver may occur in face of steep fronts that characterize PSA separation: the PDAEs which constitutes the bed model exhibit steep fronts if mass transfer is rapid and the equilibrium is favorable. Furthermore, for complicated models error accumulation caused by complete discretisation can also lead to convergence failure [64].

The method of lines (MOL) is a two step approach: the PDEs are first discretised in space to form a system of ODEs or DAEs, which are then integrated by standard time integration routines [13].

Different methods have been used in literature to get a spatial discretisation of the PDEs:

- centered finite difference method [64]
- Galerkin finite difference method [124]
- orthogonal collocation [23, 98, 99]
- finite volume method [25, 54, 130]

Orthogonal collocation has firstly been proposed by Ravaghan and Ruthven [98], and it is particularly effective for boundary value problems and requires less computer time at a fixed accuracy of solution than standard finite difference method. The method has been indicated as the most effective also in the comparative study by Cruz *et al.* [23].

A finite volume method for the solution of PSA/VSA was proposed in [130], with suitable higher order interpolation schemes, able to accommodate boundary conditions for a variety of steps occurring in a PSA/VSA process.

2.3.1 Simulation of fast cycles

When simulating fast cyclic adsorption processes, as the cycle time is reduced in comparison to the characteristic diffusional time constant, it is necessary to increase the number of spatial discretisation points in order to use the Fickian diffusion model. In fact, when the diffusional time constant R^2/D is large with respect to the cycle time, the internal concentration profile in the outer region of the adsorbing particle varies rapidly [6]. A discretisation scheme which uses a equally spaced elements has to be tested iteratively to establish the minimum number of intervals needed to avoid numerical dispersion of the adsorbent profile [106]. When applying the LDF approximation to shorten simulation time, a cycle time dependent correction is necessary [87]. The LDF equivalence correctly predicts the increase in the mass transfer rates that is consistent with the increased concentration gradients and the amplitude ratios of the concentration; yet, the LDF fails to describe the correct phase lag between the input signal and the output signal of an adsorption process subject to a cyclic perturbation. Rouse and Brandani [106] proposed a method to correctly predict cyclic steady state concentration profiles, which requires to modify the volume of solid taken into account in order to consider only the active section of the adsorbent particle. Introducing two corrections (the time dependence of the mass transfer constant on time and the correction of the active volume) it is then possible to predict correctly both the amplitude ratio and the phase lag. This method allows a correct description of the cyclic steady state but fails to describe the transient behaviour of the operation. The transient behaviour is not described since the model considers a reduced capacity of the system to take into account that at CSS the internal part of the solid adsorbent is unaffected by the varying external concentration. Therefore the detailed diffusion model is necessary to fully describe the transient behaviour.

Ahn and Brandani [6] proposed a new model for fast cycles was based on the definition of two different regions within the adsorbent particle: in the outer region concentration vary significantly with large internal gradients, leading to enhanced mass fluxes, while in the internal region the concentration profile is virtually flat. This description is in agreement with the penetration theory mentioned in section 2.2.2.1. This turns into a numerical grid which has a constant number of elements, independent of the process cycle time. This method was based on the observation that the LDF approximation is still valid when the particle is described as two separate regions, so it is possible to apply a similar approach to the numerical solution of the diffusion equation for fast cyclic processes. In [6] a predictive method to assign a numerical grid for the solution of the diffusion equation for fast cyclic adsorption processes that is scaled so that the number of collocation intervals is independent of the cycle time. The procedure is demonstrated on a model for the simulation of a heatless dryer pressure swing adsorption process.

2.4 Summary

The key aspects of the simulation of PSA processes have been addressed. A particular attention has been paid to the mass transfer term (see sections 2.1.2 and 2.1.2.2). Usually mass transfer is described by the homogeneous diffusion equation, eq. 2.19, which corresponds to a mass balance in the particle according to the Fickian model for the diffusion. This equation is often simplified in the linear driving force (LDF) model. The simplified LDF model fails to describe the adsorption process in case of fast PSA cycles and of kinetically controlled separations.

The theoretical background about PSA processes provided in this chapter provides a foundation to introduce the modelling and simulation of the case studies chosen, presented in the next chapter.

Chapter 3

Modeling and Simulation of Case Studies

A necessary step into the development of an optimisation framework for the design of PSA cycles is the development of simulation tools to evaluate the performance of the cycles at different design points. In this chapter the modeling and simulation of two PSA processes are presented. The two processes of interest are the separation of air for N_2 production and the capture of CO_2 from flue gases.

The production of N_2 from air is an interesting case study because the separation is kinetically controlled and, hence, it requires the solution of a detailed diffusion model, as opposed to the simplified linear driving force (LDF) model, to describe the mass transfer phenomena involved. Given the increased computational burden required by the diffusion model, kinetically controlled separations have seldom been investigated in the literature [90, 109], especially within optimisation frameworks where multiple evaluations of the model are required.

The modelling and simulation of the CO_2 case are presented in section 3.2. CO_2 separation is a relevant design problem not only for the environmental relevance discussed earlier: CO_2 is the most strongly adsorbed species on the majority of adsorbents known, hence the operation is characterised by steep adsorption waves which make the numerical simulation of the process difficult to solve; furthermore, complex cycles are needed to meet the high level of CO_2 purity required.

3.1 Production of N_2 from air: modelling of a kinetically controlled separation

The problem of interest is the separation of air on activated carbon sieve for N_2 production. This separation is kinetically controlled by microporous diffusion. Fast cycles are necessary to exploit the different adsorption rates of the components and produce a purified N_2 stream: since O_2 diffuses faster than N_2 within the pores of the adsorbent, during the adsorption step the O_2 remains trapped in the bed, while the N_2 is collected as a refined stream at the product end. However, since the equilibrium isotherms of the two components on activated carbon are very similar, if the adsorption step lasts long enough for the equilibrium to be reached, as much N_2 as much O_2 would be adsorbed and no separation would be achieved.

A simplified model of the operation has been developed to reduce the computational costs involved with the simulation. The aim is to have a simulation which can be embedded within a design framework with ease, which means that a manageable computational time is required. The aim of the simulations is to correctly detect the relation between the design variables and the performance, so that the optimiser can distinguish the performance at different design points in a time effective way.

The main assumptions used in the model are perfect mixing, ideal gas phase, spherical adsorbent particles and negligible pressure drop, isothermal operation. These assumptions are reasonable for air separation on a carbon molecular sieve [44, 45].

A series of CSTRs (continuous stirred tank reactors) has been used to simulate each bed. A CSTR is an ideal continuous reactor whose content is continuously stirred and, hence, homogeneous. As a consequence, the product of a CSTR has the same composition as the fluid inside the reactor. A more realistic model for an adsorption column is the dispersed plug flow reactor, where the composition of the fluid changes from one point to another. This more complex behaviour can be modelled using a series of CSTRs [74]. The model for a single CSTR has been developed; then, the output of each CSTR has been used as the input to the next, using linear interpolation.

This simplification reduces the computational requirements since the mass balance of the fluid phase for a CSTR is not dominated by a convective term as in the case of a plug flow reactor [22]. As seen earlier, the fluid mass balance in a fixed bed reactor with dispersed plug flow is given by:

$$-D_L \frac{\partial}{\partial z}\left(c\frac{\partial y_i}{\partial z}\right) + \frac{\partial}{\partial z}(cv_i) + \frac{\partial c_i}{\partial t} + \frac{(1-\varepsilon)}{\varepsilon}\frac{\partial \overline{q_i}}{\partial t} = 0 \qquad (3.1)$$

where D_L is the Fickian axial dispersion coefficient, v is the interstitial fluid velocity, $y_i(t)$ is the molar fraction of species i, c and c_i are the concentrations of the fluid phase and of the species i respectively, ε is the bed voidage, and $\overline{q_i}$ is the average concentration of species i in the solid.

The first term of eq. 3.1 takes into account axial dispersion, the second term takes into account advection. The third and fourth terms express the build up of species i in the fluid and the solid phase, respectively.

The mass balance in a CSTR is given by:

$$\frac{\varepsilon V}{R_g T}\frac{d(y_i\,P)}{dt} + (1-\varepsilon)V\frac{d\overline{q}_i}{dt} = F_{in}\,y_{i,in} - F_{out}\,y_i \tag{3.2}$$

where V is the volume of the reactor, T the temperature, P the pressure, R_g is the universal gas constant. F_{in} and F_{out} are the inlet and outlet flow rates respectively, while $y_{i,in}$ is the molar fraction of species i in the inlet. In a CSTR there is no diffusion in the z direction as perfect mixing is assumed, so the axial diffusive term $-D_L\frac{\partial^2 c_i}{\partial z^2}$ is not present in eq. 3.2. The advection term $\frac{\partial}{\partial z}(cv_i)$ is substituted by the finite difference $(F_{in}\,y_{i,in} - F_{out}\,y_i)$. Eq. 3.1 is usually a parabolic or hyperbolic equation (when the diffusive term is negligible) controlled by advection, and very hard to solve numerically [22]. Using eq. 3.2, in lieu of eq. 3.1, allows a faster solution of the mass balance equation.

The model of each CSTR is summarised in Table 3.1 where $i,j \in \{O_2, N_2\}$. The variables are the solid concentration $q_i(t,r)$, the molar fraction $y_i(t)$, the gas concentration $c_i(t) = \frac{Py_i}{R_gT}$, the outlet flow rate $F_{out}(t)$, and the pressure $P(t)$. The constants are the temperature, T, the radius of the adsorbent particle, R_p, the universal gas constant, R_g, the saturation limit, $q_{i,s}$, the Langmuir constant, b_i, the constant intrinsic mobility, D_{i_0}, the bed voidage, ε, the liquid film mass transfer coefficient, k_f, the volume of the reactor, V, the inlet flow rates F_{in}, and $y_{i,in}$, the molar fraction in the inlet flow. Kinetic and equilibrium data used for the simulation are summarised in Table 3.2.

The mass balance of the solid particle is expressed by eq. 3.3. Although it might be possible to use the LDF approximation by fitting the parameters of the model with experimental data [108], the resulting simulation would not be representative of the operation outside the range of parameters experimentally investigated to evaluate the LDF mass transfer coefficient. It could have been possible to simplify equation 3.3 by neglecting the dependence of the diffusivity on the concentration, but the simplification would have still restricted the field of applicability of the relation [108]. Accordingly, a Fickian equation has been used to describe the mass balance in the particle (eq. 3.3)

Table 3.1: Model equations for N_2 separation in a CSTR unit.

Mass balances:
$$\frac{\varepsilon V}{R_g T}\frac{d(y_i P)}{dt} + (1-\varepsilon)V\frac{d\overline{q}_i}{dt} = F_{in}\, y_{i,in} - F_{out}\, y_i$$

Diffusion:
$$\frac{\partial q_i}{\partial t} = \frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2 D_i \frac{\partial q_i}{\partial r}\right) \tag{3.3}$$

Constraint:
$$\sum_i y_i = 1$$

Boundary conditions:
$$\left(\frac{\partial q_i}{\partial r}\right)_{r=0} = 0 \tag{3.4}$$
$$\left(\frac{\partial q_i}{\partial r}\right)_{r=R_p} = k_f(c_i - c_{i_{R_p}}) \tag{3.5}$$

Equilibrium:
$$q_i(R_p) = \frac{q_{i,s} b_i c_{i,R_p}}{1+\sum b_i c_{i_{R_p}}} \tag{3.6}$$

Initial conditions:
$$y_i = y_{i,0}$$
$$q_i = q_i^*(y_{i,0})$$

which takes into account the dependence of the diffusivity on the gradient of the solid concentration within the particle: The diffusivity D_i has been described by

$$D_i = \frac{D_{i_0}}{1-\theta_i-\theta_j}\left((1-\theta_j) + \theta_i\frac{\partial q_j/\partial r}{\partial q_i/\partial r}\right) \tag{3.7}$$

where $\theta_i = \dfrac{q_i(t,R_p)}{q_{i,s}}$. This expression was proposed by Hagbood (cf. [108]) and applies to binary Langmuir systems where the saturation limits of the two species are the same.

The boundary condition, eq. 3.4, is the “symmetry condition”. The boundary condition at the surface of the particle, eq. 3.5, specifies that the gradient of the concentration of the adsorbate phase at the surface be proportional to the difference between the concentration of the fluid phase in the bulk and the value in equilibrium with the surface concentration in the solid. The equilibrium is described by the extended Langmuir isotherm, eq. 3.6.

3.2 Model for CO_2 separation from flue gases: an equilibrium controlled separation

As mentioned earlier, CO_2 separation is a relevant design problem not only for its environmental implications: CO_2 is the most strongly adsorbed species on the majority of adsorbents known, hence the operation is characterised by steep adsorption waves which

Table 3.2: Kinetic and Equilibrium data used in the simulations of air separation for N_2 production. Data are from [108], unless otherwise stated.

Air composition	O_2 N_2	0.21 0.79	
Particle size, R_p		0.3175	cm
Bed voidage, ε		0.5	
Ambient Temperature		298.15	K
Saturation constant, $q_{i,s}$	O_2 N_2	$2.64\,10^{-3}$ $2.64\,10^{-3}$	mol cm^{-3}
Equilibrium constant, $K_i = b_i\, q_{i,s}$	O_2 N_2	9.25 8.9	
Limiting diffusional time constant, D_{io}/R_p^2	O_2 N_2	$2.7\,10^{-3}$ $5.9\,10^{-5}$	s^{-1}
Mass transfer coefficient in external film [90], kf		0.02	ms^{-1}

make the numerical simulation of the process difficult to solve; furthermore, complex cycles are needed to meet the high level of CO_2 purity required, as the CO_2 is recovered in the extract phase. Since the choice of the adsorbent is a key element of PSA cycles, the models of the adsorption of flue gases on two viable adsorbents, HTlcs in section 3.2.1 and zeolite 13X in section 3.2.2, are presented so that a comparison of the performance of the two adsorbents can be investigated.

3.2.1 Model on HTlcs

The case study of interest is the separation of CO_2 from a flue gas. The feed is a typical stack effluent at 575 K, containing 15% of CO_2, 75% N_2 and 10% of H_2O [101]. The adsorbent is a K-promoted hydrotalcite-like compounds (K-HTlcs). HTlcs have a high selectivity towards CO_2 and show a higher capacity at high temperatures compared to other adsorbents. N_2 and H_2O are inert on the HTlcs [33], so that their capacity is not decreased by the presence of water whereas on other adsorbents H_2O is competitively adsorbed with CO_2.

As in the case of air separation, a series of CSTRs (continuous stirred tank reactors) has been used to describe each bed. This simplification reduces the computational requirements since the mass balance of the fluid phase for a CSTR is not dominated by a convective term as in the case of the PFR (plug flow reactor) [22].

The mass balance of the i-species in a CSTR is expressed by eq. 3.8.

$$\frac{dn_i}{dt} + V_{solid}\frac{d\overline{q}_i}{dt} = F_{in}\, y_{i,in} - F_{out}\, y_i \tag{3.8}$$

where n_i is the number of moles of species i, $\overline{q}_i$ is the average concentration in the solid phase, F_{in} and F_{out} are the inlet and output stream respectively, while $y_i = n_i/n$ and $y_{i,in}$ and are the molar fraction of species i in the reactor and in the inlet. In this case, the mass balance has been expressed in terms of number of moles rather than in terms of molar fractions (see eq. 3.2) as computational experiments showed that such equations could be more easily solved. This might be due to the fact that the amount of moles is intrinsically conservative.

A temperature dependent Langmuir isotherm model has been used to describe the equilibrium concentration of CO_2 [33, 101]:

$$\begin{aligned} b =& 2.03\, e^{\left(\frac{1.118}{T}\right)} \\ q^*_{\mathrm{CO}_2} =& (-1.5277\, 10^{-3}\, T + 1.7155)\left(\frac{b\, P_{\mathrm{CO}_2}}{1 + b\, P_{\mathrm{CO}_2}}\right) \end{aligned} \tag{3.9}$$

where $P_{\mathrm{CO}_2} = y_{\mathrm{CO}_2} P$ is the partial pressure of component i in the gas phase. Since N_2 and H_2O are inert, the adsorbed quantity at equilibrium is always zero for both species.

The energy balance has been included in the model as the process is non-isothermal (eq. 3.10):

$$\frac{1}{C_s}\frac{dT}{dt} = \sum_i \Delta H_i \frac{d\overline{q}_i}{dt} \tag{3.10}$$

where C_s is heat capacity of the bed.

The linear driving force (LDF) model has been used to describe the mass transfer in the solid phase [108]:

$$\frac{d\overline{q}_i}{dt} = k_i(q_i^* - \overline{q}_i) \tag{3.11}$$

The problem of the choice of the proper mass transfer mechanism for the case of CO_2 separation was underlined Diagne *et al.* [32], who noticed that the usual LDF approach fails to describe the kinetic of the desorption step of CO_2 on HTlcs. In fact the results showed the mass transfer coefficient to be significantly different between adsorption and desorption steps and mass transfer control was found to dominate the adsorption, depressurisation and purge step of the operation. A more detailed mass transfer model should be used, describing the mechanisms of intraparticle diffusion, but hit is not possible due to the lack of diffusivity data [32]. However, two different values of the mass transfer coefficient k of eq. 3.11 have been used in the adsorption and desorption steps (see Table 3.3), as suggested by Diagne *et al.* [32].

Other assumptions are ideal gas phase and negligible pressure drop. All data used are from [101], and are summarised in Table 3.3. Since N_2 and H_2O are inert on the K-HTlcs, $k_{N_2} = k_{H_2O} = 0$, and, similarly, $\Delta H_{N_2} = \Delta H_{H_2O} = 0$.

TABLE 3.3: Parameters for the adsorption model of CO_2 on HTlcs [101]. The data only refers to the CO_2 because N_2 and H_2O are not adsorbed on HTlcs.

Parameters		Values	Units
Bed length		0.2724	m
Bed radius		0.0387	m
CO_2 mass transfer coefficient	ads	0.0058	s^{-1}
	des	0.0006	s^{-1}
Heat of adsorption, ΔH		2.2200	kcal mol^{-1}
Heat capacity of the solid, C_s		850.0000	J kg^{-1} K^{-1}

Overall mass balances are carried out around each bed and around the whole cycle to check the correctness of the model and of the simulation.

3.2.2 Model on Zeolite 13X

HTlcs are more efficient adsorbents than zeolites due to their hydrophobicity, the lack of competitive adsorption of the N_2 in the feed, and the high capacity showed even at high temperatures [101]. The simulation of a PSA process on zeolite 13X for CO_2 capture has been included in this study to verify and quantify the differences in the performance of the two adsorbents. The model adopted is explained in this section, while the comparison is carried out in section 5.5.2.

TABLE 3.4: Value of parameters of the Dual-Site Langmuir used in eq. 3.13

	CO_2 (i=1)	N_2 (i=2)	units
$\xi_{1,i(1)}$	2.817269	1.889581045	mol kg^{-1}
$\xi_{2,i(1)}$	-3.500000 10^{-4}	-2.246200000 10^{-4}	K^{-1}
$\xi_{3,i(1)}$	2.830000 10^{-9}	1.163388000 10^{-9}	Pa^{-1}
$\xi_{4,i(1)}$	2598.203000	1944.605788000	K
$\xi_{1,i(2)}$	3.970888	1.889581045	mol kg^{-1}
$\xi_{2,i(2)}$	-4.950000 10^{-3}	-2.246200000 10^{-4}	K^{-1}
$\xi_{3,i(2)}$	4.311000 10^{-9}	1.163388000 10^{-9}	Pa^{-1}
$\xi_{4,i(2)}$	3594.071000	1944.605788000	K

Zeolites are porous crystalline aluminosilicates, consisting of an assemblage of SiO_4 and AlO_4 tetrahedra. The crystal lattice contains pores of molecular dimensions into which guest molecules can penetrate [107]. The characteristic of zeolites is that the micropore structure is perfectly uniform, with no pore size distribution since the lattice is determined by crystal lattice. In a zeolite X, the Si/Al ratio lies within the range 1-1.5, hence the zeolite is hydrophilic: the transition from hydrophilic to hydrophobic behaviour occurs for Si/Al ratios above 8. While the expression of the mass balance equation. eq. 3.8, does not change using a different adsorbent, the value of the LDF mass transfer coefficient and the equation of the equilibrium isotherm are different. The model used to describe mass transfer and equilibrium have been obtained by Ko *et al.* [65]. The data refer to a binary mixture of CO_2 and N_2, as in the case of adsorption of flue gases on zeolites the H_2O is always eliminated before the separation process.

A dual-site Langmuir isotherm is used to describe the adsorption equilibrium:

$$q_i = \frac{q_{mi(1)} b_{i(1)} y_i P}{1 + \sum_{i=1}^{n} b_{i(1)} * y_i * P} + \frac{q_{mi(2)} b_{i(2)} y_i P}{1 + \sum_{i=1}^{n} b_{i(2)} y_i P} \tag{3.12}$$

The isotherm parameters q_{mi} and b_i where calculated using equations 3.13 and the isotherm parameters ξ_{ij} shown in Table 3.4.

$$\begin{aligned} q_{mi(1)} &= \xi_{1,i(1)} + \xi_{2,i(1)} T \\ b_{i(1)} &= \xi_{3,i(1)} exp(\xi_{4,i(1)}/T) \\ q_{mi(2)} &= \xi_{1,i(2)} + \xi_{2,i(2)} T \\ b_{i(2)} &= \xi_{3,i(2)} exp(\xi_{4,i(2)}/T) \end{aligned} \tag{3.13}$$

The mass transfer has been described by the LDF equation 3.11. The value of the LDF mass transfer coefficient has been calculated via eq. 3.14.

Table 3.5: Equations to calculate the diffusion coefficients for a binary mixture of CO_2 and N_2 on zeolite 13X [65]

Knudsen diffusion	$D_{K,i} = 48.5\, D_{\text{pore}} \sqrt{\dfrac{T}{M_{w,i}}}$
Molecular diffusion	$D_M = \dfrac{100\, T^{1.75} \sqrt{\frac{M_{w\mathrm{N}_2} + M_{w\mathrm{CO}_2}}{M_{w\mathrm{N}_2}\, M_{w\mathrm{CO}_2}}}}{P \left(D_{v,\mathrm{N}_2}^{1/3} + D_{v,\mathrm{CO}_2}^{1/3} \right)^2}$
Effective diffusion	$D_{e,i} = \dfrac{\epsilon_p}{\tau_p} \dfrac{D_{K,i} D_M}{D_{K,i} + D_M}$

$$k_i = \frac{15 D_{e,i}}{R_p^2} \tag{3.14}$$

where $D_{e,i}$ is the effective diffusion coefficient of species i, which takes into account both Knudsen and molecular diffusion. The diffusion coefficients involved have been calculated according to the equations in Table 3.5.

The physical data used were also taken from [65], and are summarised in Table 3.6.

Table 3.6: Parameters for th adsorption model of a binary mixture of CO_2 and N_2 on zeolite 13X

Parameters		Values	Units
Pore diameter, D_{pore}		$1.00\, 10^{-9}$	m
Particle radius, R_p		$1.00\, 10^{-3}$	m
Particle tortuosity, τ_p		4.50	m
Diffusion volume, D_v	CO_2	26.90	
	N_2	18.50	
Heat capacity of the solid, C_s		504.00	J kg^{-1} K^{-1}
Heat of adsorption [18], ΔH_i	CO_2	$3.50\, 10^4$	J mol^{-1}
	N_2	$2.50\, 10^4$	J mol^{-1}

3.3 Simulation of a PSA cycle

In this chapter, two different cycle configurations are used for the two case studies: the configuration used for the N_2 case is the 2-bed/4-step Skartrom cycle of Fig. 1.2: each bed undergoes pressurisation with feed, adsorption, blowdown and desorption with purge. For the CO_2 the configuration (Fig. 3.1) is a modified Skarstrom cycle, operating among 4 beds [101] illustrated in Table 3.7.

TABLE 3.7: 4-bed/4-step PSA process used for the CO_2 case. The configuration has been proposed by Reynolds *et al.* [101]

	STEP			
BED	*I*	*II*	*III*	*IV*
1	Ads	Blow ↓	Des	Press ↑
2	Blow ↓	Des	Press ↑	Ads
3	Des	Press ↑	Ads	Blow ↓
4	Press ↑	Ads	Blow ↓	Des

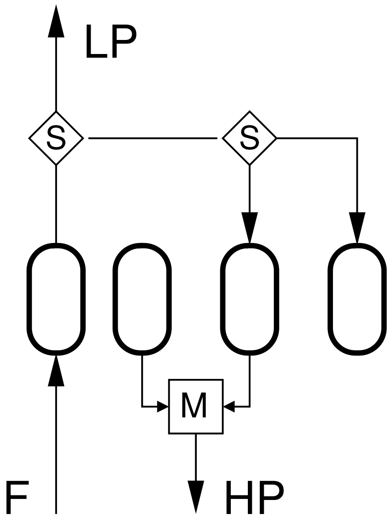


FIGURE 3.1: Flowsheet of the 4-beds/4-steps VSA cycle. Representation of the schedule using only one bed. The bed used for reference is bed 1 in Table 3.7

The four steps are high-pressure adsorption (P_H) with feed (step *I*), countercurrent blowdown to a vacuum pressure (P_L) (step *II*), desorption with light product purge (step *III*) and repressurisation with light product gas. It is a modified Skarstrom cycle where the repressurisation is done with a fraction of the light product obtained during the adsorption step rather than with the feed. Hence, less CO_2 is adsorbed in the bed during pressurisation, and the bed has a bigger capacity to adsorb CO_2 during the following adsorption step. The CO_2 adsorbed during the adsorption step is collected as an enriched-CO_2 stream during steps *II* and *III*.

Although different bed interconnections are implemented in the two cycle configurations, the building steps are essentially the same. The models introduced in the previous sections allow to describe the behaviour of the concentration, temperature, pressure and outlet streams profiles during the steps of a PSA cycle given an inlet stream. In the remainder of this section, it is explained how the models have been used to simulate the four basic steps of the Skarstrom cycle.

During the constant pressure steps (adsorption at high pressure, and desorption at low pressure) the unknowns are the concentrations of the two species in the solid and gas phases, q_i and y_i (or equivalently n_i in the case of CO_2) respectively, and the outlet flow rate F_{out}. During blowdown, the same set of unknowns (q_i, y_i, F_{out}) holds, but the value of the inlet flow rate is set to be zero. For brevity, a tuple $B = \langle y(t), q(t), T(t)\rangle$ is defined, where all the variables which describe the condition of the beds are stored. The solution of the model for a constant pressure step can be summarised by the following mathematical expression:

$$[B(t),\, F_{out}(t),\, P(t),\, B_f] = M(P_0,\, P_f,\, B_0,\, y_{in},\, F_{in},\, t_0,\, t_f,\, f) \tag{3.15}$$

where the concentration profile within the bed ($B(t)$), the pressure profile $P(t)$ and the final conditions of the bed (B_f) are calculated as a function, M, of the initial and final value of the pressure (P_0 and P_f) during the step, the initial conditions of the bed (B_0), the concentration and feed rate of the inlet (y_{in} and F_{in}), and the duration of the step (t_0, t_f). The function"M" corresponds to the model. The last argument of the function, f, indicates the direction of the flow during the step: in the implementation, $f = 1$ if the step is cocurrent, -1 otherwise. During adsorption and desorption the pressure profile is constant ($P_0 = P_f$). The dependence of the pressure with time during blowdown is described by an exponential function, eq. 3.16, where $\overline{P}$ and $\bar{t}$ are the average values of pressure and time over the blowdown operation, and k is equal to 10 in order to have a steep change in P. Eq. 3.16 reproduces well the fact that pressure changes are usually fast, and it is more realistic than a linear trend with time [108]. The dependence is

$$P(t) = \overline{P} + (P_0 - P_f)\left\{1 - e^{[k_P(t-\bar{t})]}\right\} \tag{3.16}$$

The balances in the pressurisation steps are described by the same set of equations, but F_{out} is known and is equal to zero, while the pressure profile has to be calculated. P_H is a function of the duration of the pressurisation step, t_{press}, and of the inlet feed rate, F_{in}, and it will be determined by the mass balances which describe the operation. The lower pressure, P_L is a process specification. The solution of the model for a pressurisation step can be summarised by the mathematical expression:

$$[B(t),\, P_H,\, P(t),\, B_f] = M_P(P_0,\, B_0,\, y_{in},\, F_{in},\, t_0,\, t_f, f) \tag{3.17}$$

the main differences with the expression in eq. 3.15 is that the profile of the pressure is calculated, while no output stream needs to be calculated.

3.3.1 The unibed approach for the simulation of a PSA cycle

In the previous section, it has been explained how the model is solved to obtain the desired quantities ($P(t)$, $B(t)$, $F_{out}(t)$) for the basic steps involved in PSA configurations. The aim of the section is to explain how the bed interconnections are implemented within the simulation tool. To this end, it is necessary to introduce the method used to achieve the convergence to cyclic steady state (CSS).

In literature, two main strategies to address the CSS constraint have been proposed: the Successive Substitution (SS) and the Simultaneous Discretisation (SD) method. In the successive substitution method the CSS is determined by repeated dynamic simulations, starting from a given initial state to some operating cycles. In the second approach, the SD, the equations describing the system are simultaneously discretised both in the spatial and temporal domains, and the periodicity conditions are posed as a constraint [63]. The SD approach has shown some limitations when used for complex PSA cycle configurations [90]. To avoid such limitations, a SS approach has been used in this thesis.

To achieve a reduction in computational time, a method to achieve a faster convergence to CSS has been adopted, originally proposed by Kumar *et al.* [69]: since at CSS each bed undergoes identical steps in a sequential manner, it is possible to simulate a multibed cycle using only one bed. As in [69], flow and composition of the outlet feed from a process step are stored in temporal effluent arrays, so that they can be used as an inlet stream when the bed is undergoing the appropriate process step. The resulting model is illustrated in Figs. 3.2 and 3.3. The method does not provide a correct description of the transition to CSS, as its basic assumption, that all the beds behave identically, is true only when CSS has been reached. However, it represents a powerful tool for the design of the operation since it allows a faster and correct prediction of the performance at CSS.

Now that the models of the two operations are known, as well as the unibed approach used, it is possible to introduce the algorithm used for the simulation of a PSA cycle. For simplicity, the Skarstrom cycle is used in the example (algorithm 3.1).

In the algorithm, the subscripts a, b, p and d refer to the adsorption, blowdown, pressurisation and depressurisation step, respectively. The four steps are simulated cyclically until the condition for convergence to CSS is met. The inputs are the cycle time, t_c, the low pressure, P_L, the initial condition of the bed, B_0, the split ratio, r_s and the schedule, S. The product stream flow rate, F_{prod}, is the fraction of the outlet flow rate withdrawn during the adsorption step (the *raffinate*). This percentage is expressed by the "split ratio", r_s, of the splitter which separates the raffinate into the product stream, and the

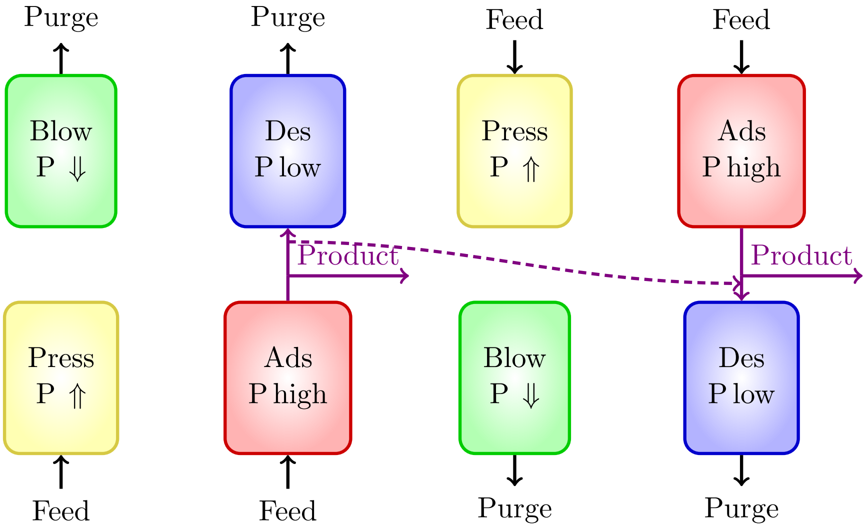


FIGURE 3.2: Full representation of a 2-bed/4-step Skarstrom Cycle. At cyclic steady state, the product provided by one bed during the adsorption step, has exactly the same composition as the purge that it will receive form the other bed during the desorption step. A shortcut can then be applied and an appropriate fraction of what the bed produces during adsorption can be stored and used as a purge gas for the desorption step, as illustrated in Fig. 3.3.

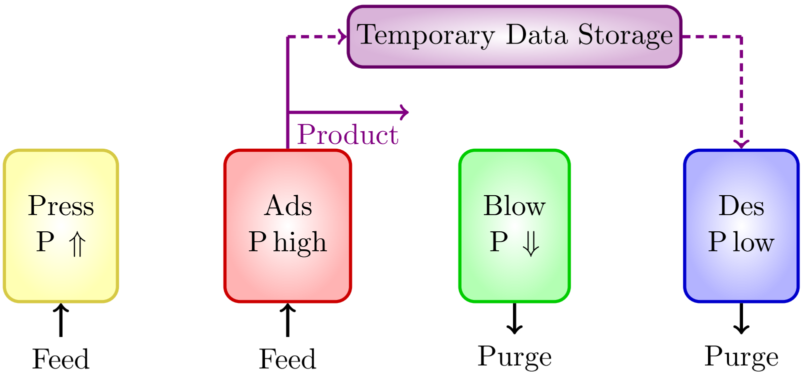


FIGURE 3.3: Compact representation of a 2-bed/4-step Skarstrom Cycle using only one bed. A fraction of the output of the adsorption step is stored in a temporal effluent array and used as a purge gas during the desorption step.

purge to be sent to the other bed (see line 15 of alg. 3.1):

$$F_{prod} = r_s \times F_{out,a}$$

.

The schedule, $S \in [0, 50]\%$, expresses the fraction of the cycle time occupied by each of the adsorption and desorption steps: $S_a = S_d = St_c$. The fact that adsorption and desorption steps are interconnected requires for the two steps have the same duration.

Algorithm 3.1 Dynamic simulation of a Skarstrom cycle

```
Output: $B(t)$, $P(t)$, $F_{prod}$: concentrations and pressure profiles and product stream
Input: $t_c$, $P_L$, $B_0$, $r_s$, $F_{in}$, $S$
 1: $t_0 \leftarrow 0$, $y(t) \leftarrow [\,]$, $P(t) \leftarrow [\,]$ {Initialise output vectors}
 2: $S_a = S_d \leftarrow S \times t_c$ {Steps duration}
 3: $S_p = S_b \leftarrow (S - 1) \times t_c$
 4:
 5: while The condition for CSS is not met do
 6:     I. Pressurisation step:
 7:     $t_{0,p} \leftarrow t_0$, $t_{f,p} \leftarrow t_{0,p} + S_p$
 8:     $[y_{in,p}, F_{in,p}] \leftarrow \mathrm{f}_{in}(t_{0,p}, t_{f,p}, F_{in}, y_{in})$
 9:     $[B_p(t), P_H, P_p(t), B_{0,a}]=\mathrm{M}_P(P_L, B_0, y_{in}, F_{in,p}, t_{0p}, t_{f,p}, 1)$;
10:
11:     II. Adsorption step:
12:     $t_{0,a} \leftarrow t_{f,p}$, $t_{f,a} \leftarrow t_{0,a} + S_a$
13:     $[y_{in,a}, F_{in,a}] \leftarrow \mathrm{f}_{in}(t_{0,a}, t_{f,a}, F_{in}, y_{in})$
14:     $[B_a(t), F_{out,a}, P_a(t), B_{0,b}]=\mathrm{M}(P_H, P_H, B_{0,a}, y_{in,a}, F_{in,a}, t_{0,p}, t_{f,p}, 1)$;
15:     $[F_{prod}, F_{in,d}]=\mathrm{splitter}(F_{out,a}, r_s)$ ;
16:
17:     III. Blowdown step:
18:     $t_{0,b} \leftarrow t_{f,a}$, $t_{f,b} \leftarrow t_{0,b} + S_b$
19:     $[y_{in,b}, F_{in,b}] \leftarrow \mathrm{f}_{in}(t_{0,b}, t_{f,b}, 0, 0)$
20:     $[B_b(t), F_{out,b}, P_b(t), B_{0,d}]=\mathrm{M}(P_H, P_L, B_{0,b}, y_{in,b}, F_{in,b}, t_{0,b}, t_{f,b}, -1)$;
21:
22:     IV. Depressurisation step:
23:     $t_{0,d} \leftarrow t_{f,b}$, $t_{f,d} \leftarrow t_{0,d} + S_d$
24:     $y_{in,d} \leftarrow y_a(t)$; $F_{in,d}$ calculated on line 15
25:     $[B_d(t), F_{out,d}, P_d(t), B_{f,d}]=\mathrm{M}(P_L, P_L, B_{0,d}, y_{in,d}, F_{in,d}, t_{0,d}, t_{f,d}, -1)$;
26:
27:     $B_0 \leftarrow B_{f,d}$ {The initial conditions of the bed for the next cycle are the final
        conditions after depressurisation}
28:     $t_0 \leftarrow t_0 + t_c$
29:     $B(t) \leftarrow y(t) \bigcup [B_p(t), B_a(t), B_b(t), B_d(t)]$
30:     $P(t) \leftarrow y(t) \bigcup [P_p(t), P_a(t), P_b(t), P_d(t)]$
31: end while
```

Consequently, the duration of the other two steps is given by $S_p = S_b = (1 - S)t_c$. The duration of the steps is calculated in lines 2 and 3 .

For steps p, a and b, a function f_{in} is used to generate the vectors y_{in} and F_{in}, describing the concentration and feed rate profiles of the inlet stream, given the duration of the step and the constant values of the concentration and feed rate. The inlet of the depressurisation step, defined in line 24, is a fraction of the output stream of the adsorption output stream (see line 15) and has the same concentration as the product.

3.4 Numerical methods for the simulation of PSA processes

The solution of the equations 3.15 and 3.17 requires the solution of the models introduced in sections 3.1 and 3.2. The analysis for the appropriate numerical methods for the two case studies is carried out in this section.

3.4.1 N_2 case study

The system of equations summarised in Table 3.1 was turned into a system of DAEs by spatial discretisation using the finite difference central operator (eq. 3.18). Ten points have been used for the discretisation of the radius. The resulting DAEs were solved using the Crank-Nicholson method and an explicit adaptive-step Runge-Kutta method [95]. In the latter case, the codes ode23 and ode45, provided by Octave-Forge have been used [91].

The solution strategy for the system of PDAEs has been:

1. Discretisation in space: *Finite Difference Method*

$$e.g. \left(\frac{d^2u}{dx^2}\right)_{\overline{x}} \simeq \frac{u(\overline{x}-\delta x) - 2u(\overline{x}) + u(\overline{x}+\delta x)}{\delta x^2} \tag{3.18}$$

2. Two methods have been implemented for the integration in time:

   - *Crank-Nicholson Method*

   $$\left(\frac{dy}{dt}\right)_{\bar{t}} = f(\bar{t}) \Rightarrow y(\bar{t}+\delta t) - y(\bar{t}) \simeq \frac{\delta t}{2}\left(f(\bar{t}+\delta t) + f(\bar{t})\right) \tag{3.19}$$

     The resulting nonlinear system of equations has been solved by the *Newton-Raphson Method*

   - Adaptive $4th/5th$ and $2th/3th$ order *Runge-Kutta* method

The concentration profiles obtained by ode23 and ode45, with the default value of tolerance of 10^{-6}, reproduced exactly the concentration profiles obtained using the Crank-Nicholson method using a small δt. The comparison between these profiles, over a PSA cycle, is shown in Fig. 3.4.

To trace the concentration profile of a PSA cycle with a cycle time of $120\,s$, ode23 required 290 points (i.e. points in time at which the full system of equations was solved),

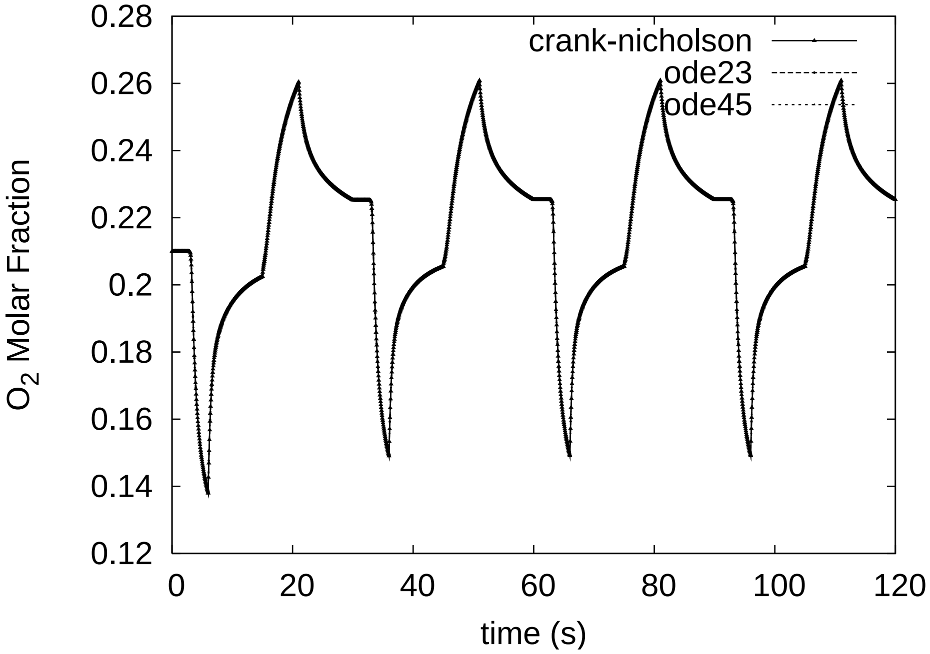


FIGURE 3.4: Concentration profiles of the O_2 during a PSA cycle. The profiles have been obtained from my Crank-Nicholson code with $\partial t = 0.05\,s$, and with the two adaptive time-step algorithms implemented by ode23 and ode45, with a tolerance of 10^{-6}.

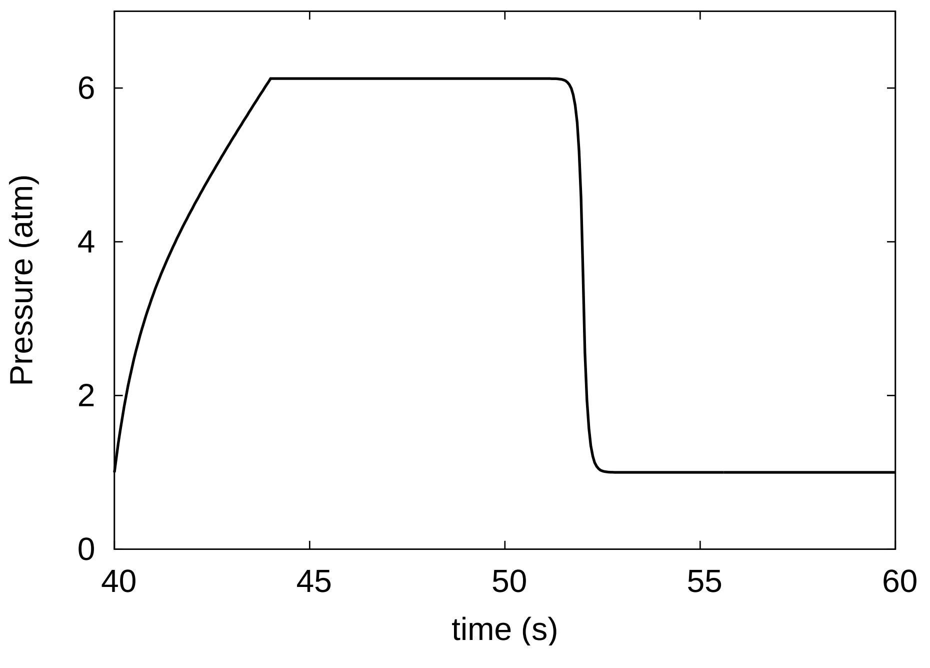


FIGURE 3.5: Pressure profile during the 4 steps of the Skarstrom cycle.

ode45 required 900 points and the Crank-Nicholson method 2417 points. As a result, both ode23 and ode 45 are faster than the Crank-Nicholson solution scheme. ode23 is the fastest of the three methods.

In Fig. 3.5 it is shown how the pressure varies during the cycle. The pressure profile during the adsorption step is calculated through eq. 3.17, while during the other steps is calculated by eq. 3.16.

3.4.2 CO_2 case study

The model of CO_2 consists of a set of DAEs, so that no spatial discretisation is needed. The first attempt to solve these equations with ode45 and ode23 as in the N_2 case failed, as the solver could never meet the accuracy required without excessively decreasing the time step δt. A possible reason for the increased difficulty to solve this model is that the concentration fronts are sharper because the separation is equilibrium controlled. This difficulty motivated further investigations into the shape of the concentration front and the choice of an appropriate numerical method.

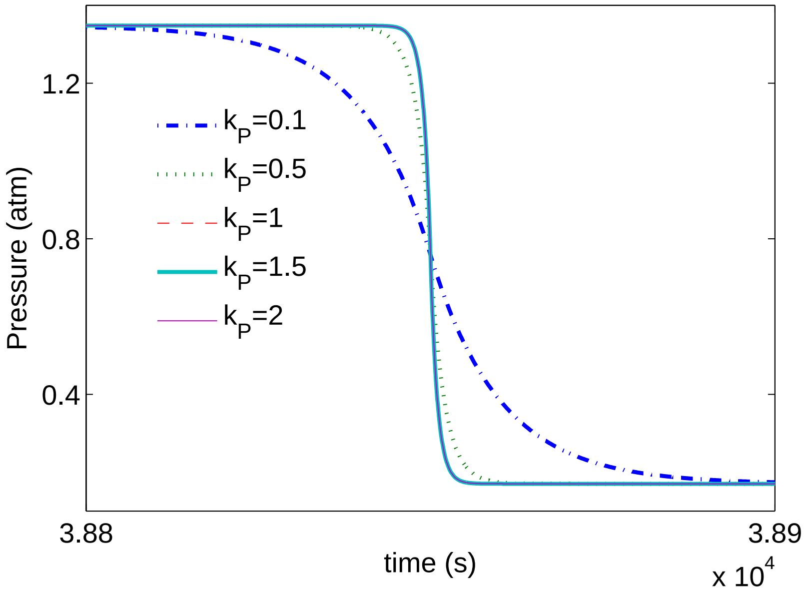


FIGURE 3.6: Pressure profiles during the blowdown steps for different values of k_P. k_P is the coefficient of eq. 3.16 that determines how quickly the pressure change with time: the lower k_P, the sharper the pressure profile during the blowdown step and, consequently, the sharper the concentration profiles

A preliminary investigation showed the influence of the pressure profile $P(t)$ during the blowdown step on the concentration profile: as shown in Figs. 3.6 and 3.7, the lower the value of k_P in equation 3.16, the sharper the pressure profile during the blowdown step

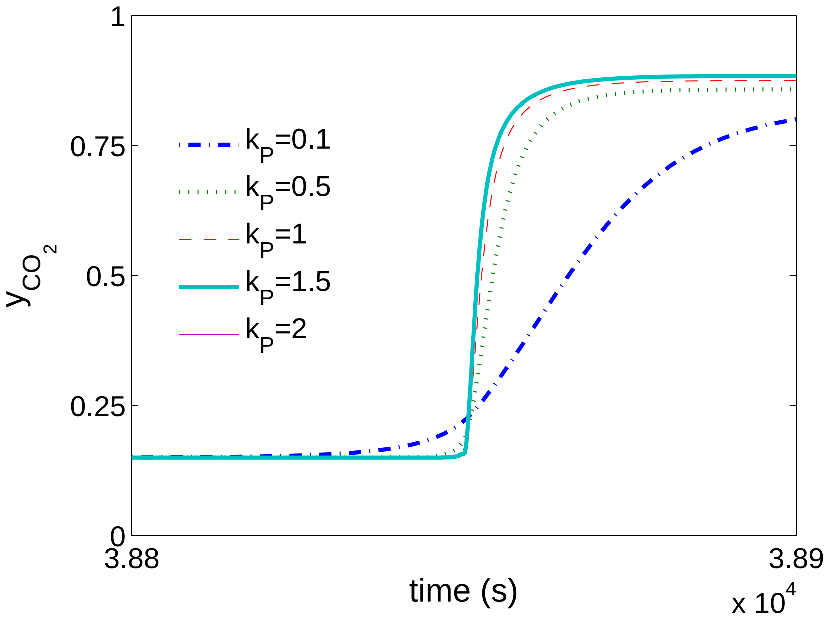


FIGURE 3.7: CO_2 concentration profiles during the blowdown steps for different values of k. As the pressure decreases the CO_2 desorbs and its concentration in the gas phase increases.

and, consequently, the sharper the concentration profiles. Hence, a lower k_P implies a bigger challenge to the numerical solver. A value of 0.5 has been chosen for k_P as a reasonable trade-off between a reasonable computational time and a pressure profile with the desired exponential shape.

For the CO_2 case study, the interpolations between one CSTR and the other have been avoided by solving all the CSTRs at the same time, by including the equation of all the CSTRs involved in a single set of equations. This approach could not been applied in a straightforward way since the solution of the fluid mass balance of each CSTR, eq. 3.8, requires that the inlet flow rate F_{in} is known. When simulating the CSTRs in series, the $F_{in,k}$ to the k-th CSTR is obtained from the simulation of the $(k-1)$-th CSTR. This difficulty has been overcome by exploiting the fact that we are assuming that the gas follows the ideal gas law and that no pressure drop occurs within the bed, so that the term dP/dt must be the same in all CSTR. According to the ideal gas law, the number of moles n_k in the k-th CSTR is given by $n_k = \dfrac{PV_f}{R_gT_k}$, where T_k is the temperature of the reactor. If we differentiate, we obtain

$$\frac{dn_k}{dt} = \frac{V_f}{R_g}\left[\frac{1}{T_k}\frac{dP}{dt} - \frac{P}{T_k^2}\frac{dT_k}{dt}\right] \tag{3.20}$$

For the k-th CSTR, defining $J_k = V_{solid} d\overline{q}/dt$ and using eq. 3.20, the fluid mass balance equation 3.8 can be rewritten as

$$\frac{V_f}{R_g}\left[\frac{1}{T_k}\frac{dP}{dt} - \frac{P}{T_k^2}\frac{dT_k}{dt}\right] + J_k = F_{k-1}y_{k-1} - F_k y_k \tag{3.21}$$

Eq. 3.21 holds for all the CSTRs in the series. By imposing that the reactors k and $k+1$ have the same dP/dt, i.e. imposing that the right hand side of equation 3.21 written for reactor k equals the one written for reactor $k+1$, we obtain the system of equations:

$$-T_k y_{k-1}F_{k-1} + (T_k + T_{k+1}) y_k F_k - T_{k+1} y_{k+1} F_{k+1} = T_{k+1}J_{k+1} - T_k J_k + n_k \frac{dT_k}{dt} - n_{k+1}\frac{dT_{k+1}}{dt} \tag{3.22}$$

which can be solved for all F_k, after eq. 3.10 and 3.11 have been calculated to obtain the gradients of T and $\overline{q}$.

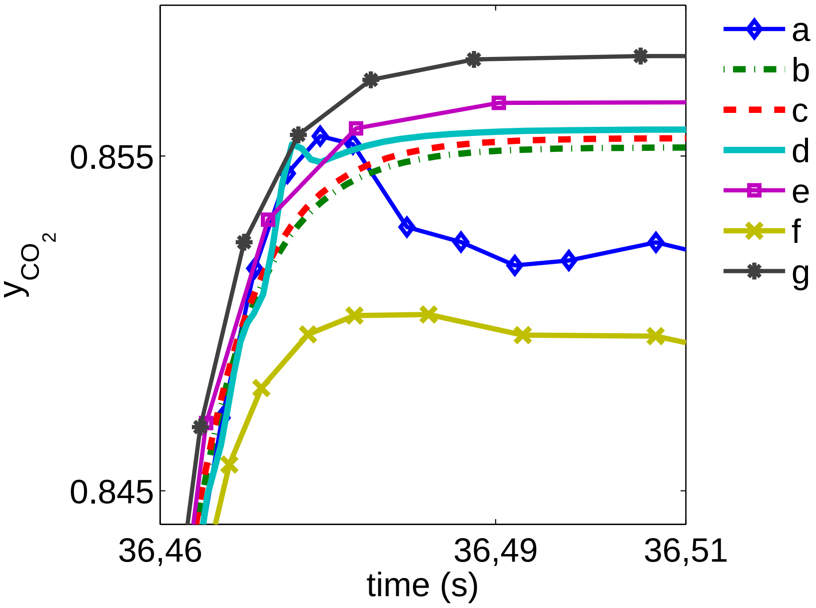
A trade-off between the quality of the concentration profiles obtained (in terms of regularity of the profiles) and the computational time required by the simulation needs to be taken into account to choose the appropriate numerical method. All the DAEs solvers provided by Matlab 7.1 [125] have been tested and the results are summarised in 3.8. HTlcs has been used as adsorbent in the simulations.

The Runge-Kutta methods (ode23 and ode45) fails to describe the concentration profiles: the step size always grows too small before the required accuracy of the solution is met. The same occurs when using the Adams-Bashforth-Moulton algorithm (by calling ode113).

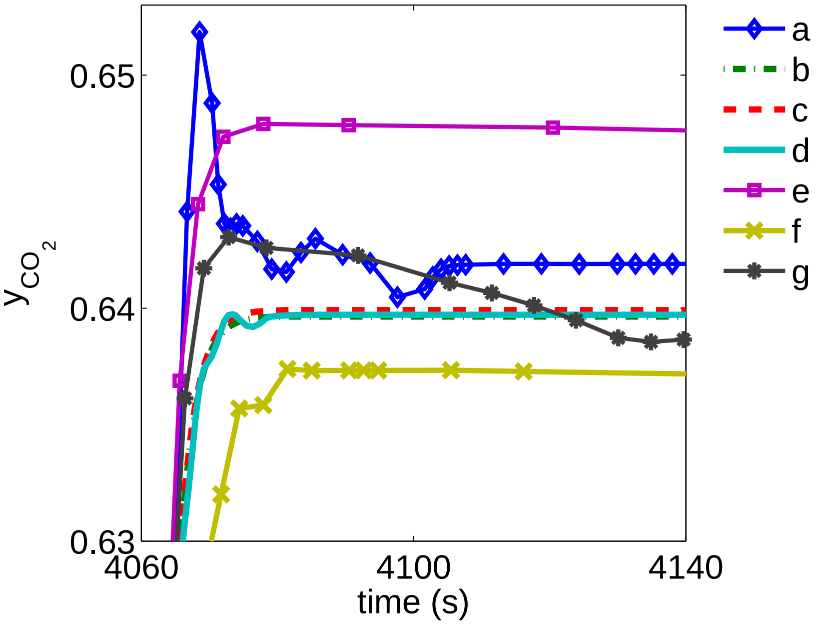
To investigate the performance of the Numerical Differentiation Formulas (NDF), the influence of the requested accuracy has been investigated. At each step, the solver estimates the local error e in the i-th component of the solution. This error must be less than or equal to the acceptable error, which is a function of the specified relative tolerance, $RelTol$, and the specified absolute tolerance, $AbsTol$:

$$|e_i| \leq \max(RelTol\,|y_i|, AbsTol(i)) \tag{3.23}$$

where y is the vector of the unknowns in the set of ordinary differential equations $y' = f(y,t)$. The effect of $RelTol$ and $AbsTol$ is explained in the MATLAB manual [125]. The relative error tolerance, $RelTol$, applies to all components of the solution vector y. It is a measure of the error relative to the size of each solution component. It controls the number of correct digits in all solution components, except those smaller than thresholds $AbsTol(i)$. The default, 1×10^{-3}, corresponds to 0.1% accuracy.



(a)



(b)

FIGURE 3.8: Effect of the choice of the numerical method on the smoothness of the concentration profiles at two different times. The details of the numerical methods used are explained in Table 3.8.

TABLE 3.8: Summary of the performance of different numerical methods for the simulation of a PSA cycle to separate CO_2 from a flue gas using HTlcs. "Time" refers to the computational time needed to simulate a cycle until CSS.

Symbol	Method	Time (s)	Observation
a	ode15s: Numerical Differentiation Formulas (NDFs), Rel Tol=1e^{-3} Abs Tol=1e^{-6}	182.73	Irregular
b	ode15s: NDFs Rel Tol=1e^{-8}, Abs Tol=1e^{-10}	945.40	Good profiles, high time
c	ode15s: NDFs+Backward Differentiation Formula (BDF), or Gear's Method Rel Tol=1e^{-3}, Abs Tol=1e^{-6}	164.30	Best
d	ode15s: NDFs +BDF Rel Tol=1e^{-6}, Abs Tol=1e^{-8}	296.00	Oscillations
e	ode23s: modified Rosenbrock formula of order 2	386.40 .	Good profiles, high time
f	ode23t: implementation of the trapezoidal rule (TR) using a "free" interpolant	142.00	Irregular
g	ode23tb: implementation of TR-BDF2, an implicit Runge-Kutta formula with a 1st stage that is a trapezoidal rule step and a 2nd stage that is a BDF of order 2.	144.20	Irregular
–	ode45: explicit Runge-Kutta (4,5) formula	–	Fails
–	ode23: explicit Runge-Kutta (2,3)	–	Fails
–	ode113: Adams-Bashforth-Moulton	–	Fails

The absolute error tolerance, $AbsTol$, applies to the individual components of the solution vector. $AbsTol(i)$ is a threshold below which the value of the i-th solution component is unimportant. The absolute error tolerances determine the accuracy when the solution approaches zero.

The ODE solvers deliver less accuracy for problems integrated over "long" intervals and problems that are unstable. Difficult problems may require tighter tolerances than the default values.

The NDFs provided irregular profiles when using the default values of the tolerances (case *a* in figures 3.8 and Table 3.8). In case *b*, tighter tolerances have been used and smoother profiles have been obtained but at a much higher computational time. When the Backward Differentiation Formula (BDF) is used, case *c*, regular profiles are obtained when using the default tolerances. On the other hands, when tighter tolerances are used, case *d*, irregular oscillations are observed, and a higher computational time is required.

Also the Rosenbrock formula, case *e*, successfully describes smooth profiles with the default tolerances, but a higher computational time is required than in case *c*.

The trapezoidal rule, case *f*, obtains irregular concentration profiles. When combining the trapezoidal rule with a backward differentiation step, case *g*, better profiles are obtained but yet less regular than in case *c*.

According to the above analysis, the implicit differentiation method ode15s (NFD+BDF) has been used for the simulation of CO_2 adsorption.

3.5 Validation of the unibed approach and detection of CSS

Figure 3.9 shows the profile of the recovery of CO_2 values evaluated at different design points. The recovery of CO_2 is defined as the fraction of CO_2 in the product stream, F_{prod}, with respect to the amount fed to the system over the cycle:

$$\text{recovery}_{\mathrm{CO_2}} = \frac{2 \int\limits_0^{t_{ads}} \left(F_{prod}\, y_{\mathrm{CO_2}} \right) dt}{\int\limits_0^{t_c} \left(F_{in}\, y_{\mathrm{CO_2},in} \right) dt} \tag{3.24}$$

The evaluation of the recovery requires the dynamic simulation of the PSA cycle until CSS is reached. In the results shown in Fig. 3.9 two different approaches for the convergence to CSS have been used: the unibed approach described earlier and the rigorous approach including all the 4 beds in the simulation. The aim is to validate the simplification introduced using the unibed approach.

The 4-bed/4step cycle used for the CO_2 case study has been used for this analysis because of the higher complexity of the configuration in Fig. 3.1 in comparison to the Skarstrom cycle. Each design point is indicated by a set of values for the cycle time t_c, the inlet flow rate F_{in} and the split ratio r_s.

Each point of the graph is the result of one simulation until CSS is reached, at a number of design points belonging to the line, in the design space, which connects point one, given by $x_1 = (t_{c,1}, r_{s,1}, Fin_1)$, and in point two $x_2 = (t_{c,2}, r_{s,2}, Fin_2)$. The value of λ is zero in x_1, and 1 in x_2, according to:

$$x = x_1 + \lambda(x_2 - x_1) \quad \text{for } \lambda \in 0, 1$$

In figure 4.1 the value of the recovery is shown as a function of λ. The two profiles obtained show that for each given design point, the same value of the recovery has been achieved both by using the unibed approach and the full simulation. The computational time required to generate the recovery profile decreased from 5392 s to 1583 s by using the unibed approach, thus justifying the use of such approach.

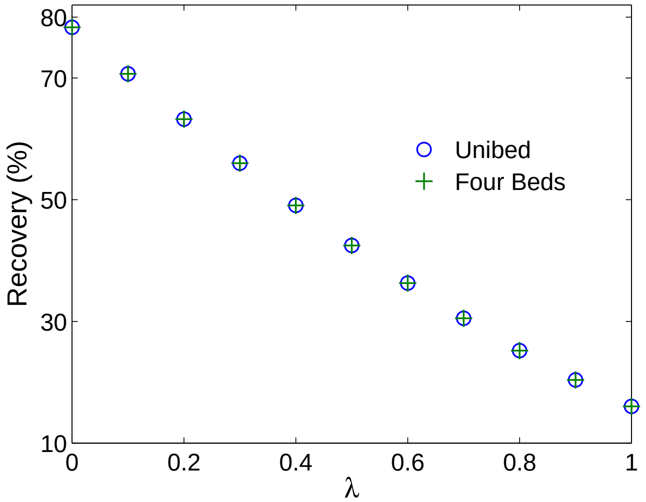


FIGURE 3.9: The values of recovery at CSS evaluated with a unibed approach match the results obtained with the full simulation. Computational time decreases by approximately 60% using the unibed approach.

The unibed strategy has been employed in this thesis to allow a decrease in the computational time. A decrease in computational time makes less problematic the use of successive substitution for detecting the CSS.

To detect CSS both accurately and time efficiently, it is necessary to choose appropriately how small the difference between the bed conditions at the beginning, B_0, and at the end, B_f, of the cycle has to be for the CSS to be reached. A common method to detect CSS is to require the difference ϵ_{CSS} (see eq. 3.25) between the condition of the bed (B) at the beginning and at end of the cycle to be a very small number:

$$\epsilon_{CSS} = ||(B_f - B_0)||_\infty \longrightarrow 0 \tag{3.25}$$

In Figs. 3.10 and 3.11 the effect of the value of ϵ_{CSS} on the recovery profile and on the computational time is shown, respectively. The data refer to the N_2 case study. As in Fig. 3.9, the recovery profiles are shown against the parameter λ. As the value of ϵ_{CSS} decreases, the recovery profiles converge. Convergence is achieved for $\epsilon_{CSS} = 10^{-6}$. Any further decrease in the value of ϵ_{CSS} does not improve the quality of the recovery profile, while on the other hand implies a significant increase in the computational time required.

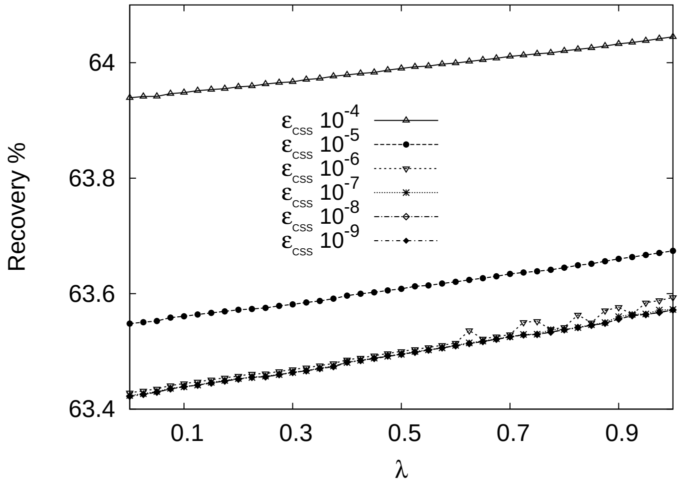


Figure 3.10: Influence of tightness of the constraint to cyclic steady state on the recovery.

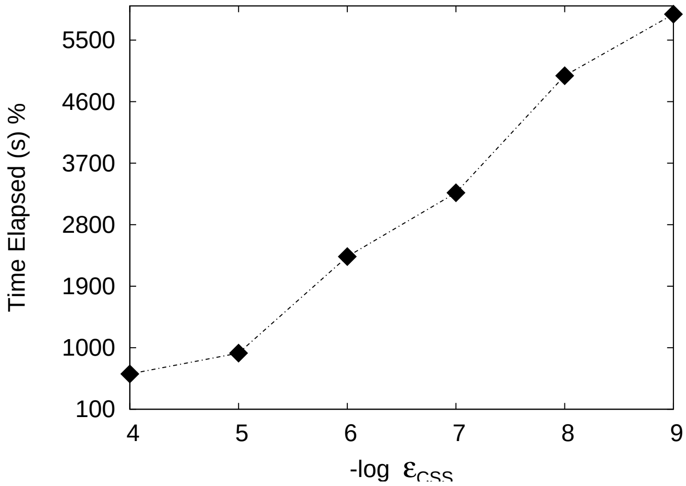


Figure 3.11: Increase of computational time due to an increase in the tightness of the constraint to cyclic steady state.

However, the criterion expressed by eq. 3.25 does not converge for the CO_2 case. As shown in Fig. 3.12, ϵ_{CSS} reaches very small values ($\simeq 10^{-5}$) before 30 cycles, and it then takes higher values again. On the other hand, the value of the purity reaches a

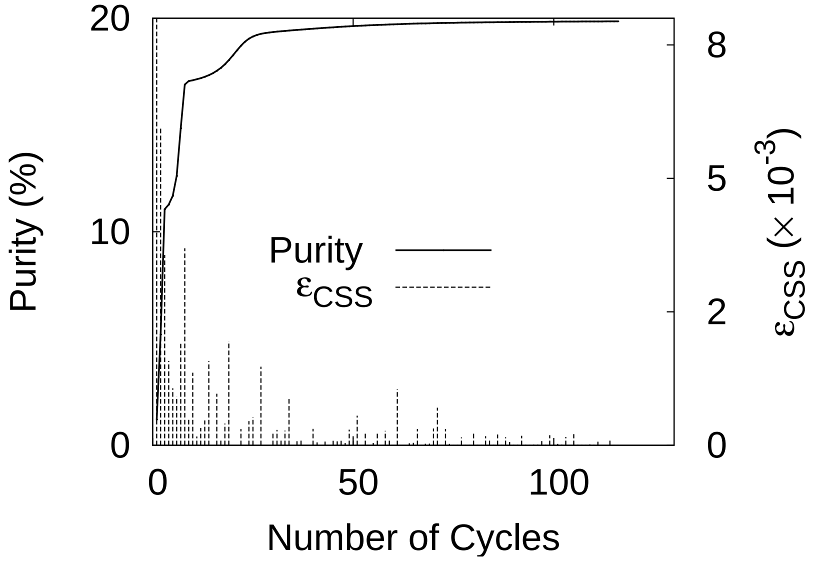


FIGURE 3.12: Convergence of ϵ_{CSS} (vertical impulses) and of the Purity (continuous curve) with the number of cycles

plateau only after around 60 cycles. As a result, the condition for the convergence to CSS in the CO_2 case study is the standard deviation of the value of the objectives in the last 10 cycles to be less than 10^{-3}. This value has been chosen as the best trade off between accuracy and computational time, with a procedure similar to the one adopted to decide the value of ϵ_{CSS} for the N_2 case.

3.6 Summary

In section 3.1, a detailed diffusion model has been used to describe the separation of air for N_2 production, a kinetically controlled separation. Computational time has been cut by simulating each bed involved in a cycle as a series of CSTRs. A unibed approach has been adopted to speed up the conversion to cyclic steady state.

The same approach has been used to describe the adsorption of a flue gas for CO_2 capture. Two adsorbents (HTlcs and zeolite 13X) have been taken into account, each requiring different relations to model mass transfer and equilibrium isotherms. In both cases, the mass transfer can be described by the LDF approximation because the separation is equilibrium controlled, but the steep adsorption fronts generated required an extra effort to detect an appropriate numerical method.

The aim of the project is to develop automated tools for the design of PSA cycles. The first step of the project has been the development of appropriate simulation tools, and has been discussed in this chapter. These provide the basis for the definition of a suitable objective function for the automated design of PSA operations, as presented in the next chapter.

Chapter 4

Introduction to the design of PSA cycles

The performance of PSA cycles depends on both the cycle configuration used, including the adsorbent chosen, and the operative condition under which it operates. The complex behaviour of the performance with respect to the process variables requires the development of automated tools for the design of PSA operations. Optimisation can be the right approach towards the systematic design of PSA cycles.

In this chapter, an overview of previous studies which focused on the design of PSA studies is given. This helps not only to familiarise with the design problem, but also to detect the gaps or areas of development which require further research effort.

The design of a simple Skarstrom cycle for the production of N_2 from air is used to start the investigation into the design problem. Accordingly, the focus of section 4.2 is on the characterisation of the objective function. The results of this analysis justify the class of optimisation methods (i.e. direct search method) used to detect the design conditions which maximise the recovery of N_2 in the product. The performance of different algorithms is compared.

4.1 Design of PSA cycles: an overview

A general formulation of the optimisation problem based on cyclic steady state of PSA systems can be formulated as follows [13]:

$$
\begin{aligned}
&\max_{y,\, y_0,\, q} f \\
&\\
&s.t. \\
&F(y, y_0, q, t) = 0; \\
&y(t_0) = y_0; \\
&t \in [t_0, t_1]; \\
&W(y(t), y_0, q) \leq 0; \\
&C(y_0, q) = y_n(t_n) - y_0 = 0; \\
&LB \leq (y, q) \leq UB;
\end{aligned}
\tag{4.1}
$$

where y is the combined vector holding state variables, y_0 are initial conditions for the state variables, q are the decision variables and both are subject to the lower and the upper bounds (LB and UB, respectively). The equality constraint $F(y, y_0, q, t)$=0 represents the solution of the model itself, while the inequality constraints, $W(y(t), y_0, q) \leq 0$, can express some specific requirements of the process (e.g. minimum purity). C are the CCS conditions, which have to be met. The decision variables in q can be geometric parameters (e.g. bed length, diameter, adsorbent packing) or process parameters (flow rates, step times, operating pressures). The objective function, f, can be, for example, the overall recovery or of the costs at desired purity or minimisation of work/power consumption at a desired purity.

In the review on simulation and optimisation of PSA operations carried out by Biegler *et al.* [13], four categories of optimisation strategies for PSA operations were identified:

- simplified optimisation methods
- black box optimisation
- equation-based optimisation based on complete discretisation
- simultaneous tailored optimisation

4.1.1 Simplified Optimisation Methods

These methods consist of the development of a simple PSA model, and of its fine-tuning using pilot-plant data, actual plant data or more detailed models. The results obtained for the simplified model are then implemented and compared with the more detailed one. Lewandowski *et al.* [75], after considering the expense of incorporating detailed adsorption model in an optimisation problem, developed a neural network model to describe a PSA process. The neural network model was used to minimise the N_2

production costs. The main limit of the approach, as pointed out by the authors, is that it only covers the design space covered by the experimental data used to build the model.

Smith and Westerberg [116, 117] reduced the PDAE model to a simple algebraic loading model, which was introduced into a mixed integer non-linear programming formulation for the synthesis of optimal process schedules (minimum number of beds, time to allocate to each step) to maximise profit or minimise annualised operating and capital costs.

A training neural network to describe PSA processes has also been developed by Sundaram [121]. Here, a methodology is proposed to determine the ranges of input-output measurements within which the neural network can reliably predict the process behaviour. The dependence of the model on the availability of experimental data is still an issue. The model developed was not used within an optimisation framework.

Guan *et al.* [42] developed an electrical network model for the simulation of PSA cycles. The aim was to develop an object-oriented programming technique to develop a model which could be used within a flowsheet design framework for PSA cycles. In [42], a four-bed/12-step PSA cycle for air separation for O_2 production has been proposed. Results of the simulation have been successfully compared with literature data. The authors tried to apply the sequential quadratic programming to the optimisation of the operation, but no convergence of the optimiser was achieved although a small single-objective problem, in a two-dimensional design space, was taken into account.

Agarwal *et al.* [4] recently considered “model reduction” as a valuable approach to reduce the computational efforts needed to simulate PSA processes within an optimisation framework. They developed reduced-order models (ROM) based on proper orthogonal decomposition (POD) as a low-dimensional approximation to a detailed PSA model. The ROM has been successfully used to optimise the recovery of H_2 from CH_4 in a simple PSA Skarstrom cycle. However, the ROM was able to mimic the behaviour of the PSA process only in the neighbourhood of the experimental conditions used to generate it. Hence, it could not be used as a model to optimise the process in a wide design space.

Rajasree and Moharir [99] developed an adaptive simulation/tuning/optimisation software for PSA design, able to provide the number of beds and optimal operating conditions for the desired performance. A simplified simulation was employed, tuned to match the performance of a rigorous PSA model. Single objective functions were considered, such as purity at a desired recovery level.

Simplified methods reduce computational effort, but on the other hand they cannot guarantee convergence to optimal performance and the conclusions obtained for a specific case are difficult to transfer among different PSA systems.

4.1.2 Black Box Optimisation

In the black-box approach the optimisation problem can be formulated as:

$$\begin{aligned} & \max_{q,\,y(q)} \mathrm{f} \\ & \\ & s.t. \\ & W(q) \leq 0; \\ & LB \leq q \leq UB; \end{aligned} \tag{4.2}$$

where the bed models and CSS conditions are solved implicitly. This means that the objective function is evaluated at CSS at a trial point of the search space determined by the optimiser. The values of the objective function and of the constraints are returned to the optimiser and, if necessary, gradients are calculated numerically via finite differencing.

Based on the information, the optimiser takes a new search direction. The black-box approach is carried out by developing a search grid that relates process performance (values of the objective function, e.g. purity, costs) to operating variables (decision variables q which constitute the search space, e.g. product flow, feed pressure, cycle time, valves and temperatures). Good results have been achieved by Kvamsdal and Hertzberg [70] using this method, together with the use of successive quadratic programming (SQP).

Black-box programming followed by a quadratic programming approach has also been used by Cruz *et al.* [24]. They developed a simplified procedure, considering either operating costs or capital costs as dominant factors in the economic performance of the system.

The black-box approach is advantageous because it is easy to implement, but it is usually expensive since each objective function evaluation requires the simulation till CSS is reached. While this drawback can be relatively small for gradient free methods (such as direct search and evolutionary methods), if gradient based methods are employed (e.g. SQP), a higher number of function evaluation is required to calculate derivatives. Furthermore, the use of derivatives implies a numerical approximation, which leads to loss of accuracy and deterioration of the optimisation performance [13].

4.1.3 Equation-based Optimisation Based on Complete Discretisation

The complete discretisation strategy is applied in time and space to the bed equations and the objective function, inequality constraints, CSS condition and bed equations. These are combined together and solved simultaneously. The problem was solved using nonlinear optimisation SQP (successive quadratic programming) in gPROMS for RPSA and small air separation cycles in an efficient way by Nilchan *et al.* [63, 90]. The approach is efficient for simple cycles, but it often has convergence problems with complicated cycle configurations and steep concentration waves. In these cases a fine spatial and temporal grid is required, which leads to a "non manageable" optimisation problem: within this thesis, the expression "non manageable" problems is used to for optimisation problems which require an impracticable computational time to be solved. For complicated models, the solver may fail because of the error accumulation due to complete discretisation [64].

4.1.4 Simultaneous Tailored Optimisation

The tailored approach was developed by Ding *et al.* [34] and Jiang *et al.* [54]; the formulation of the problem for this method is the following:

$$\begin{aligned}
&\max_{y, y_0, q} \text{ f} \\
&s.t. \\
&W(y(t, y_0, q)) \leq 0; \\
&C(y_0) = y_0 - y(t_n, y_0, q) = 0; \\
&LB \leq (y_0, q) \leq UB
\end{aligned} \tag{4.3}$$

The bed models have been eliminated from eq. 4.3 by solving it implicitly; the convergence of CSS is incorporated as a constraint in the optimisation problem. At each iteration, a detailed DAE bed model is solved in an inner loop, in order to obtain sensitivities as well as values of the constraints and objective function. Since satisfaction of the CSS is required only at the optimum, the time-consuming CSS loop is eliminated. The sharp concentration fronts are handled by the flux limiters and initialisation is straightforward. A stationary point of the equality constraints and the reduced gradient of the objective function is sought. The method has global properties, thus guaranteeing convergence even from poor starting points. The method was presented in three different publications: [54–56], as well as in the review on PSA design by the same authors [13].

Jian *et al.* [54] used a direct determination approach to detect CSS. The method employed a Newton-based method with accurate sensitivities analysis to achieve fast and robust convergence to CSS. Newton algorithms can achieve quadratic or superlinear convergence rates and require fewer iterations than successive substitution methods for the evaluation of CSS. However, extra computational time is needed for the calculation of the Jacobian. A disadvantage of the direct sensitivity approach is that the computational cost of sensitivities increases with the number of state variables and the number of decision variables. For the optimisation a state-of-the-art "reduced space Successive Quadratic Programming" rSQP-based optimisation algorithm was used.

The above strategy solution was extended to a multi-bed system [56]. The multiple bed configuration was approached by solving simultaneously all the beds; due to the size of the system, only half of the beds involved in the cycle could be considered in the study, so that a partial optimisation problem was presented.

The main drawback of this method is the sensitivities analysis required by the Newton-based approach. Such method leads to a better accuracy in the solution but it increases computational time. Biegler *et al.* [55] addressed the problem by using parallel computing for the sensitivity evaluations. In fact, the sensitivity calculation with respect to each parameter is independent, and parallelisation can be made by a distributed parameter approach: the sensitivity parameters are divided into different sets redirected to different processors, each running a copy of the state equations and computing a subset of sensitivity variables. As a result, the computational effort needed is independent of the number of design variables taken into account, which represents a big advantage with respect to the successive substitution methods for the detection of CCS.

Ko *et al.* [64, 65] developed a simultaneous tailored approximation approach based on the method proposed by Jiang *et al.* [54]. A PSA, a high temperature PSA (simple Skarstrom cycle) and a FV (fractionated volume) PSA are examined for CO_2 sequestration applications. The objective function is expressed as a function, Φ, based on purity, recovery or power consumption. The CSS condition is parametrised in function of a tolerance ϵ (e.g. $\epsilon = y_0 - y(t_n, y_0, q)$) and the overall objective function to be minimised is $\Phi + M\epsilon$, where M is a penalty constant (order of magnitude 10^5). The reduced space SQP (SRQPD) algorithm in gPROMS is used to solve the nonlinear programming problem for optimisation.

The method leads to a small optimisation problem which could be solved in gPROMS by the SQP solver. The main drawback is that CSS condition was only approximated and if the tolerance ϵ is small, convergence would be difficult to reach or computational time would be too long. A nonlinear regression step is needed to satisfy the modified expression of CSS, in order to describe the condition of CSS accurately. Furthermore, the nonlinearity of the CSS condition increases the difficulty of the computation.

4.1.5 Alternative approaches to PSA design

In the classification by Biegler *et al.* [13], non mathematical approaches to PSA design have been excluded. These methods improve the understanding of PSA operations, but do not provide a systematic solution to the design of optimal cycles.

One of these approaches [59] consists of the derivation of a model of the adsorbed phase using adsorption thermodynamics, which allowed the identification of entropy generation within the adsorption cycle. The objective function in this study is the minimisation of entropy generation. Results showed that major losses were due in the valve in the blowdown step, and feed compressor after cooler. Bed exergy losses were observed mainly during feed step and not during blowdown or purge step. The results would probably be the opposite in hte case of a VSA cycle.

The design of PSA processes was also approached by heuristic analysis [53] in order to develop easy-to-use rules to be used as a guide in PSA process design. The analysis was based on analysis of the properties of adsorbate-adsorbent systems and simulation results. Heuristics to help in the selection of adsorbent, particle size, bed size, bed configuration, purge volume, pressure equalisation and vacuum swing adsorption were provided, together with judgments on the impact of each variable on operating performance.

Optimal operating conditions for producing maximum net product have been identified via simulation by Kumar [59]. In particular, the author focused on the role of the adsorbent in the achievement of the better performance: the improved transfer characteristic of an adsorbent is useful but only up to a limit, after which it does not help to further reduce the mass-transfer resistance. The best adsorbent cannot be chosen by considering only the selectivity or working capacity, but it is necessary to consider a combination of the two.

4.2 Design of a Skarstrom Cycle for N_2 production from air

In most optimisation studies mentioned in the previous section, no proper justification for the choice of the optimisation method used has been provided. In this section, the single-objective optimisation of the performance of a Skarstrom cycle for the separation of air is carried out. The objective function is characterised to help choose the most appropriate class of optimisers to solve the optimisation problem.

The problem of interest is the maximisation of the recovery of N_2 in the product stream. A black-box approach has been used to avoid the limitations that complete discretisation approaches have with complex cycles.

The optimisation problem has been formulated as:

$$\begin{aligned}
&\max_{t_c,\, r_s,\, F_{in}} \ \text{Recovery at CSS} \\
\\
&s.t. \\
&\text{Evaluation of the model at CSS;} \\
&\text{Pressure} < 7\text{atm;}
\end{aligned} \tag{4.4}$$

As mentioned earlier, the recovery of N_2 is defined as the fraction of N_2 in the product stream, F_{prod}, with respect to the amount fed to the system over the cycle:

$$\text{recovery}_{N_2} = \frac{2 \int\limits_0^{t_{ads}} \left(F_{prod}\, y_{N_2} \right) dt}{\int\limits_0^{t_c} \left(F_{in}\, y_{N_2,in} \right) dt} \tag{4.5}$$

where $y_{N_2,in}$ is the molar fraction of N_2 in the inlet flow, F_{in}; t_{ads} is the duration of the adsorption step.

The design variables are the cycle time, t_c, which indicates the overall duration of the four steps, the flow rate of the inlet flow, F_{in}, and the split ratio r_s. For the simulation, the schedule for the whole operation was specified: if t_c is the duration of the entire cycle, the duration of the adsorption and of the desorption step are 3/10 of t_c, while blowdown and pressurisation step last 2/10 of t_c. As a consequence, as t_c increases, the duration of all the steps increases as well. The values of the schedule and of the maximum value of pressure allowed are indicated by Ruthven [107] as the most commonly used for air separation. If a design point violates the pressure constraint, the corresponding values of the recovery are set to 0.

For this preliminary analysis, only one CSTR has been used to simulate each bed of the cycle. Accordingly, we expect lower values of the recovery, since we will not take advantage of the countercurrent disposition of the current during blowdown and desorption steps. Figure 4.1 shows the shape of the objective function, the recovery, between two design points x_1 and x_2. The value of the recovery is shown as a function of a parameter $\lambda \in 0, \ldots, 1$, as in section 3.3.1. The Crank-Nicholson (CN) method has been used to integrate the system of ODEs obtained after spacial discretisation. Different values of δt have been used for the numerical solution of the model.

As δt decreases the profiles converge to a common value, thus suggesting that the solution method is convergent. The profiles show some ripples, more visible the higher the value of δt investigated (see Fig. 4.2), which indicate that the objective function is not smooth. By decreasing the δt the ripples become less appreciable, but we cannot assure that they disappear for a given value of δt, however small.

As δt decreases, the computational effort needed increases more than linearly. However, all the profiles obtained showed the same qualitative behaviour: as δt decreases the profile of the recovery shifts towards lower values, but maintains the same shape. This shift does not affect the position of the optima in the design space. As a consequence, the higher value of the δt ($\delta t = 0.05\,s$), and consequently the lower computational effort, can be used to detect the optima.

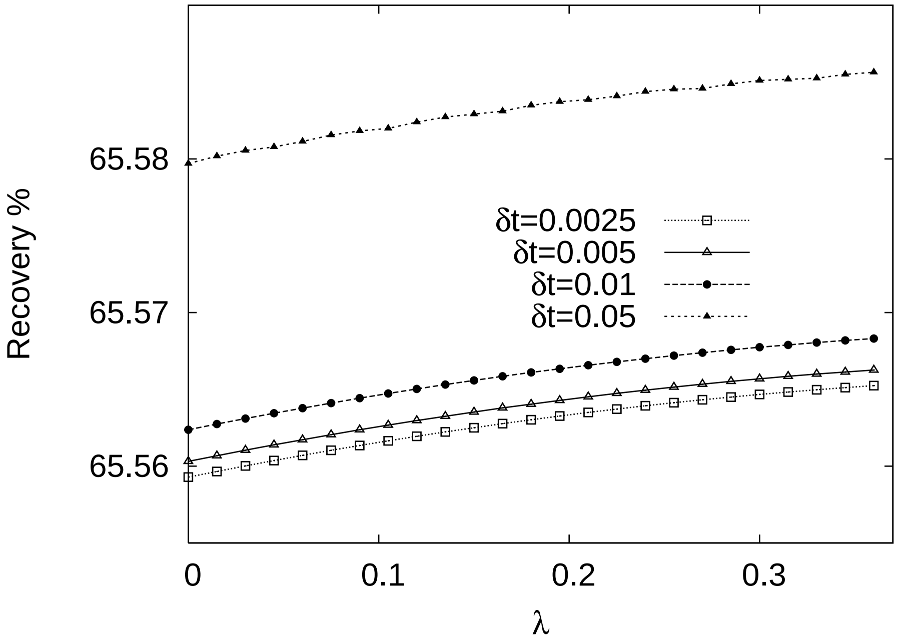


Figure 4.1: Influence of time step δt on the profile of the objective function when solving the system of ODEs with the Crank-Nicholson method (eq. 3.19.

The origin of the ripples has been further investigated by generating the recovery profiles using different quadrature methods for the evaluation of equation 4.5 and by using adaptive time-step integration methods to solve the system of ODEs.

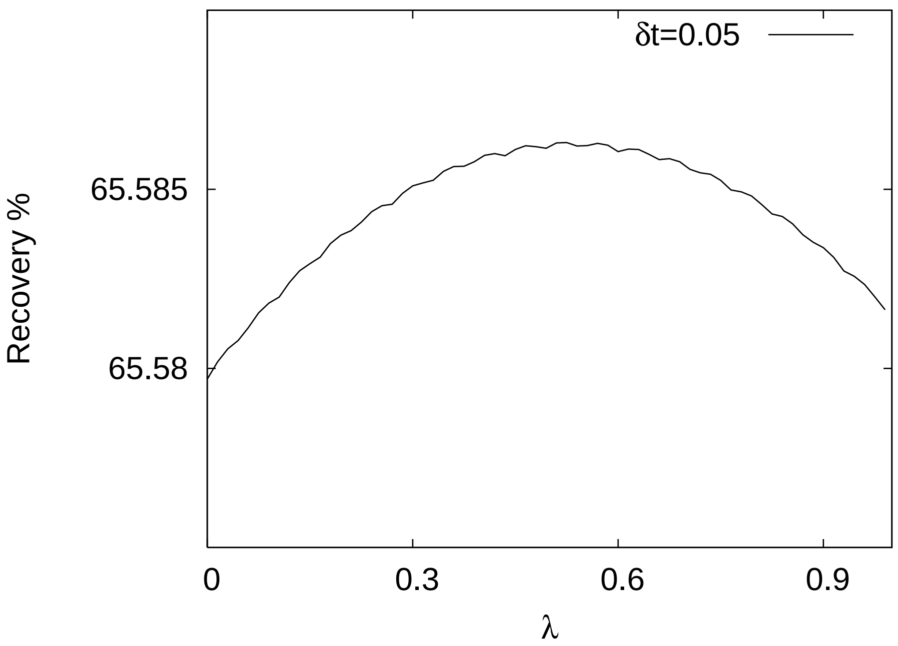


FIGURE 4.2: Recovery profile obtained with $\delta t = 0.05s$

Influence of quadrature method

The results shown in Fig. 4.1 and Fig. 4.2 have been generating by carrying out the integral 4.5 using a low accuracy quadrature method, known as the "rectangle rule":

$$I = \int_{t_0}^{t_n} f(t)dt \simeq \sum_{i=1}^{N} f(t_i)\delta t_i \tag{4.6}$$

where t_i are the points of the time grid used by the CN method. The order of accuracy is $O(\delta t_i^2)$. I compared the profiles of the objective function obtained by using eq. 4.6 with those obtained with a more accurate quadrature method, the trapezoidal rule [39]:

$$I = \int_{t_0}^{t_n} f(t)dt \simeq \sum_{i=1}^{N} \left[f(t_i) + f(t_{i+1})\right] \frac{\delta t_i}{2} \tag{4.7}$$

the order of accuracy of the trapezoidal rule is $O(\delta t_i^3)$ [39].

Using the same time spacing δt, the profiles of the recovery obtained with the two different quadrature methods shows no appreciable difference, as shown in Fig. 4.3 for $\delta t = 0.01$ s and $\delta t = 0.05$ s. This results did not motivate any further investigation into more accurate quadrature methods, such as the Simpson's rule.

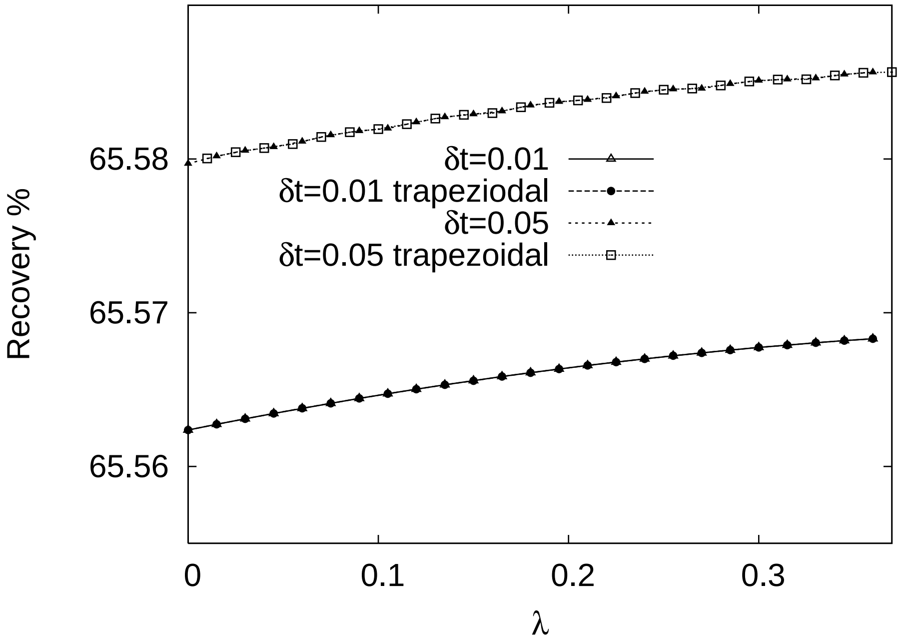


FIGURE 4.3: Profile of the recovery obtained with two different quadrature methods (4.6 and 4.7) with $\delta t = 0.05s$ and $\delta t = 0.01s$

The previous result allows us to conclude that the accuracy of the integration method is not responsible of the ripples, but they might be due to the approximations introduced by the numerical methods. This motivates the use of an adaptive-time stepping scheme to solve the ODEs, which could provide accurate results and at the same time improve the computational efficiency of the simulations.

Comparison with an adaptive time-step algorithm

As mentioned in section 3.4.1, the three numerical integrators used (i.e. CN, ode23 and ode45) can successfully detect the concentration profiles during a PSA cycle, as shown in figure 3.4. Even if the profiles of the concentration match those found by the Crank-Nicholson method with a small δt, the profiles of the recovery obtained from the quadrature (using the trapezoidal rule) of these profiles are not identical, as shown in Figure 4.4.

This might depend on the lower number of points that ode23 and ode45 use to describe the concentration profiles, with respect to the Crank-Nicholson method. This hypothesis is confirmed by the following observation: if the solvers use a tighter tolerance (e.g. 10^{-8}) the concentration profiles are described by a larger number of points and the recovery profiles obtained are more similar to those obtained with the Crank-Nicholson method, as shown in Fig. 4.5.

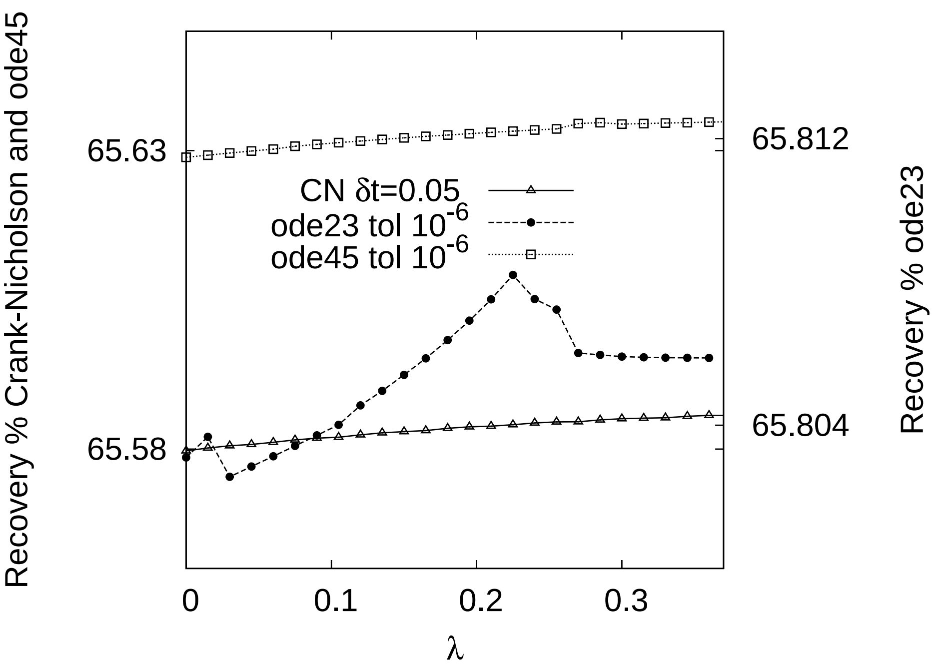


FIGURE 4.4: Recovery plots obtained by the Crank-Nicholson method with $\delta t = 0.05\,s$, and with the two adaptive time-step algorithms implemented by ode23 and ode45, with a tolerance of 10^{-6}

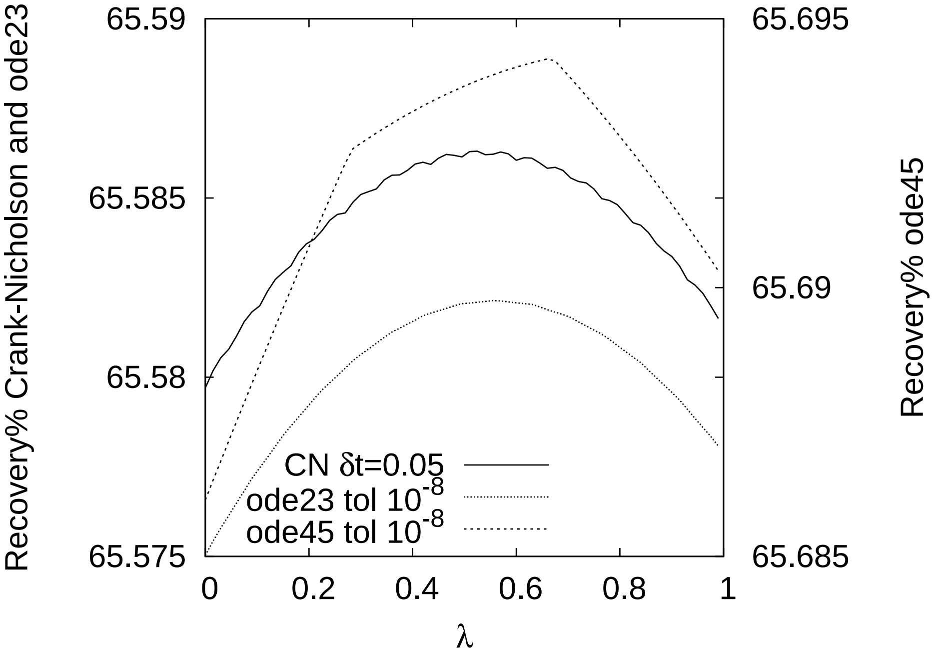


FIGURE 4.5: Comparison between recovery plot obtained with Crank-Nicholson and with the ode23 and ode45 solvers with a tolerance of 10^{-8}

The ode23 gave a better result than the ode45, which provided a segmented profile. On the other hand, the profile drawn by ode45 are more quantitatively similar to the profile drawn by the CN method. However, ode23 took around 3 hours to simulate the profiles in Fig. 4.5 and ode45 employed 1.5 hours. In both cases, simulations have been run on a 2.8GHz Pentium processor. This means that tightening the tolerance of the integrator improves the result, but increases the computational time so that no benefit is achieved with comparison to the fixed-step procedure.

Recovery calculated by the ODE integrator

The solution to the problem of the irregularity of the profiles calculated by the ODEs solvers has been found by letting the integrator (e.g. ode23 or ode45) calculate the integrals defining recovery and purity, thus allowing to have the profiles of recovery and purity which have the same accuracy of the concentration profiles.

The ode methods solve calculate of $y(t)$ by knowing the right hand side (RHS), $f(t)$, of the equation

$$\frac{dy(t)}{dt} = f(t) \tag{4.8}$$

For the ode solvers to evaluate the integrals necessary to evaluate the recovery, eq. 4.5, the terms "$F_{prod}\, y_{\mathrm{N}_2}$ and "$F_{in}\, y_{\mathrm{N}_{2,in}}$" have been added to the set of RHS equations provided. Thus, the simulations provides the total amount of N_2 obtained in the product and the total amount fed during the cycle. The ratio of these two quantities is the recovery.

The profile calculated by ode23 shows some irregular peaks, as shown in Fig. 4.6. This might be due to some instability of the numerical method at certain design points. Conversely, the profiles calculated by ode45 are smoother and coherent with those found with the previous strategy (CN method followed by the quadrature of the concentration profiles). Furthermore, the values of the recovery calculated by ode45 are similar to those calculated with the CN method with a very small δt, as shown in Fig. 4.7. A significant saving in computational time is achieved using ode45 rather than the CN method.

4.3 Direct Search Method for Optimal Design

The previous analysis showed that the objective function of the design problem is non-smooth. When dealing with an unconstrained optimisation problem of the form f :

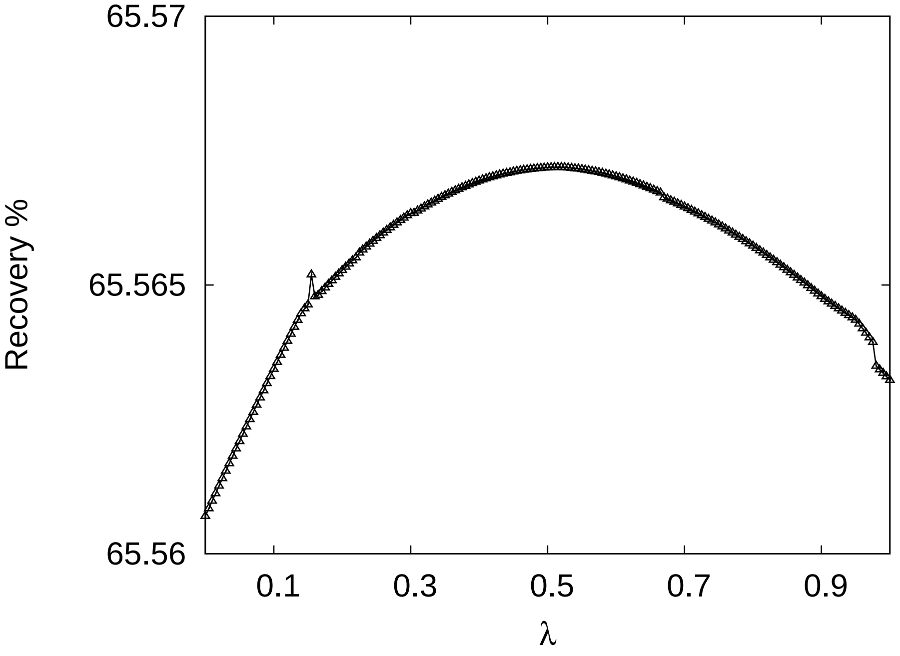


Figure 4.6: Recovery plot calculated by ode23

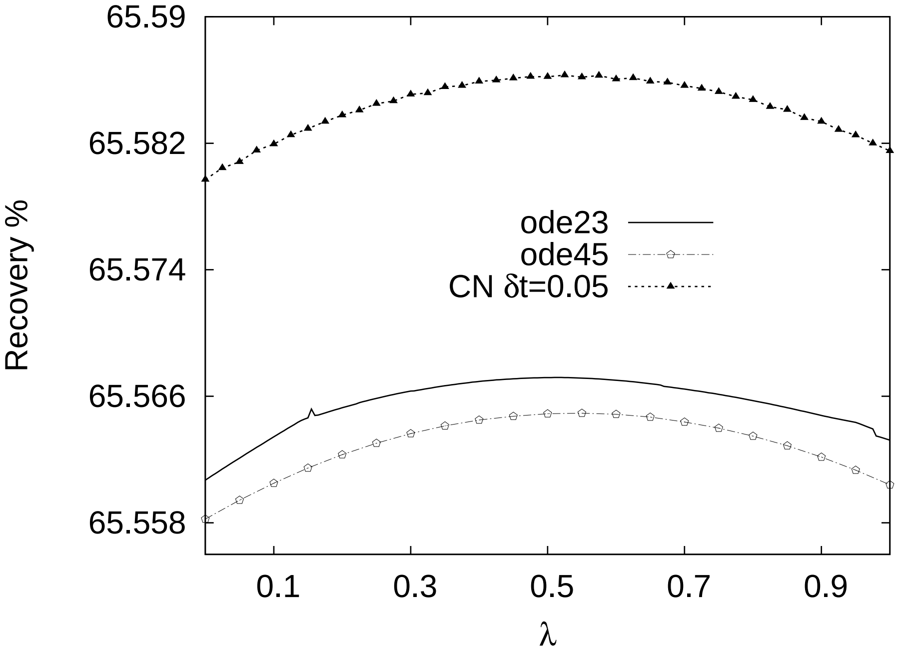


Figure 4.7: Recovery plot calculated by ode45

$\mathbb{R}^n \rightarrow \mathbb{R}$, the purpose is to find a point of minimum or maximum for this function. When the function f is either continuous but non-smooth, or not even continuous, at least in some points of $\mathbb{R}$, we must use an optimisation method which does not require information about the derivative of f, as it might not exist in some particular point. Direct Search Methods (DSMs) are based on optimisation algorithms which can handle the lack of smoothness of the objective function [113]. DSMs are derivative-free algorithms, which use exclusively function values in their attempt to find the optimum. A distinctive characteristic of DSMs is that at each iteration DSMs explore the objective function in a linearly independent set of n directions, as an alternative to information coming from the gradient [76]. If this set of directions has the right structure, it is possible to derive convergence results for these algorithms [128]. DSMs involve the comparison of each trial solution with the best trial solution, thus, they require the relative rank of the objective function values rather than the numerical function values, and they do not require any specification about "how much better" the next trial has to be.

Usually, three different types of DSMs are indicated [76]:

1. Simplex Methods;
2. Pattern search methods;
3. Methods with adaptive sets of search directions.

Torczon [128] proposed a different taxonomy, which uses as distinctive characteristics of each method both the way the set of n search directions is chosen and maintained and the way exploratory steps are taken in each of the n directions. The methods are collected in the following two categories:

- Methods for which the the set of search directions is modified at the end of each iteration;
- Methods for which the set of search directions remains fixed across all iterations.

DSMs are widely used even if for many of them there is no proof of their convergence properties. In fact, they often are successful in practice since many of them rely on techniques of classical analysis, even if in ways that are not always readily apparent from their original heuristic formulations [76]. With the exception of the simplex-based methods, DSMs are robust and satisfy the first-order necessary conditions for a minimiser (i.e. convergence to a stationary point) [128]. A second reason behind the actual success of these methods is that they succeed in some nonlinear optimisation problems where the more sophisticated quasi-Newton method fails [76]. Furthermore, they are easy to implement, so that using such methods can represent a good trade-off between efforts and results, especially in industrial contests where a fast answer is needed. A common use of these method is to provide a good starting point for the implementation of a more expensive analysis [48, 76].

DSMs are insensitive to the lack of smoothness in the objective function f. However, when applied to smooth problems, DSMs have the main drawbacks of showing at best a linear rate of convergence and they are unable to determine the nature of the point at which they terminate, since no derivative information is available.

4.3.1 Simplex Methods

The concept behind simplex method is to use a non-degenerate simplex to drive the search of the optimum. A simplex is defined as $n+1$ points in a $\mathbb{R}^n$ space. A non-degenerate simplex is one for which the set of edges adjacent to any vertex in the simplex forms a basis for the space, which means that any point in the domain of the search can be determined as a linear combination of the edges adjacent to any given vertex. This allows to use $n+1$ objective evaluations to get an improvement on the iterate, rather than the $2n$ or 2^n needed by other DSMs. The method uses the simplex to sample the space, and exploits the fact that by reflecting a vertex through the centroid of the opposite face, we obtain a new simplex.

The basic heuristic of the model is the following: we reflect the worse vertex (the one with the less desirable objective function) in the direction of the other vertices, eventually finding a best vertex. When the reflected vertex gives a significant decrease/increase in the value of the objective function, it can be concluded that a candidate for a minimiser/maximiser has been found.

The common structure for all the simplex based DSMs, as described by Singer and Singer [113], is the algorithm 4.1.

The algorithm **INIT** constructs the initial simplex $S_0 = S(x_0, ...x_n)$ around or near the initial point x_0 and computes the function values at all vertices. The most frequent choice is a S_0 right-angled in x_0, based on the coordinate axes, or $x_j = x_0 + h_j e_j$, where $j = 0, ...n$, with step sizes h_j in directions of unit vectors e_j in $\mathbb{R}^n$. S_{init} can also be a regular simplex, with all the edges having the same length [113]. The inner loop algorithm **TRANSF** determines the type of the simplex based DSMs used.

Algorithm 4.1 Generic Scheme of a Simplex DSM

```
Input: INIT: construct an initial working simplex $S_0$
Output: $x_{opt}$= Optimum
  $S \leftarrow S_0$
  Repeat
    Calculate termination test information;
    if The termination test is not satisfied then
      TRANSF: Transform the working simplex
      $S \leftarrow TRANS(S)$
    end if
  Until Termination test is satisfied
Output: $x_{opt}$:=the best vertex of the current simplex $S$;
```

One or both the following criteria need to be satisfied to prove the convergence of a simplex-based DSMs [71]:

1. the edges of the simplex remains uniformly linearly independent at every iteration;
2. a descent condition stronger than simple decrease is satisfied at every iteration;

4.3.1.1 The Nelder-Mead Simplex Method

The Nelder-Mead (NM) Simplex method [88] is a simplex direct search method for multidimensional unconstrained minimisation of non-smooth functions. The NM method has been implemented in many different ways, the main differences lying in the way the iteration process is broken, i.e. the choice of the convergence (or termination) test [114]. The contribution of Nelder and Mead to simplex optimisation methods was the introduction of some additional moves to the simplex search, which made the search faster and turned the heuristic rule into an optimisation algorithm. In particular, they supplemented the basic reflection move with additional moves which allowed a better and faster fit of the objective function: expansions and contractions.

A detailed explanation of how the NM algorithm works can be found in [60, 113].

In a n dimensional design space, the operation that the NM algorithm apply to the simplex can be described as follows [113]:

1. determine indices h, s, l of the worst, the second worst and the best point, respectively:

$$f_h = \max_{j} {}_j; \quad f_s = \max_{j\neq h} f_j; \quad f_l = \min_{j\neq h} f_j$$

2. calculate the centroid $\overline{x}$ of the $n-1$ hyperplane defined by all the vertices of the simplex but the one corresponding to the worst point, given by

$$\overline{x} = \frac{1}{N}\sum_{i=1}^{n} x_i$$

3. compute the new working simplex from the old one. First, try to replace the worst point with a better point x_{new}, by using reflection, expansion or contraction (inner or outside contraction), according to

$$x_i^{\text{new}} = (1+\mu)\bar{x} - \mu x_h \tag{4.9}$$

   the value of the parameter μ determine how the simplex changes. If this fails, shrink the simplex towards the best point x_l, according to:

$$x_i^{\text{new}} = x_l - \mu(x_i^{\text{old}} - x_l) \quad \forall i = 2, \ldots, n \tag{4.10}$$

   where “new” and “old” refers to the new and to the old simplices, respectively.

The transformations the initial simplex can go through are shown in Fig. 4.8.

A termination procedure is needed for the algorithm to converge in a finite number of steps. One of the following conditions is used as different termination test: the algorithms stops [113, 114]

1. when the working simplex is small in some sense (i.e. when all the vertices of the simplex are close enough);
2. when some or all the function values f_j are closed enough in some sense;
3. when the maximum number of transformations allowed is reached.

A general lack of convergence for the NM method has been observed [80, 131]. This derives from the fact that the method does not fulfill any of the requirements discussed earlier. In [114], it is stressed that numerical evidence suggests that the NM algorithm becomes inefficient as the dimension of the search space increases. More often the inefficiency of the NM algorithm is experienced when the objective function is discontinuous. Furthermore, compared with other DSMs, the NM algorithm requires a lower number of function evaluation to detect a new simplex, thus it has been widely used despite it often breaks or fails.

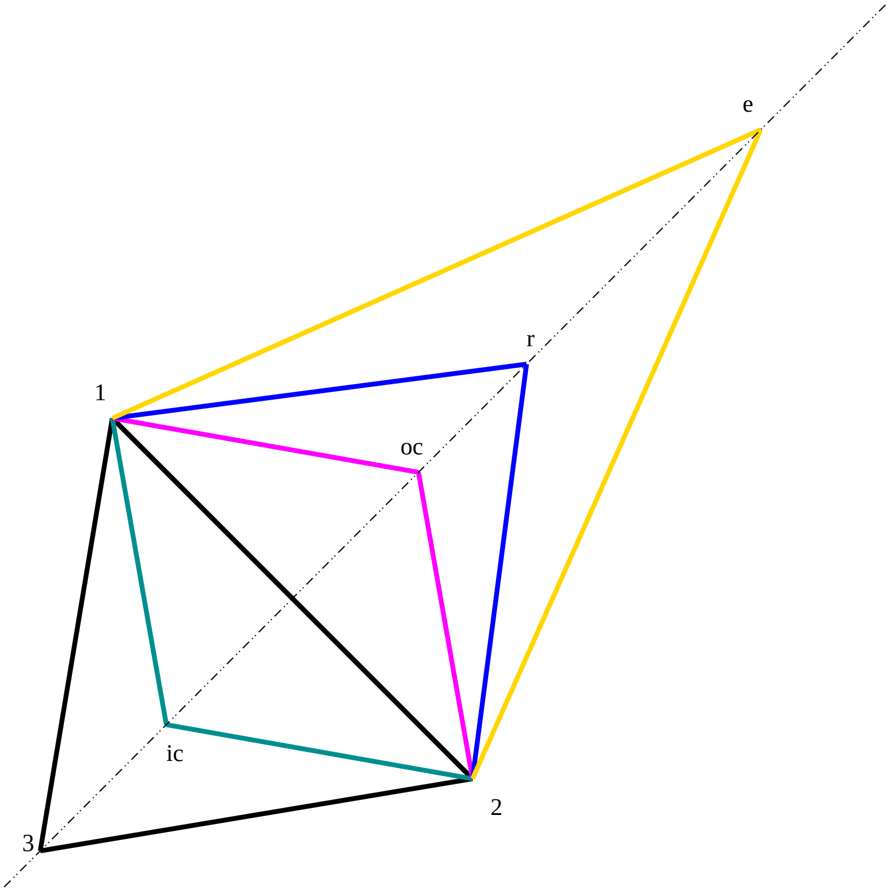


FIGURE 4.8: Example of how a simplex (1-2-3) is transformed by the NM algorithm, for a 2 dimensional optimisation problem. First, the vertices of initial simplex S are ranked, and ordered. Vertex 3 is the worst vertex. The algorithm tries to replace the worst vertex 3 with a better one, by reflection (simplex 1-2-r), expansion (simplex 1-2-s), inner contraction (simplex 1-2-ic), outer contraction (simplex 1-2-oc).

4.3.2 Pattern Search Methods

Pattern search methods consist of a series of “exploratory moves” (objective function evaluations) at the points describing a regular lattice within the domain [76]. These methods maintain uniform linear independence of the simplex edges and require only simple decrease in the best function value at each iteration [71]. The exploratory moves consist of a systematic strategy to visit the points in the lattice in the immediate vicinity of the current iterate. Some characteristic features of this methods are:

- no mathematical formulation of a model for the objective function is required;
- the evaluations are made at the points of rational patterns, since the magnitude of the steps is predetermined;

- the step size of the simplex is decreased only when no increase or decrease in any of the design variables further improve the objective function.

The global convergence of these methods has been proved under different conditions [76].

4.3.2.1 The Multidirectional search method

The multidirectional search algorithm is a direct search method designed for unconstrained optimisation. A proof of convergence of this algorithm has been provided by Torczon [128].

The multidirectional search is a pattern search method which proceeds by reflecting a simplex. The method uses three trial steps: the reflection step, the expansion step and the contraction step. The algorithm always computes a reflection, then if a new best vertex has been found, an expansion step is computed; otherwise automatically a contraction step is carried out. The procedure is illustrated in Fig. 4.9.

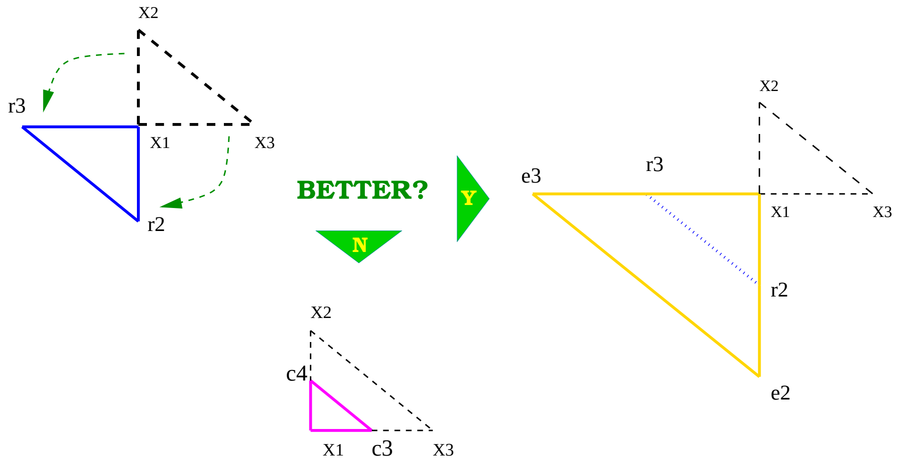


FIGURE 4.9: The multidirectional search method uses three trial steps: the reflection step, the expansion step and the contraction step. The algorithm always computes a reflection, then if a new best vertex has been found, an expansion step is computed; otherwise automatically a contraction step is carried out

The multidirectional search method requires at least $2n$ independent function evaluations per iteration.

4.3.2.2 Alternating Directions Method

The description of the method is in [48]. Let us consider the following optimisation problem:

$$\max_{x \in \mathbb{R}^n} f(x),$$

$$f : \mathbb{R}^n \rightarrow \mathbb{R}$$

Given a starting value x, the method attempts to solve the previous problem by repeatedly maximising over each coordinate direction in turn, as illustrated in algorithm 4.2:

Algorithm 4.2 Alternating Direction Method

```
Input: Starting value $x$
  Repeat {One iteration comprises a loop over all component of $x$}
    for $i = 1$ to $n$ do
      find $\alpha$ such that $f(x + \alpha e_i)$ is maximised; {line search}
      set $x \leftarrow x + \alpha e_i$;
    end for
  Until converged
```

The alternating search method is one of the simplest DSMs and its main weakness is that it ignores any interactions between variables.

4.3.3 Methods with adaptive sets of search directions

This category is related to methods which collect information about the curvature of the objective function during their search and use it to accelerate the search.

4.3.4 Implicit Filtering

Implicit Filtering is an optimisation technique specifically designed to optimise noisy objective functions [61]: the algorithm “filters” high-frequency low-amplitude oscillations by using large difference increment. A noisy function could be modelled as the perturbation of a smooth function f_s by a small function ϕ according to

$$f(x) = f_s(x) + \phi(x) \tag{4.11}$$

However, Implicit Filtering has no global convergence property, hence it works efficiently with functions whose large-scale behaviour, represented by f_s, can be easily optimised. Kelley [61] introduces Implicit Filtering as an extension of the Coordinate Search method.

Let us suppose we want to find the minimum of a function $f(x)$, where $x \in \mathbb{R}^N$. The Coordinate Search method takes into account a possible approximation to the optimum, x, to start its search. It evaluates f at the $2N$ points described by the *stencil* S, defined as

$$S(x,h) = x \pm he_i \tag{4.12}$$

where h is the pattern size, or *scale*, and e_i is the unit vector in the i-th coordinate direction. Let $z^* \in S(,x,h)$ be a point which minimises f, i.e. $f(z^*) = \min\limits_{z \in S(,x,h)} f(z)$. If $f(z^*) < f(x)$, then x is replaced by z^* and the previous steps are implemented again. If $f(z^*) \geq f(x)$, then the previous steps are implemented again, but with a smaller value of h. The second case, $f(z^*) \geq f(x)$, is referred to as a *stencil failure*, as the stencil has failed to improve the objective function value.

The search could be stopped after a certain number of iterations or when h has reached the minimum allowed value.

Similarly, the Implicit Filtering evaluates f at the along a stencil $S(x,h)$ and reduces the size of the stencil if [61]:

1. stencil failure occur;
2. $\|\nabla_h f(x)\| < \tau h$, for some $\tau > 0$, where $\nabla_h f(x)$ is the central different gradient.
3. the line search fails to satisfy the sufficient decrease condition: no λ can be found so that the following condition holds

   $$f(x - \lambda \nabla_h f(x)) < -\alpha\lambda \|\nabla_h f(x)\|^2 \quad \text{for some}\, \alpha > 0 \tag{4.13}$$

   for some $\alpha > 0$. The value of λ is looked for via backtracking the linesearch a maximum number of times $a_{\max}$, i.e. looking for an integer number $m \in 0, \ldots, a_{\max}$ so that eq. 4.13 holds for $\lambda = \beta^m$.

If none of the above termination criterion is satisfied, the search is continued using $x = x - \lambda \nabla_h f(x)$ as a starting point. The values of α, τ and β are parameters of the algorithm. Proof of the convergence of the Implicit Filtering algorithm has been provided [60, 61].

4.3.5 The Hooke-Jeeves Algorithm

As the Implicit Filtering method, the Hooke-Jeeves algorithm evaluate the objective function on a stencil, and compute the direction of the search based on the values found [60].

Input of the Hooke-Jeeves algorithm are a starting point, x, and a pattern size h. The objective function f is evaluated at the vertices of the stencil built around x. Let z^* be the point of the stencil which minimises f, i.e. $f(z^*) = \min_{z \in S(,x,h)} f(z)$. If a stencil failure occurs, $f(z^*) \geq x$, the process is started over using x as a center and a smaller h. Instead, If $f(z^*) < f(x)$, then a new direction search is defined as $d^{HJ} = z^* - x$, and at the new iteration the search is initialised from a point x_c defined as

$$x_c = x + 2d^{HJ} = z^* + d^{HJ}$$

If the stencil built on x_c does not improve upon $f(z^*)$, then z^* is used as the center of the next stencil. Should this new stencil fail, then h is reduced and the process is started over from z^*.

4.4 Performance of the optimisers

For the solution of the design problem stated in eq. 4.4 the following DSM methods have been considered:

- Nelder-Mead Algorithm;
- Multidirectional Search;
- Alternating Directions;
- Implicit Filtering;
- Hooke-Jeeves Method.

Ten random starting points within the feasible region of the design problem have been used to compare the performance of the different DSMs. Starting from these points, each optimiser found ten different solutions. The performance of the different optimisers has been evaluated and compared according to the following criteria:

1. higher value of the recovery found (Best);

2. average value of the recovery found (Average);
3. dispersion of the optima found, expressed by the standard deviation (Std Dev);
4. average number of function evaluations needed to detect an optimum (Eval).

Table 4.1 shows the performance of Nelder-Mead (with right and a regular-angled simplex), alternating directions and multidirectional search methods. For these evaluation the Crank-Nicholson method has been used for the simulations.

TABLE 4.1: Performance of the optimisers

Algorithm	Recovery (%)			Eval
	Best	Average	Std Dev	
NM right-angled simplex	65.84	**64.22**	**2.14**	66
NM regular-angled simplex	65.22	61.78	3.57	**61**
Alternating directions	65.99	63.72	2.43	82
Multidirectional search	**66.91**	62.51	5.49	164

All the optimiser showed an average optimal recovery (expressed in percentage) bounded between 61% and 64.5%.

Even if the best absolute value has been found by the multidirectional search method (recovery= 66.91%), the more reliable and efficient optimiser is the NM search method, using a right-angle method. This showed the best average result as well as the lowest deviation standard among all the methods. In term of number of objective function evaluations, the most expensive method is the multidirectional search method, followed by the alternating direction search, which needed half the evaluations required by the former method. The two variants of the NM methods have been the more efficient, the one with the regular simplex being the best one. However, finding 10 solution took approximately 12 hours.

Explanations to the better performance of the right-angled NM method, compared to the regular simplex one, have been looked for, but the topic has not been investigated in the literature. According to the description of the two algorithms provided by Octave-Forge, it is suggested that the NM right-angled usually shows better results, but no reason is provided. Singer and Singer [114] stated that usually the initial simplex is chosen to be a right-angle one, but in some implementation it may be a regular simplex, having all the edges of the same length. However, no comments is added which can help us to say whether one is better than the other. Our hypotheses are:

- the condition to have a right-angled simplex can avoid the possibility of getting a degenerated simplex;
- the introduction of a fixed "shape" for the simplex might force a regularity in the pattern constituted by the points the simplex explores, thus enhancing the convergence of the algorithm (it becomes somehow more similar to the multidirectional algorithm).

For the design of the operation, the important information is the area of the design space where the optima are located. The optima having a recovery value of $Recovery = 65.8 \pm 0.5\%$ have been plotted in Fig. 4.10. The value of the split ratio for such points was found to be $r_s = 0.98 \pm 0.01$. The position of the optima, in a plane inlet feed rate (F_{in}) vs cycle time (t_c) are located as illustrated in Fig. 4.10. The graph shows that to achieve the same value of the recovery the inlet feed rate needs to decrease as the cycle time increases. This is due to the fact that we imposed an upper boundary to the value of the maximum pressure, P_{max}, that we want to achieve: since we are pressurising by feeding the reactor while no output is allowed, as F_{in} increases the sooner the value of P_{max} will be reached.

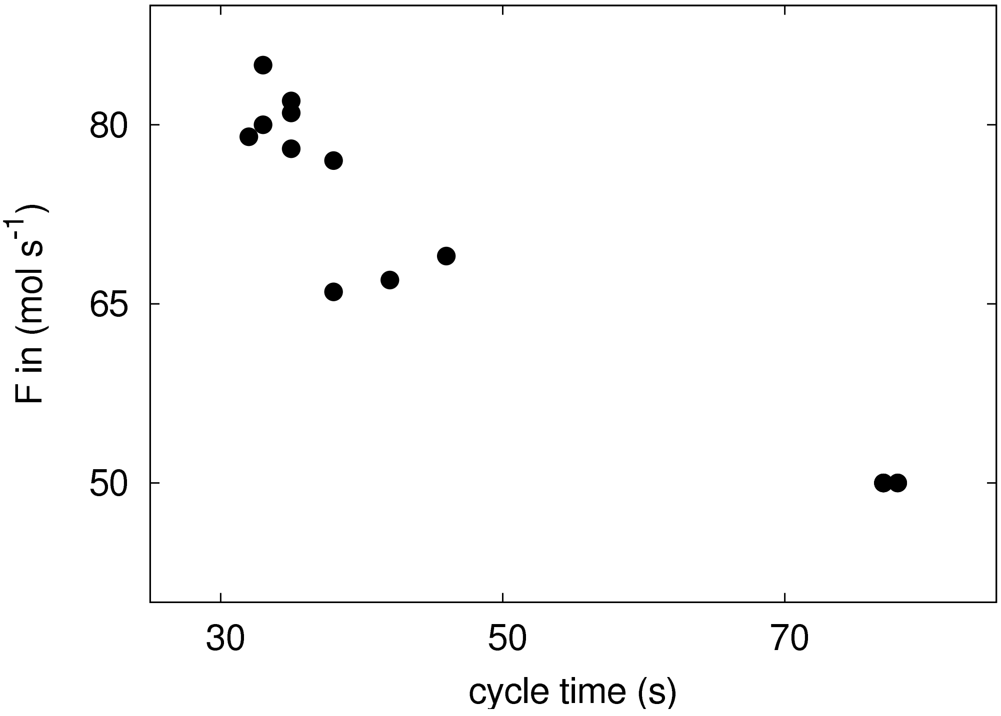


FIGURE 4.10: Optima on a F_{in}vs t_c plane, with recovery=65.8% and r_s=0.98.

Results of the optimiser using ode45

The introduction of a time adaptive solver, using ode45, allowed us to have a more smooth objective function, and to decrease the computational time required. The recovery must be calculated by the integrator itself to improve the smoothness of the recovery profiles. The aim of this section is to investigate which advantages could be brought to the solution of the design problem of eq. 4.4 if ode45 is used within the simulation. To this end, I repeated the exploration starting from the same 10 randoms points used to generate the results in Table 4.2, when the simulation is carried out by the integrator ode45. The optimiser used for this evaluation is the most successful and robust optimiser detected: the Nelder-Mead optimiser using a right-angled simplex.

In the NM algorithm, the simplex shrinks when it is around an optimum, and the iterations stop when the simplex size has reached the minimum size allowed. The effect of the minimum simplex size allowed on the performance of the algorithm has been investigated.

Table 4.2: Performance of the NM method using a right-angled simplex and ode45 as solver. The effect of the minimum size of the simplex allowed is investigated.

Simplex min size	Recovery (%)			Eval	Time (min)
	Best	Average	Std Dev		
10^{-3}	65.84	64.22	2.14	66	15
10^{-4}	66.02	65.10	1.75	90	22
10^{-5}	66.07	65.13	1.77	116	42
10^{-6}	66.07	65.14	1.76	140	54

The optima obtained running the simulation with ode45 do not differ from those obtained with the previous fixed-time step method (see Table 4.1 and 4.2 with minimum simple size 10^{-3}). There is some improvement in the value of the recovery when decreasing the minimum simple size from 10^{-3} to 10^{-4}, but such improvement is no longer appreciable when diminishing the simplex size further; besides, the number of evaluation as well as computational time increase.

Using ode45 for the simulation, the performance of the Hooke and Jeeves and the implicit filter optimisers have been investigated. The same starting points have been used for these optimisations. The result is showed in Table 4.3.

As shown in Table 4.3, the two optimisers performed worse than those seen before: the average value of the recovery is lower then for the all the other methods. However,

Table 4.3: Performance of Hooke-Jeeves and Implicit Filtering method using ode45

Algorithm	Simplex min size	Recovery (%)			Eval	Time (min)
		Best	Average	Std Dev		
Hooke-Jeeves	10^{-6}	63.70	55.63	5.35	125	84
Hooke-Jeeves	10^{-8}	64.81	60.70	3.44	243	180
Implicit Filtering	10^{-4}	65.81	63.62	2.30	152	22

a strange feature of the optima found by Hooke and Jeeves algorithm is that all the points have almost the same value of inlet feed rate and cycle time, respectively $F_{in} = 50 \pm 0.17\,mol\,s^{-1}$, $t_c = 78.39 \pm 0.07s$. The value of the split ratio found is much lower than that detected by previous method, having an average value of 0.8.

DSMs tend to detect local optima. However, further progress can be made by restarting the optimisation process from the same maximiser. I tried to improve the results obtained previously by combining the more robust method, Nelder-Mead using a right-angled simplex, and the multidirectional method which previously found the best absolute value of the recovery (see Table 4.1). In particular, the multidirectional search method was started from the optima found by the Nelder-Mead algorithm. The results obtained are shown in Table 4.4.

Table 4.4: Results obtained by the multidirectional search method using as starting points the optima found by the Nelder-Mead algorithm.

Algorithm	tol	Recovery (%)			Eval
		Best	Average	Std Dev	
MDS after NM	10^{-6}	67.56	66.23	1.62	281

In this case, a same optimum was detected in five out of ten optimisation runs. This optimum is located at $F_{in} = 50 \pm 0.0006\,mol/s^{-1}$, $r_s = 0.9999$, and $t_c = 78.39 \pm 0.0002\,s$. Some considerations are needed. This point has the same coordinates as those found by Hooke-Jeeves in terms of t_c and F_{in}, but with a much higher value of the r_s. Evidently, the Hooke-Jeeves method was able to tune till the optimal condition F_{in} and t_c, but it was not able to detect the optimal value of r_s. Conversely, almost all the other methods found the best value of the split ratio, pushing it to the value of one (the upper bound). The value of the split ratio found is not realistic since it approaches the value 1, which means the elimination of the purge step. The value of the inlet feed rate found is the minimum allowed in the search space: the optima are at the maximum cycle time allowed by the pressure upper boundary (the pressure at the optima is almost P_{max}).

I combined the Nelder-Mead method (able to push through the best value of the split ratio), and the Hooke & Jeeves algorithm, able to find the best value of F_{in} and t_c. The combination of these two methods did not improve the results.

4.5 Conclusion

This chapter deals with the maximisation of the recovery of N_2 obtained from a 2-bed/4-steps Skarstrom cycle. The design variables chosen are the cycle time, the split ratio and the inlet flowrate. The constraints are a maximum value of the pressure and the evaluation of the model at cyclic steady state (CSS). In the simulation, the CSS has been detected via successive substitution.

To choose an appropriate optimisation method, the profile of the objective function has been characterised. Using as integration method Crank-Nicholson, with a fixed time step, the profile of the objective function is non-smooth. Direct Search methods have been detected as the best optimisation methods for non-smooth objective functions, and the performance of some of these method has been investigated. Among the optimisation algorithms used, the Nelder- Mead method, with a right-angled simplex, gave the more robust and efficient performance. However, the detection of 10 optima took approximately 12 hours.

Time-adaptive integration schemes have been introduced to reduce the time of the simulation: ode23 and ode45, both base on the Runge-Kutta method. It has been found that to get the same accuracy in the objective function profile, the values of the recovery has to be calculated by the integrator itself, and it cannot be calculated from the concentration profiles by a quadrature scheme. Using ode45 and the NM algorithm, the same optima were found, but in much less time: finding 10 optima took approximately 15 minutes.

Each optimiser found optima quite distributed in the search space, having similar values of the recovery, but with a significant standard deviation. Applying a hybrid optimisation technique (combining the Nelder-Mead and the multidirectional search method) the same optimum was found half of the times, starting from 10 different points randomly chosen in the search space. The combination of these two methods has been the most successful and robust optimisation strategy detected.

DSMs proved to be reliable method for the single objective optimisation of the PSA process. However, as the recovery is increased, the purity of N_2 in the product stream decreases. Hence, it is necessary to take into account the effect of the design variables on the conflicting criteria to properly design the operation. The multi-objective optimisation of the process is carried out in the following chapter.

Chapter 5

Multi-objective Optimisation of PSA cycles

The performance of PSA cycles is usually characterised by conflicting aspects of the performance: the higher the recovery, the lower the purity of the product; the higher the purity, the higher the costs of the operation [105, 109]. These trade-offs should be taken into account during the design of the process. However, only two previous studies [63, 109]adopted a multi-criteria approach for PSA cycle optimisation and the methodologies developed were not completely satisfactory. Hence, we decided to address the paucity of options for PSA multi-objective optimisation by investigating an appropriate strategy to this problem.

Multi-objective optimisation is the simultaneous optimisation of more that one objective function. The result of a multi-objective optimisation in most cases is a set of equally good solutions, known as the Pareto set. The Pareto set can then be used to make some decision about the preferred possible solution. However, there are also multi-objective methods which do not result in a Pareto set (e. g. weighted sum; lexicographic etc.).

For single objective optimisation, the results obtained using DSMs were promising (see Chapter 4). However, for the multi-objective design problem, the use of single objective DSMs is problematical due to the need for a weighted combination for the objective function. As the number of objectives to be considered simultaneously increases, the number of discrete optimisation problems that must be solved using a weighted objective function increases exponentially.

An alternative to direct search or gradient based methods is the class of evolutionary stochastic optimisation algorithms (EAs). These methods are similar to DSMs in that they do not require gradient information. They are, however, able to generate a Pareto front directly. Another advantage is that EAs can be easily implemented in parallel [20, 123]: this could be an important property if a reduction in computational time is desirable, as in the case of PSA design.

Genetic algorithms have been extensively applied for the multi-objective optimisation of simulated moving beds (SMBs), a continuous countercurrent chromatographic separation processes [134, 137]. SMBs processes have similar complexity to a PSA process, described by PDAEs with comparable nonlinear behaviour. Therefore, we consider a genetic algorithm for tackling the multi-objective design problem.

A theoretical background on evolutionary algorithms, of which genetic algorithms are a subclass, is given in section 5.1. A simple multi-objective genetic algorithm has been implemented in this work and it is illustrated in section 5.2. Although many well established genetic algorithms are readily available to use, it has been preferred to implement a “in–house” algorithm so as to help the reader, as well as the author of this thesis who is not an expert in computational intelligence, to gain a better understanding of these techniques. Another advantage of implementing an “in–house” algorithm is that it allows a complete control on the parameters which regulate the optimisation process, as explained in section 5.2. The performance of the algorithm is analysed in section 5.4, and assessed against DSMs in section 5.4.1 for the optimisation of a Skarstrom cycle for the production of N_2 from air.

The multi-objective optimisation of a 4-bed/4-step VSA cycle for the separation of CO_2 from flue gases is presented in section 5.5. For this case study, the focus has been into replicating and potentially improving literature results, hence the section on CO_2 starts by comparing the results achieved by our model and the results described for the same system in literature.

The aim of the multi-objective optimisation is not only to find the optimal trade-offs between the objectives, but also to investigate the effect of the design variables on the objectives so as to gain a better understanding of the separation process.

5.1 Evolutionary Multi-objective Optimisation

Evolutionary algorithms possess characteristics suitable for the solution of optimisation problems coping with conflicting objectives, and therefore they have been used for multi-objective optimisation problems in the last decade [139]. The term “evolutionary algorithm” (EA) indicates a class of stochastic optimisation methods that simulate the process of natural evolution. Evolutionary theory has been applied to engineering processes since the mid 1950s, pioneered by the work of George Box [93]. The concept has been further investigated in the '70s when genetic algorithms, evolutionary programming and evolution strategies have been developed [139]. Starting from a given set of candidate solutions, another candidate set is generated by modifying it according to two criteria: selection (competition among the points) and variation (recombination and mutation of the points). EAs are mostly effective when dealing with multiple conflicting objectives and/or with intractably large and highly complex search spaces.

Suppose we have an optimisation problem with $k > 1$ objective functions, all to be maximised and all equally important. The function to be optimised is $f : X \rightarrow Y$, where $X \in \mathbb{R}^n$ is the "decision space", and $Y \in \mathbb{R}^k$ is the "objective space". The solution to the problem can be described by a *decision vector* $\overline{x} = [x_1, x_2, ..., x_n] \in X$. According to the concept of Pareto dominance, in a maximisation problem an objective vector $\overline{y}_1$ is said to *dominate* another objective vector $\overline{y}_2$ if no component of $\overline{y}_1$ is smaller than the corresponding component of $\overline{y}_2$, and at least one component is greater. There might be several optimal (dominating) objective vectors representing different trade-offs between the objectives. The decision vectors corresponding to these optima are called *Pareto set* $X^* \subseteq X$, and their image in objective space is called *Pareto front* $Y^* = f(X^*) \subseteq Y$.

The outcome of multi-objective evolutionary optimisation is a set of mutually non dominated solutions, or *Pareto set approximation.*

5.1.1 Evolutionary computation

The detection of the Pareto set can be very time consuming, and might even not be feasible. Many stochastic methods have been proposed to overcome this limit and detect a good approximation possible to the Pareto front: evolutionary algorithms, tabu search, simulated annealing and ant colony optimisation [139].

An **"evolutionary algorithm"** is characterised by the following steps [139]:

- a set of candidates is maintained;
- a selection process is performed on such set (fitness assignment and sampling);
- several solutions might be combined to generate new solutions (recombination and mutation);
- an environmental selection process is carried out to determine which solutions, among the old and the new ones, are kept in the set of candidate solutions.

The solution candidates are called "individuals", while the set of solution candidates is called "population". The individuals of the population are often indicated to as *chromosomes*, and are defined by a set of parameters (or *genes*) which define a proposed solution to the problem that the genetic algorithm is trying to solve. In a design problem, a chromosome is a specific design points, and the genes that define it are the corresponding values of the design variables. A chromosome is often represented as a simple string, although a wide variety of other data structures are also used.

As described by Zitzler *et al.* [139], in the first step of the selection process, the fitness assignment, a scalar value is assigned to each individual of the current population, which reflects its quality according to its position in the objective space. In the sampling step, a "mating pool" is created by randomly sampling the population according to the fitness values. One possible and very common method for sampling is the "binary tournament selection": two individuals within the population are randomly chosen, and the one with the better fitness value is sent to the mating pool. The process is repeated until the mating pool is filled. The size of the tournament can be higher than two.

The variation operators are applied to the mating pool. There are two methods for the variation: recombination and mutation. The "recombination" or "mating" operator takes a certain number of individuals (parents) from the mating pool and generates a predefined number of children by combining parts of the parents. The crossover probability is associated with this operator to mimic the stochastic nature of evolution. Conversely, the "mutation" operator modifies individuals by changing small parts in the associated vectors according to a given mutation rate (e.g. $(x_1, x_2, x_3, x_4) \rightarrow (x_1, x_2^{'}, x_3, x_4)$). Due to the randomness of the process, some individuals in the mate pool might not be affected by any variation.

The final step is the environmental selection which determines which individuals of the old population and of the mating pool form the new population. Such selection can be made by simply using the mating pool as new population. Otherwise it is possible, for example, to combine both sets and deterministically choose the best individuals (the fitter) for survival.

The above loop is repeated a number of times. Each loop iteration is called a "generation". A common criterion for termination is to assign a predefined maximum number of generations to investigate. Alternative stopping criteria might be stagnation in the population or the existence of an individual with sufficient quality.

The best individuals of the last population are the outcome of the method, i.e. the approximation found to the Pareto set. The goals which have to be achieved while implementing the multi-objective evolutionary algorithm (MOEA) are to guide the search towards the Pareto set and to keep a diverse set of non-dominated solutions. The approximation to the Pareto set is made by the choice of a proper selection method and of fitness values assignment. The maintainance of the diversity in the population generated concerns the selection in general: we want to avoid that the population contains almost identical solutions with respect to the objective and decision spaces. A third issue linked with the implementation of the algorithm is *elitism*, which concerns the question of how to prevent the loss of non-dominated solutions.

5.1.2 Fitness assignment

In a single-objective optimisation problem, the "fitness" of each point can directly coincide with the value of the objective function itself. This is not the case for multi-objective optimisation problems, where both fitness assignment and selection must take into account the multi-criteria nature of the problem. There are three main categories of fitness assignment methods: aggregation based, criterion-based and Pareto-based [139].

The *aggregation method* consists of aggregating the objective functions into a parametrised single-objective function: the parameters are systematically varied to find a set of non-dominated solutions instead of a single trade-off solution. An example of this method is MOEAs using weighted-sum aggregation, where the weights represent the parameters which are changed during the evolution process.

The *Criterion-based* method uses alternatively only one of the the objectives during the selection phase to decide which member of the population will be a parent for the next generation.

The *Pareto-based* method uses the Pareto dominance to assign a value of "fitness" to the individuals of the population. Three Pareto-based approaches to assign the fitness are suggested by Zitzler *et al.* [139]:

- dominance rank: according to the number of individuals by which an individual is dominated;
- dominance depth: the population is divided into several fronts and the depth reflects to which front the individual belongs to;
- dominance count: number of individuals that the individual dominates.

In the aggregation method the fitness value assigned to each individual is independent of other individuals, while in the other two methods such value is related to the whole population.

5.1.3 Genetic Operators

Each genetic algorithm is characterized by the mechanisms it uses to evolve the population of solutions from generation to generation. The most common operators used by genetic algorithms are selection, crossover and mutation. An brief overview of these operators, and their most common implementations, is provided in this section.

Selection The selection operator is responsible of choosing the individuals which will be used as parents of the individuals of the next population via the mating and mutating procedures. The selection is implemented so that the fitter the individual, the higher the probability of its participation in the breeding process. The selection procedure allows to *exploit* the genetic material contained solutions found at the previous generation; however, precious information are not only contained in the fittest individuals and so, to avoid stagnation, also less fit individuals should be given a chance to participate to the generation of new individuals.

One of the less sophisticated selection methods is the *truncation selection*: the candidate solutions are ordered by fitness, and a certain proportion p, (e.g. p=1/2, 1/3, etc.), of the fittest individuals are selected and reproduced $1/p$ times [83]. This procedure does not capitalise on the genetic material contained in the less fit individuals.

A more sophisticated technique is the *tournament selection*, where a number k of individuals are randomly selected to compete in a tournament won by the fittest. The higher the size of the tournament k, the smaller the chance of weakest individuals to be selected. This allows to change the selection pressure in different domains or at different stages of the evolution [84]. Furthermore, it is easy to code and it is efficient for both non-parallel and parallel implementations [84].

Another selection method is the so called *roulette-wheel selection* [93], or *fitness proportionate selection*: the probability of an individual to be selected is proportional to its fitness level. The selection process is similar to a roulette wheel, where the amount of space allocated to each individual depends on their relative fitness. The fitnesses values are normalised and a random selection of numbers from the range found determine which individuals are selected. Some individuals could be selected more than once as the process is repeated until the mating pool has been filled. Although the procedure favours the selection of the fittest individuals in the population, there is always a probability of the weakest individuals to be selected as well.

Despite being more sophisticated than the tournament selection, the latter is often preferred because it can be implemented more easily, has less stochastic noise and has a constant, and adjustable, selection pressure [84].

Crossover The crossover operator allows to share information between individuals of the population: it usually combines the features of two individuals to generate two offspring. The combination might allow to generate better individuals. Hence, the crossover is the operator that exploit the genetic material contained in the population. Different crossover mechanisms have been proposed in literature, and a review of the different approaches is proposed by Herrera *et al.* [46].

Let us suppose that the crossover operator has two be applied to two individuals $C_1 = (c_1^1, \dots, c_m^1)$ and $C_2 = (c_1^2, \dots, c_m^2)$, where m is the number of genes of each chromosome. The *Simple crossover* operator randomly selects a position $i \in 1, \dots, m-1$ and builds two new chromosomes as follows:

$$H_1 = (c_1^1, c_2^1, \dots, c_i^1, c_{i+1}^2, \dots, c_m^2)$$
$$H_2 = (c_1^2, c_2^2, \dots, c_i^2, c_{i+1}^1, \dots, c_m^1)$$

Another form of crossover is the *two-point crossover.* In this case two positions $i, j \in 1, 2, \dots, m-1$, with $i < j$, and the segments of the parents defined by i and j are exchanged:

$$H_1 = (c_1^1, c_2^1, \dots, c_i^2, c_{i+1}^2, \dots, c_j^2, c_{j+1}^1, \dots, c_m^1)$$
$$H_2 = (c_1^2, c_2^2, \dots, c_i^1, c_{i+1}^1, \dots, c_j^1, c_{j+1}^2, \dots, c_m^2)$$

In the *uniform crossover* two offspring are created, $H_k = (h_1^k, h_2^k, \dots, h_m^k)$, for $k = 1, 2$. The value of each gene of the offspring is determined by a random uniform choice of the values of the genes in the parents, with the following procedure: offspring is determined by a random uniform choice of the values of the genes in the parents, with the following procedure:

$$h_i^k = \begin{cases} c_i^1 & \text{if } u = 0 \\ c_i^2 & \text{if } u = 1 \end{cases}$$

More complex crossover schemes are available in literature (e.g. arithmetical, geometrical,linear, dynamic etc.) and details of their implementation can be found elsewhere [46, 120].

Mutation By means of the mutation operator, individuals (chromosomes) off the mating pools are modified by changing small parts in the associated vectors (genes): e. g. $(x_1, x_2, x_3, x_4) \rightarrow (x_1, x_2^{'}, x_3, x_4)$.

Within an EA, the mutation operator is defined by the size of the change it applies to a variable (mutation *step*) and by the probability at which mutation occurs (mutation *rate*). The are two main approaches to define the mutation step and rate in an algorithm [120]:

- The two parameters are kept constant during the whole evolutionary run. This is usually the case for mutation of both real and binary variables.
- One or both parameters are adapted depending on previous mutations. This approach is usually applied to mutation step–sizes in the area of evolutionary strategies.

Some of the most common mutation strategies are cited by Sumathi *et. al.* [120]:

Boundary The mutation operator replaces the value of the chosen gene with either the upper or lower bound for that gene (chosen randomly).

Non-Uniform This mutation operator increases the probability that the amount of the mutation will be close to 0 as the generation number increases. It avoids the stagnation of the population at an early stage of the evolutions, and lets the genetic algorithm to fine tune the solution at a later stage.

Uniform This mutation operator replaces the value of the chosen gene with a uniform random value selected between the user-specified upper and lower bounds for that gene.

Gaussian In this case, the operator adds a unit Gaussian distributed random value to the chosen gene. The new gene value is accepted only if it falls within the user-specified lower or upper bounds for that gene.

5.1.3.1 Diversity Preservation

The most common method implemented to preserve the diversity of the population is to decrease the possibility of survival of an individual as its neighborhood is more dense with points. The estimation of the density can be carried out using statistical methods for density evaluation, as the *Kernel methods* [139].

The *Nearest neighbour techniques* evaluate the density taking into account the distance of the point from its nearest neighbour. *Histograms* estimate density by defining a grid to identify the neighbourhood: the density around an individual is simply estimated by the number of individuals in the box. The grid can be defined a priori, or designed ad hoc for the problem of interest. A comparison of the strengths and weaknesses of the three methods has been carried out by Silverman [112].

5.1.3.2 Elitism

Elitism addresses the problem of losing good solutions during the optimisation process due to random effects [139]. In a single objective EA, elitism is implemented by simply retaining the best individual from the current generation to the following. In multi-objective problems, implementing a procedure to avoid losing best individuals is more complex, yet necessary to ensure convergence of the algorithm [20]. There are two main ways to avoid this problem. The new population (obtained from the mate pool after variation) might be added to the old population, and then a deterministic selection procedure can be carried out. The second way is to create a population which works as an archive for all the promising solutions in the population at each generation. The archive can be either kept separated from the algorithm or archive members can be used in the selection process. If an archive is kept, it comprises only the approximation to the Pareto set, while the dominated points are deleted from the archive. Usually dominance is not the only criterion to be taken into account, but also diversity is estimated to further reduce the number of solutions to store.

A possible problem arising when combining dominance and diversity to select the members of the archive is “deterioration”: solutions contained in the archive at generation t might be dominated by solutions which were in the archive at time $t' < t$, and have now been eliminated.

5.1.4 Exploration versus exploitation

As it became apparent in the previous sections, a genetic algorithm explores the search space and exploits the previously found genetic material throughout its search for better solutions. The choice of the genetic operators, and the balance among them, is critical for the success of the algorithm.

The correct balance among the different operators varies largely on the problem of interest, in particular on its modality (number of local optima present within the fitness landscape) and the relative position of the local optima [93]. If many optima exist it is appropriate to choose high crossover and mutation probabilities to encourage the exploration of the design space and to avoid premature convergence. Elitism, on the other hand, might lead to convergence to local optima prematurely.

In many cases the knowledge of the modality of the problem is not known or could be computationally difficult to acquire. In this case it is suggested to favour an exploratory approach at the beginning of the search, and subsequently encourage exploitation as the search proceeds [93]. Fine-tuning the GA parameters is one of the major drawbacks of evolutionary computing when dealing with problems characterised by multiple criteria, high modality and high dimensionality. However, if the designer is not interested in obtaining global optimal solution, as it is often the case, GAs are a powerful methods to explore the design space and identify "better" solutions within the time constraints [93].

5.1.5 Performance of Multi-objective Evolutionary Algorithms

The performance of a MOEA can be assessed either theoretically, by analysis, or empirically, by simulation.

The limit behaviour of a MOEA needs to be discussed to assess its performance, in term of the convergence properties of the algorithm. The limit behaviour of an algorithm addresses the question of what the algorithm can achieve when unlimited time resources are available.

A broad definition of convergence for MOEA, as given by Zitzler *et. al.* [139], is that a MOEA is called "globally convergent" if the sequence of Pareto front approximation $A^{(t)}$ it produces converges to the true Pareto front Y^* while the number of generations t goes to infinity. Because of memory constraints, in real application a global convergent algorithm will find a subset of the Pareto front: $A^{(t)} \in Y^{'} \subseteq Y^*$.

There are two conditions to be satisfied for the convergence of single-objective optimisation algorithm [139]:

1. an elitism selection rule, which ensures that an optimal solution is not lost and no deterioration can occur;
2. A strictly covering mutation distribution, which ensures that any solution $x^{'} \in X$ can be produced from every $x \in X$ by mutation with a positive probability;

A proper choice of the mutation and recombination operations can easily guarantee the fulfillment of the mutation condition. On the other hand, the selection rule cannot be applied as easily to the multi-criteria case since comparing two solutions is a multi-dimensional optimisation problem might not be straightforward. If too many non-dominated solutions arise than can be stored in the population, some have to be discarded. This environmental selection strategy determines whether an algorithm is convergent or not. In order to eliminate the danger of deterioration and to guarantee convergence, it is necessary to discard a solution only to replace it with a dominating alternative. This criterion can ensure a sufficient monotonicity in the sequence in which the solutions are found.

5.1.5.1 Quality assessment of a Pareto set approximation

The quality of an algorithm is evaluated both in term of computational efficiency (which can either be evaluated in terms of number of fitness evaluation or of running time on a particular machine) and in terms of quality of the outcome. In the latter case, there is a difference between single- and multiple- objective optimisation. While in the former case, if we are minimising, the smaller of two solution will be the best one, the comparison between two Pareto fronts is less straightforward. However, the concept of Pareto dominance can be used for the task. The solution can still be complicated when we want to compare two sets of solutions: some solutions in either set may be dominated by solutions in the other set, while others may be incomparable. Other aspects are also important to assess the quality of a Pareto set, such as wideness of the range of solution covered. Graphical plots have been often used to compare the outcomes of different MOEAs.

Quantitative ways to compare two Pareto sets are available, and are described in [139, 141].

It is possible to characterise a Pareto set approximation by using some numbers which can describe its quality in a rough way. Some examples of such numbers are:

1. generational distance measure, which gives an average distance of the objective vectors in the Pareto front under consideration to the closest optimal objective vector ;
2. hypervolume measure, which considers the volume of the objective space dominated by a Pareto front approximation;

However, a study by Zitzler *et al.* [141], proved the limitation of unary quality measures to make a good comparison between two algorithms. On the other hand, they indicate that binary quality measures, if properly designed, are capable of indicating whether a Pareto set approximation, S, is better than another, T. For example, a binary ϵ- quality measure can be defined on the basis of the concept of ϵ- Pareto dominance [139]:

1. **Binary ϵ- quality measure**: Let $S, T \subseteq X$. Then the binary ϵ- quality measure $I_\epsilon(S,T)$ is defined as the minimum $\epsilon \in \mathbb{R}$ such that any solution $b \in T$ is ε-dominated by at least one solution $a \in S$:

$$I_\varepsilon(S,T) = min\{\epsilon \in \mathbb{R}|\ \forall b \in T\, \exists a \in S:\ a \succ_\varepsilon b\}$$

A solution a is said to ϵ-dominate a solution b for some $\epsilon > 0$, and it is indicated with $a \succ_\varepsilon b$, if

$$\epsilon\, a_i \geq b_i \qquad \forall i \in 1, \ldots, n$$

where n is the number of objectives.

From the previous definitions we can deduct that:

- whenever $I_\epsilon(S,T) < 1$, all the solutions in T are dominated b a solution in S;
- if $I_\epsilon(S,T) = 1$ and $I_\epsilon(T,S) = 1$, then S and T represent the same Pareto front approximation;
- if $I_\epsilon(S,T) > 1$ and $I_\epsilon(T,S) > 1$, then S and T are incomparable, e.g. both contain solutions not dominated by the other set.

5.2 A multi-objective genetic algorithm (MOGA) procedure for the design of PSA cycles

In the introduction to this chapter, the paucity of alternatives for the multi-objective optimisation of PSA processes has been noticed. In this section a multi-objective genetic algorithm (MOGA) is introduced to test the ability of genetic, and more widely evolutionary, algorithms to efficiently address the multi-criteria design of a given PSA cycle in the design space defined by its operating variables. The aim is to obtain a set of optimal trade-offs between multiple objectives, the Pareto front. Valuable information can be extracted from the optima found, not only regarding the degree of separation achievable, but also about the individual and combined effect of the design variables on the objectives. The interest of the designer is to obtain from the optimisation procedure information about the extremes of the Pareto front: the points which maximise recovery and purity, respectively. Such information supports decision-making early in the design stage. Accordingly, the MOGA implemented in this work has been specifically designed to extend the breadth of the Pareto front.

The goal, for our specific design problem, is to generate as broad a Pareto set as possible, stretching the end points to identify the extremes of the trade-offs between the criteria to give the engineer sufficient information to make design decisions early in the design process [139]. In section 5.2.1, a brief overview of how desire has been addressed in literature is provided.

As mentioned earlier, a "in–house" algorithm, MOGA, has been implemented in this work as a learning exercise, as well as to facilitate both the customisation of the fitness function and the investigation of the relative importance of the different genetic operators. The MOGA is introduced in section 5.2.2.

5.2.1 Overview of well established genetic algorithms

Schaffer [110] presented a Vector Evaluated Genetic Algorithm (VEGA), designed to solve machine learning problems [20]. In this algorithm, a population is subdivided into subgroups, governed by different objective functions. Since the search directions of VEGA are exclusively parallel to the axes of the objective space, the algorithm is able to find mainly the extreme solutions on the Pareto front. A Niched Pareto algorithm, proposed by Horn *et al.* [50], incorporates the concept of Pareto domination in the selection procedure and spreads the population along the Pareto front by applying a niching pressure. Murata *et al.* [85] presented a MOGA with variable direction search, which uses a weighted sum of multiple objective functions to formulate a scalar fitness function. They compared the MOGA with the VEGA [110] and the Niched Pareto algorithm [50], showing that the VEGA outperforms the other two algorithms in the detection of points belonging to the extremes of the Pareto front, while failing to detect points at intermediate positions. Deb [26] presented a steady-state multi-objective evolutionary algorithm (MOEA) which attempts to maintain the spread while converging to the Pareto front. However, no proof of convergence for the method has been provided. Subsequently, Laumanns and co-workers [72] proposed an algorithm which caters for convergence to the Pareto optimal set while aiming to cover the whole range of non-dominated solutions, based on the concept of ϵ-dominance. Laumanns *et al.* [73] analysed the performance of different algorithms using a volume based approach (cf. [140]), with some modification: a reference volume between the origin and an utopia point – defined as the profit sums of all items in each objective – is taken into account. The aim is to minimise the fraction of that space which is not dominated by any of the final archive members. This is considered by the authors the most appropriate scalar indicator since it combines both the distance of solutions (towards the utopian trade-off surface) and the spread of solutions. Density based selection can further improve the algorithm performance by a broader distribution of solutions along the trade-off surface.

Herrero *et al.* [47] presented the $\epsilon\!\!\!\nearrow$-MOGA method, designed to achieve a reduced but well-distributed representation of the Pareto front. The algorithm adjusts the limits of the Pareto front dynamically, preventing the loss of solutions which belong to the ends of the Pareto front.

In SPEAs (Strenght Pareto Evolutionary Algorihms) both an archive with the old population and the regular population are maintained. Fitness values are assigned to both the population and the archive [139, 140]:

1. Each individual i in the archive is assigned a strength value $S(i) \in [0.1)$, which at the same time represents its fitness value $F(i)$. $S(i)$ is the number of population members j that are dominated by or equal to i with respect to the objective values, divided by the population size plus one.
2. The fitness $F(j)$ of an individual j in the population is calculated by summing the strength values $S(i)$ of all archive members i that dominate or are equal to j, and adding one at the end.

As a consequence, the algorithm has the drawback that individuals dominated by the same number of archive members have the same fitness value. Hence, when the archive contains only one individual, all population members have the same rank, indipendently of their relative position. An improved version of SPEA has been implemented, called SPEA2, where to allow two points dominated by the same archive members have identical fitness values, SPEA2 takes into account both the dominating and dominated solutions. A raw fitness is calculated according to the strength of its dominators in both archive and population, while in SPEA only aarchive members are considered at this point. Although the raw fitness assignment provides a mechanism based on the concept of Pareto dominance, it may fail when most individuals do not dominate each other. Density information are used to discriminate between individuals with the same raw fitness value. The density at any point is a decreasing function of the distance to the *k-th* nearest data point.

NSGA-II (nondominated sorting genetic algorithm II) is a nondominated sorting-based multiobjective EA (MOEA), originally proposed by Deb et al. [27]. It was proposed to address three common weaknesses of multi-objective evolutionary algorithms (EAs) that use nondominated sorting and sharing

1. $O(MN^3)$ computational complexity (where M is the number of objectives and N is the population size);
2. nonelitism approach;

3. the need for specifying a sharing parameter.

NSGA-II uses a fast nondominated sorting approach with O(MN^2) computational complexity. A selection operator creates a mating pool by combining the parent and offspring populations and selecting the best (with respect to fitness and spread) N solutions. The definition of dominance is changed to solve constrained multiobjective problems efficiently. Once the population is initialized, it is sorted based on non-domination into each front. The first front consists of the nondominated points in the current population, the second front consists of points dominated only by the individuals in the first front, and so forth. The fitness of each individual is based on the front to whom it belongs. Individuals in first front are given a fitness value of 1 and individuals in second are assigned fitness value as 2 and so on. In addition to fitness value, a new parameter called "crowding distance" is calculated for each individual. The crowding distance is a measure of how close an individual is to its neighbors. Large average crowding distance will result in better diversity in the population. Parents are selected from the population by using binary tournament selection based on the rank and crowding distance. An individual is selected in the rank is lesser than the other or if crowding distance is greater than the other 1. The selected population generates offsprings from crossover and mutation operators. The NSGA-II used in this thesis has been can be found in [111]. It applies the "Simulated Binary Crossover" (SBX) operator for crossover and polynomial mutation. Further details can be found in [111].

5.2.2 MOGA

The implementation of a genetic algorithm for any new problem requires the definition of the following elements: a representation of (hopefully) feasible solutions, the crossover and mutation operators, a selection procedure together with an appropriate fitness function, and the properties of the evolution of the population. For the MOGA used in this work, the solution representation consists of real-valued design variables. A multi-point crossover operator is defined and mutation consists of selecting a single design variable and assigning it a randomly chosen value from the domain for that variable. A tournament selection with a tournament size of 2 has been used. The population policy is one of replacement with elitism.

The key property of the MOGA used, specific to this problem, is the definition of the fitness function. Two key requirements can be extracted from the previous works. First, the need for an elitism operation which ensures that the extreme points are not lost in the evolutionary procedure. Second, that there be a driving force to extend the Pareto set outwards. Our approach is based on elitism applied to the whole Pareto set (subject to size constraints mentioned below) combined with a fitness function chosen to emphasise points that may have genetic material that could extend the Pareto set at the extreme ends.

In our MOGA, elitism is implemented by copying over the complete Pareto set from the old population to the new, from generation to generation. The only qualification is that if the Pareto set is the whole population, only half of the members of the Pareto set, chosen with a binary tournament selection procedure, are copied over to the new population intact.

The fitness of a design point in the population is based on a modified measure of the distance of that point to the approximation to the Pareto front. Solutions with lower values are fitter. For our design problem, with two criteria, the distance of a dominated point to the Pareto front approximation is the minimum of the distance of that point to each of the points in the Pareto set and the distance to infinite projections from the end points parallel to the two axes. The aim of the latter is to give emphasis to those solutions which may be far from any points in the current Pareto front approximation but which may help in generating new solutions that would extend the breadth of the Pareto front. The procedure to assign the fitness is illustrated in fig. 5.1.

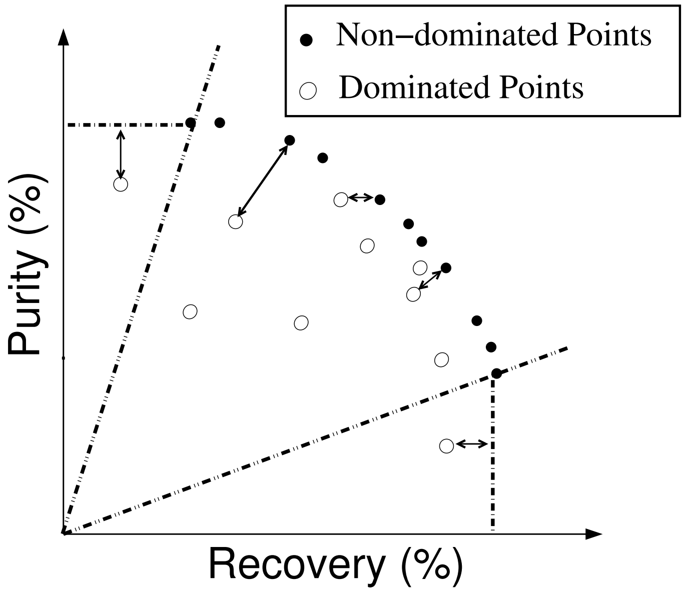


FIGURE 5.1: Illustration of the procedure to assign the fitness.

5.2.3 Implementation of the MOGA

In this section, a detailed description of how the MOGA has been implemented is presented. The goal is the simultaneous maximisation of the objectives (e.g. recovery and purity) contained in a vector z. Each design point is as well represented by a vector, whose elements correspond to the values of operating variables, such as cycle time, pressure levels and so forth.

INPUT:
a = lower bounds for the value of the decision variables;
b = upper bounds for the value of the decision variables;
x_0 = initial guess: we indicated 1 design point, corresponding to a feasible solution, as initial guess for our search;
cr = crossover rate, $0 \leq cr \leq 1$;
mr = mutation rate, $0 \leq mr \leq 1$;
n = population size;
$ne = n/2$ elite size;
ng = number of generations;
$ts = 2$, tournament size.

OUTPUT:

Pareto solutions of the multi-objective design problem. This set is represented by two arrays, one for the design points, or Pareto set, e.g. $\overline{x} = \{x_1, x_2, \ldots, x_l\}$ and one for the objective function values for these points, or Pareto front, e.g. $\overline{z} = \{z_1, z_2, \ldots, z_l\}$, where $l \leq n$. The total number of function evaluations is also returned to help evaluate performance.

STEP 1: Create initial population

The initial population consists of $\overline{x}_0$ and n-1 randomly created design points, bounded between the extremes a and b. Before a point is added to the population, a diversity check (DIVERSE) is done: a design point can be added to the population if it differs by some minimum Euclidean distance from all the points already in the population. Such check aims at ensuring a certain spread of the solutions along the Pareto front.

END OF STEP 1

STEP 2: Evolve the population

For ng times a new population will be created.

STEP 2, *I*. Fitness: Evaluate Fitness of the actual population

The first step to evaluate the fitness is to detect the non-dominated points within the population: a point $\overline{x}_1$ dominates $\overline{x}_2$ if $\overline{z}_1 \geq \overline{z}_2$ for all the objective function values and $\overline{z}_1 > \overline{z}_2$ for at least one objective function value. The set of non-dominated points is the current approximation to the Pareto set.

For all points of the population the fitness value is inversely proportional to the distance from the approximation to the Pareto front. However, if a point is nearer to the infinite projections from the extreme of the Pareto front parallel to the axes, such distance is used to estimate its fitness. The purpose of this definition of fitness is to give a better chance of survival to those points which could help to broaden the extent of the Pareto front.

STEP 2, *II*: Selection

To start with, the Pareto set from the previous iteration is copied over to the next population, as a form of elitism. Should the old Pareto set have already n individuals, we would not have any room to add new individuals. In this case,only $ne = n/2$ solutions are copied to the next generation. To halve the population we use a tournament selection procedure of size 2. For ne times, two random points are picked within the population, and only the fitter one is copied over to the next generation.

STEP 2, *III*: Genetic operations

While the size of the new population is smaller than the desired one (n), new individuals are generated according to the procedure illustrated in algorithm 5.1. To perform the MATING operation on $\overline{x}_1$ and $\overline{x}_2$, the two points are combined to obtain two new points. To mutate an individual $\overline{x}_1$, only one of the values of the decision variable of $\overline{x}_1$ is varied: e.g. $\overline{x}_1 = [x_1, x_2, \ldots, x_l] \rightarrow [x_1, x'_2, \ldots, x_l]$. A stochastic procedure is used to choose which element of Which of the elements of $\overline{x}_1$ is to be changed.

END OF STEP 2

Some considerations on the effect of the variables of the MOGA are needed. The higher *cr* and *mr* the more likely are genetic modifications to happen and the more “new genetic material” is going to be added. As a consequence the search becomes more random, as less information are kept from generation to generation. A certain amount of change is

Algorithm 5.1 How to add new individuals to the population

```
while The size of the population is smaller than $n$ do
    SELECT two points of the population, $\overline{x}_1$ and $\overline{x}_2$;The selection procedure consists of picking two random points of the population and choosing the fittest of the two
    1. Generate a random number, $0 \leq r \leq 1$.
    if $r < cr$ then
        Create two new points by MATING $\overline{x}_1$ and $\overline{x}_2$
        Call the new points $\overline{x}_1$ and $\overline{x}_2$
    end if
    2. Generate a random number, $0 \leq r \leq 1$
    if $r < mr$ then
        Create a new point by MUTATING $\overline{x}_1$
        Call the new point $\overline{x}_1$
    end if
    if $x_1$ is DIVERSE enough then
        Add $x_1$ to the population
    end if
    3. Generate a random number, $0 \leq r \leq 1$.
    if $r < mr$ then
        Create a new point by MUTATING $\overline{x}_2$
        Call the new point $\overline{x}_2$.
    end if
    if $\overline{x}_2$ is DIVERSE enough then
        Add it to the population
    end if
end while
```

desirable to explore satisfactorily the design space, but an excessive randomness might hinder the evolution towards better results. The optimal values of cr and mr are very specific to the problem of interest.

The search for optimal solution stops when a certain number of generations (ng) has been performed. As a consequence, the higher ng the higher the chances of getting a better approximation to the Pareto front, but the higher the computational expenses. In the proximity of the Pareto front, there will be a lower advantage in increasing ng at a high computational cost: as a consequence a trade-off value for this parameter has to be detected.

The bigger the size of the population n the more likely is to find a good spread of solutions along the Pareto front, and the better the approximation to the Pareto front. Also the reliability of data seems to improve. Such improvements are paid by higher computational costs.

The analysis of the performance of the MOGA is carried out in the remainder of this chapter.

5.3 Multi-objective optimisation of PSA cycles in literature

Ko and Moon [63] proposed a modified summation of weighted objective functions (mSWOF) method for the multi-objective optimisation of a small TSA cycle for the purification of a ternary mixtures of benzene, toluene, and *p*-xylene and of a RPSA cycle for air separation to produce oxygen. They used a black-box approach within gPROMS. The advantages of the proposed mSWOF, over the conventional SWOF method, are the ability to find a uniform spread of Pareto points, including the nonconvex parts of the Pareto set, and to perform sensitivity analysis. The limitation of the mSWOF method is that since the objectives are optimised independently, the extension of the method to more than two objectives is likely to be non trivial. Furthermore, Ko and Moon considered only the LDF (linear driving force) approximation to mass transfer, so no evidence of the feasibility of the method with more complex models has been shown.

Sankararao *et al.* [109] considered a simple Skarstrom cycle operating among two beds. The operation taken into account is air separation for O_2 production on a zeolite 5A. The optimisation technique used is the “modified MOSA-aJG”: MOSA-aJG is an expansion of the simple simulated annealing (SSA) for multi-objective optimisation, where “MOSA” stands for multi-objective simulated annealing, and “aJG” is a jumping-gene (JP) adaptation of the MOSA. The simulation uses a detailed diffusion model for mass transfer. However, this simulation has the handicap that it needs to be tuned on experimental data. This represents a big limit to the range of problems that this approach can tackle. The strategy used in the article to detect the performance at cyclic steady state (CSS) is the successive substitution, which, together with the detailed simulation, requires a long simulation time (24h on a Pentium 4, 2.99 GHz [109]).

Three different multi-objective optimisation problems are taken into account in [109], considering 2 or 4 objective functions, chosen among purity and recovery of the two components (N_2 and O_2) in the raffinate and in the extract at CSS. Although the optimisation method used is stochastic, the results in the article are not accompanied by any statistical analysis on the performance of the optimiser and it is therefore impossible to know what to expect from this method in general.

In the next sections, the optimisation of the two case studies introduced in chapter 3 is carried out using MOGA. The performance of the optimisation approach is tested against random search and direct search methods. A statistical analysis is carried out to investigate the effect of the tuning of the genetic operators.

5.4 N_2 Case Study: Analysis of MOGA Performance

The problem we would like to solve is the simultaneous optimisation of the recovery and purity of the desired product. For the N_2 case study, the design problem can be formulated as follow:

$$\begin{aligned} &\max_{t_c, r_s, F_{in}, S} \text{Recovery \& Purity at CSS} \\ \\ &s.t. \\ &\text{Evaluation of the model at CSS;} \\ &\text{Pressure} < 7\text{atm;} \end{aligned} \tag{5.1}$$

The recovery of N_2, eq. 4.5, is defined as the fraction of N_2 in the product stream, F_{prod}, with respect to the amount fed to the system over the cycle.

The purity is given by the concentration of N_2 in F_{prod} and is evaluated by the following integral:

$$\text{purity}_{N_2} = \frac{\int_{t_{ads}} \left(F_{prod}\, y_{N_2}\right)\, dt}{\int_0^{t_{ads}} F_{prod}\, dt} \tag{5.2}$$

There are 4 design parameters: $t_c \in [15, 250]$ s is the cycle time which defines the duration of the four steps, $r_S \in [0.20, 0.85]$ the split ratio, described in section 3.3.1, $F_{in} \in [15, 100]$ mol/s the inlet flow rate and S the schedule, S. The constraints for the optimisation problem are the evaluation of the PDAEs to CSS and a maximum value of the high pressure, $P_H < 7$ atm. If a design point violates the pressure constraint, the corresponding values of purity and recovery are set to 0.

The investigation carried out in the previous chapter showed that the objective function is non-smooth and non-convex. Interestingly, also the design space in non-convex, as shown below. Let us consider two design points, $\overline{x}_0 = [231.9, 0.84, 65.3, 44.5]$ and $\overline{x}_1 = [123.2, 0.33, 29.9, 21]$. As we move from $\overline{x}_0$ to $\overline{x}_1$ in the design space, the values of all the design variables (t_c, r_S, F_{in} and S) decrease, while t_{press} increases from 12.75 to 35.7 s. As F_{in} decreases, the value of P_H decreases as well. Conversely, the increase of t_{press} makes the pressure increase. As a result, the value of P has a maximum as we move in the space from $\overline{x}_0$ to $\overline{x}_1$. The value of this maximum is 7.37, above the P_{max} imposed as a constraint. Hence, there will be a non-feasible region in the area around this maximum which corresponds to a non-convexity in the design space, as shown in fig. 5.2.

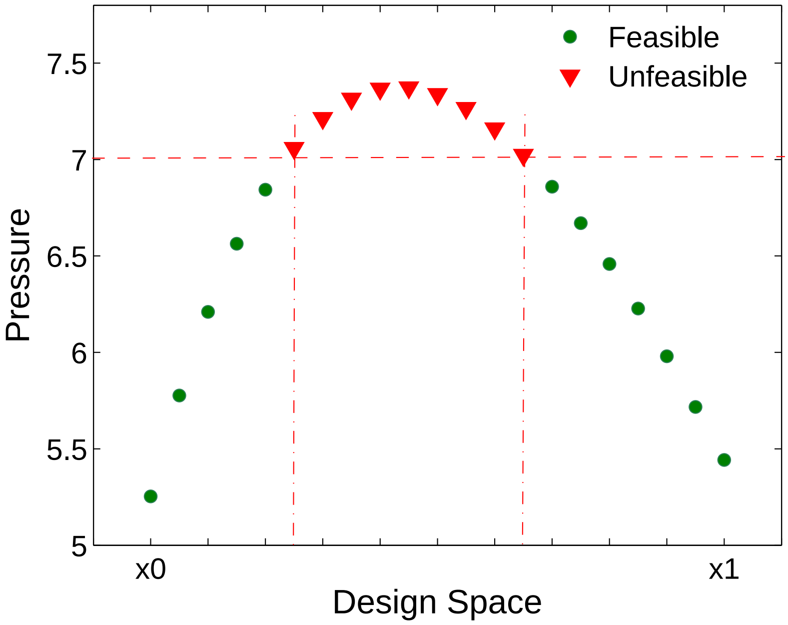
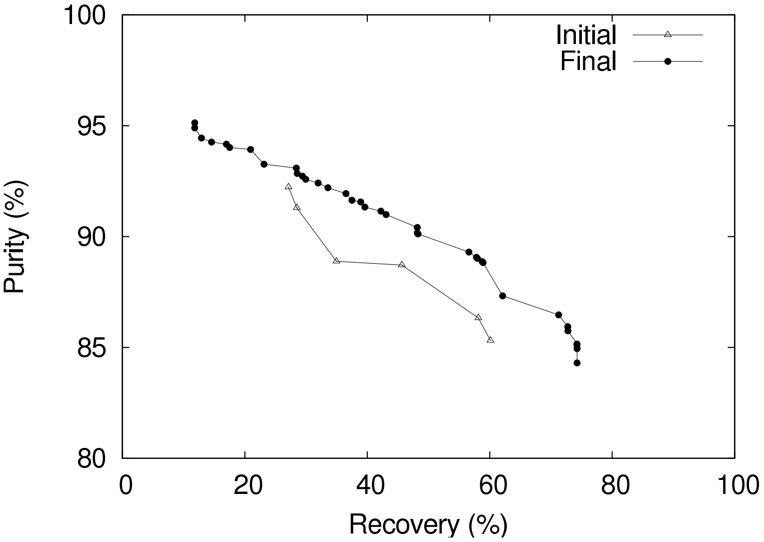


FIGURE 5.2: As we move from point $\overline{x}_0$ to point $\overline{x}_1$ in the design space, the pressure assumes values which violate the constraint on the maximum pressure allowed, hence generating a non-convexity in the feasible design region.

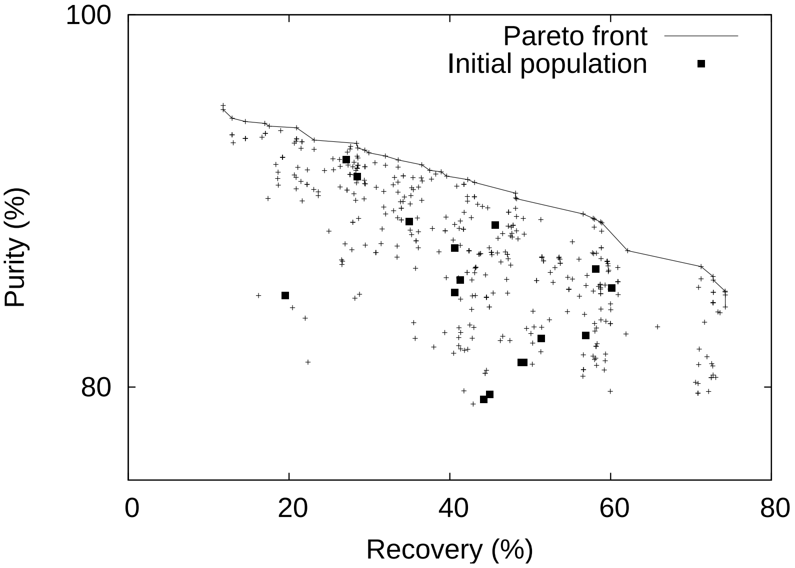
Another important characteristic of the problem is the computational demand of the simulation. The models proposed in Chapter 2 are detailed enough to describe the behaviour of the PSA operation, yet less computationally demanding than a full column simulation. A number of 6 CSTRs has been used to simulate each bed. However, the evaluation of the objectives still makes the optimisation problem complex, possibly requiring several minutes of computation for a single evaluation, depending on the resources available and the particular case study considered. To give an indication of the computational demands for this design problem, an objective function evaluation, based on convergence to cyclic steady state requiring, on average, 6 cycles, takes approximately 90 s on a 2.8 GHz Pentium processor.

The data presented in fig. 5.3 have been generated using the values summarised in Table 5.1. Fig. 5.3(a) illustrates the initial and final Pareto sets for a typical run. The initial set is from the initial randomly generated population and, in this case, consists of 6 points. The final Pareto set has 36 points, evenly distributed across the front, out of a population of 40. We see an improvement when compared with the initial set, across the front, not just in objective function values but also in the breadth of this approximation to the front.

The effectiveness of the multi-objective optimisation procedure in searching the objective function space is shown in fig. 5.3(b). The figure shows a fairly even distribution of points across the two criteria with an increasing density towards the Pareto front.



(a) Representative initial and final Pareto sets.



(b) A graphical representation of all the design points generated during the search with the initial population highlighted and the final Pareto front approximation drawn. In the initial population of 40 individuals, only 16 were feasible (P <7 atm). The unfeasible points are not shown as the values of their objectives (purity and recovery) is set to 0.

Figure 5.3: Analysis of the performance of the multi-objective genetic algorithm. All the parameters used are listed in Table 5.1.

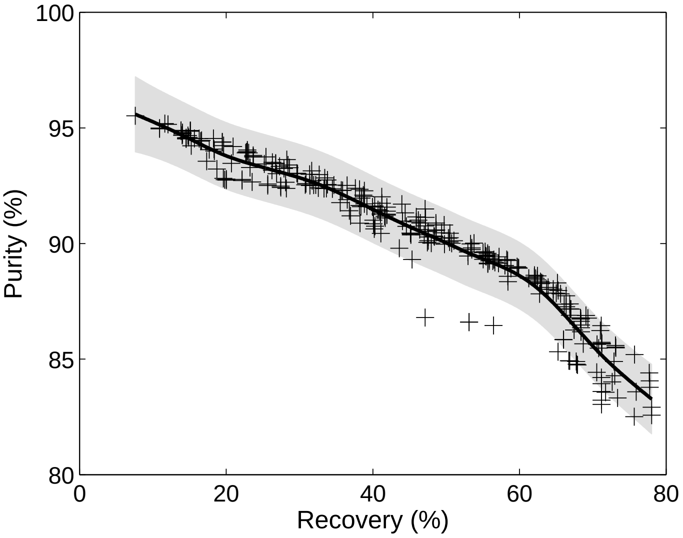


Figure 5.4: Average performance of 10 optimisation runs. All the parameters used are listed in Table 5.1.

More importantly for our design goals, when the full set of design points generated is compared with the initial random population generated by the MOGA procedure, the fitness function defined above has been effective in generating a broad Pareto front which helps the engineer gain a better understanding of the trade-offs involved.

Table 5.1: Parameters used in MOGA

Crossover rate	cr	0.7
Mutation rate	mr	0.1
Population size	n	40
Number of generations	ng	50
Tournament size		2

As genetic algorithms are stochastic, the assessment of the performance of the MOGA requires a statistical analysis of the results. For every given set of parameters, the average Pareto fronts and the standard deviations have been determined by performing a Gaussian progress regression [100] of 10 Pareto fronts. We obtained a good fit to our data (the Pareto fronts generated) by using as a covariant function the sum of a Matern covariance function with a shape parameter of 3/2 and an independent noise. The process is illustrated in fig. 5.4, generated using the set of parameters in Table 5.1. This analysis allows us to gain an insight on the effect of the different parameters of the MOGA on both the average Pareto front obtained, as well as on the dispersion of the data. In the course of this analysis, only average Pareto fronts obtained through regression will be shown, unless the standard deviation is significant for the analysis. The parameter values used are those in Table 5.1, unless otherwise noted. This statistical approach has been used to compare the performance of MOGA and DSMs (section

5.4.1), and to study the effect of the parameters of the MOGA on the resulting Pareto front (section 5.4.2).

5.4.1 Comparison between MOGA, random search and Direct Search Methods

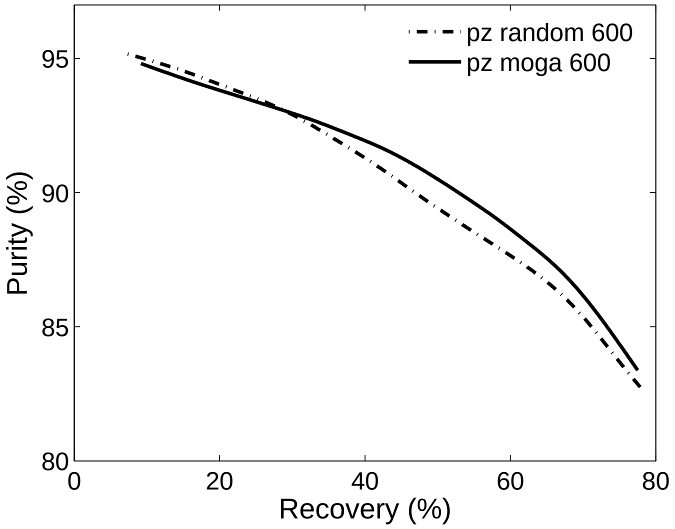


FIGURE 5.5: Comparison between the Pareto front obtained with 600 random evaluations and the Pareto front obtained with the same number of evaluation by MOGA (with ng=30 and n=40 and other parameters as in Table 5.1).

The capability of the MOGA to efficiently detect the Pareto front has been assessed by comparison with random search and DSMs. The aim of the comparison with a random search algorithm is to verify that the evolutionary procedure implemented in the MOGA proposed (i.e. definition of fitness evaluation, selection, mating and mutating procedures) is effective. A random search algorithm has been included also in the comparison among evolutionary algorithms proposed by Zitzler *et al.* [138].

The average Pareto fronts obtained with MOGA (averaged over 10 runs, using approximately 600 evaluations each) have been compared to the average of 10 sets of non-dominated points, obtained from 10 sets of 600 randomly generated points. The results are presented in fig. 5.5. The Pareto front obtained with the MOGA shows a big improvement with respect to the homologous random one, and a larger number of non-dominated points is detected by the MOGA. The performance of the two methods in the high purity region is comparable, with a slight dominance of the solutions found by the random search: this shows that it is easier to detect solutions with high purity/low recovery and the impact of the evolutionary procedure in this area is less crucial.

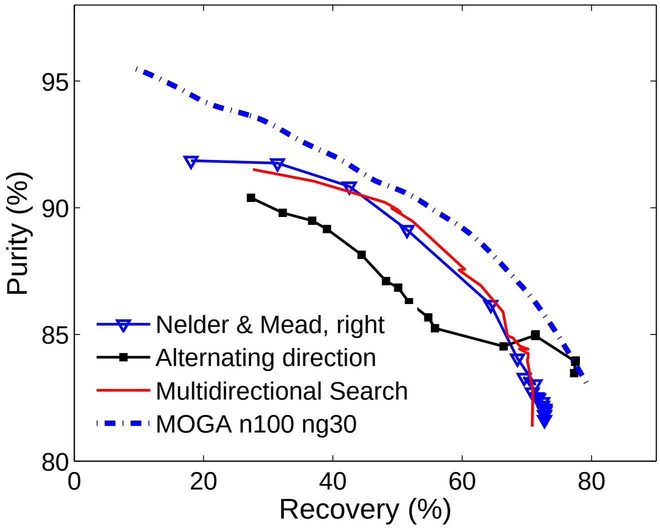


Figure 5.6: Comparison of the average performances of MOGA and DSMs

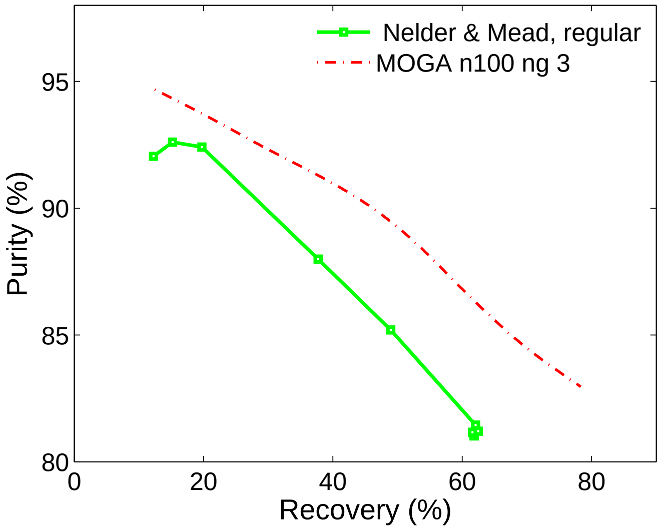


Figure 5.7: Comparison of the average performances of MOGA and Nelder & Mead, regular simplex

We also compared the performance of the MOGA and of some standard DSMs for the solution of our multi-criteria problem. We compared the performance of these two classes of optimisers to verify that evolutionary algorithms are effectively the best option for the case of interest. Although a better performance by the MOGA could be considered predictable, there are cases reported in literature [82] where the performance of a genetic algorithm and a DSM are comparable.

DSMs were originally designed to solve single objective problems. Hence, it is necessary to combine the two objectives to be maximised, i.e. recovery and purity, into a single objective function. The combined expression must allow a different weight to be given

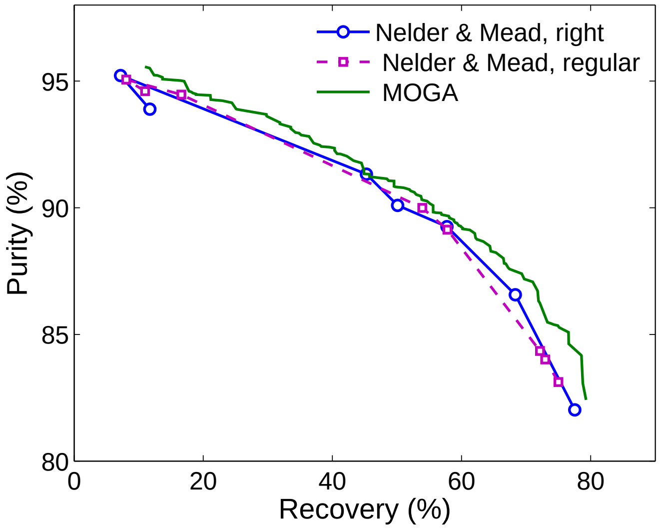


FIGURE 5.8: Comparison of the best performances of the DSMs and MOGA (with n 100, ng 30 and other parameters as in Table 5.1).

to each objective so that different trade-offs between the two variables can be optimised. The composite criterion to be maximised has been expressed as:

$$f(\overline{x}, \lambda) = f_1(\overline{x}) - \lambda\,(f_1(\overline{x}) - f_2(\overline{x})) \tag{5.3}$$

where $0 \leq \lambda \leq 1$, and $\overline{x}$ represents the vector of design variables. In this case, f_1 is the recovery of N_2, and f_2 the purity. The methods used were Nelder and Mead, with both regular and right simplex, alternating directions and multidirectional search [60]. As DSMs are deterministic methods, we used a statistical procedure to assess the performance of each direct search algorithm different from the procedure used for the MOGA. We randomly selected ten points within the feasible region of the design problem, e.g. $\overline{x}_{0,i}$, where $i = 1, ..., 10$. Starting from these points, each of DSM has generated a Pareto front by optimising the function f, eq. 5.3, with λ varying from 0 (maximisation of recovery) to 1 (maximisation of purity), with regular increment of 0.05. The performance of every method is then given by the average of the 10 Pareto fronts it generated, and the relative standard deviation.

As shown in fig. 5.6 and fig. 5.7, the average performance of MOGA is better than all the four DSMs used. The comparison has been made between the average performance of each method using a comparable number of function evaluations. This information is summarised in Table 5.2. In all cases, MOGA achieved better results using a lower number of function evaluations, which demonstrates the efficiency of the algorithm. In fact, higher values of purity (for a given recovery) have been detected, and a larger breadth of the Pareto curve has been achieved by MOGA. The distribution of points

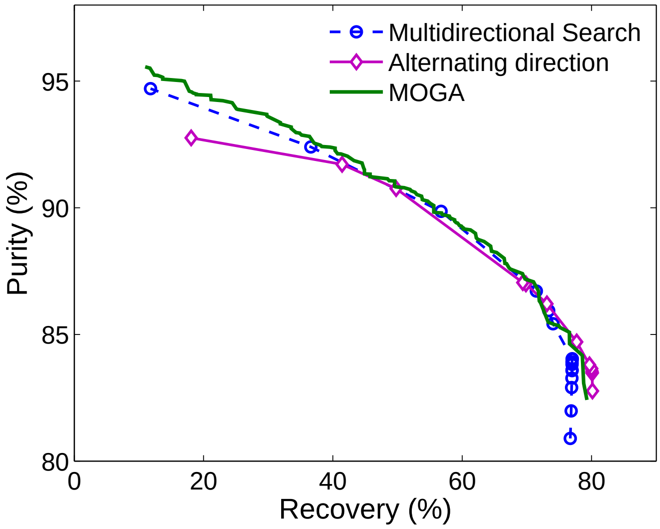


Figure 5.9: Comparison of the best performances of the DSMs and MOGA (with n 100, ng 30 and other parameters as in Table 5.1).

Table 5.2: Total number of function evaluations used by the different methods to produce the Pareto fronts of figures 5.6,5.7,5.8 and 5.9. The parameters used by MOGA for the comparison are as in Table 5.1 unless otherwise stated.

Method	N Eval per Pareto
Nelder & Mead regular	286
Nelder & Mead right	1,916
Alternating Directions	12,190
Multidirectional Search	4,000
MOGA n 100, ng 30	1,800
MOGA n 100, ng 3	250

along the Pareto fronts generated by alternating direction methods and multidirectional search methods is quite uniform, while the Nelder & Mead methods have generated more points in the high recovery region. The comparison among the best results achieved by the different methods shows that all the algorithms find an approximation to the effective Pareto front, but the approximation of MOGA is more complete (figs. 5.8, 5.9). The difference between the average and the best performance of DSMs revealed the sensitivity of DSMs to the starting point. As described before, we started each method from 10 points randomly chosen within the design space. We obtained better results in the high recovery region from starting points already belonging to that area, and the same occurred for the high purity area. This means that a pre-knowledge of the design space would be required to obtain a good approximation of the Pareto set with any of DSMs, which represents a significant drawback with respect to MOGA.

5.4.2 Impact of algorithm parameters

The parameters for the MOGA, e.g. the population size, mutation and crossover rates and the number of generations, will have an impact on the quality of the approximation to the Pareto front that we obtain.

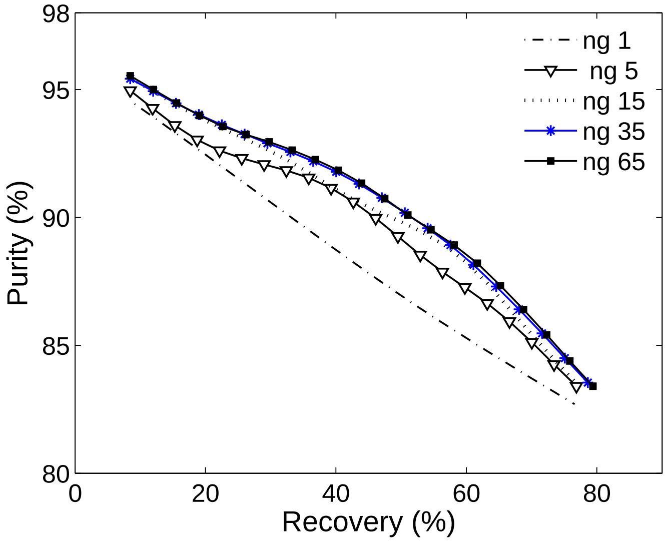


FIGURE 5.10: Influence of the number of generations (ng) allowed. Convergence is reached in 35 generations.

We changed the parameters of MOGA one at the time, while keeping the others to the values summarised in Table 5.1. As explained in section 5.4, only average Pareto fronts obtained through regression will be shown, unless the standard deviation is significant for the analysis.

Fig. 5.10 shows that as the number of generations (ng) increases, the Pareto front improves till convergence, which is reached in 35 generations.

The effect of the mutation rate (mr) is illustrated in fig. 5.11: as the mutation rate increases we get better results in the central area of the Pareto front, while the effect is not clear at the extremes of the curve. We do not observe a direct effect of the mutation rate on the deviation of the data.

The tournament size (ts) has no effect on the final performance of the algorithm in the range of values investigated (i.e. ts=2, 4, 6). However, a slight difference in the rate of convergence is noticeable in fig. 5.12, as at the same number of generations the Pareto

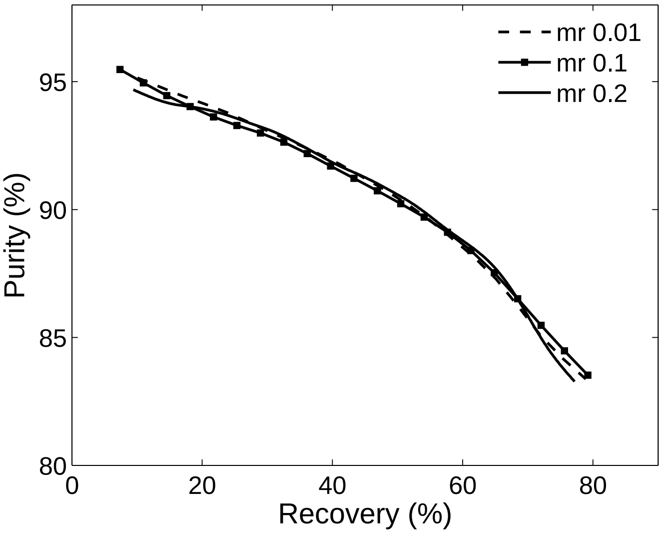


FIGURE 5.11: Influence of the mutation rate (mr), with mr=0.01, mr=0.1, mr=0.2.

front detected with tournament size 4 dominates the other, and is more extended in the high recovery region.

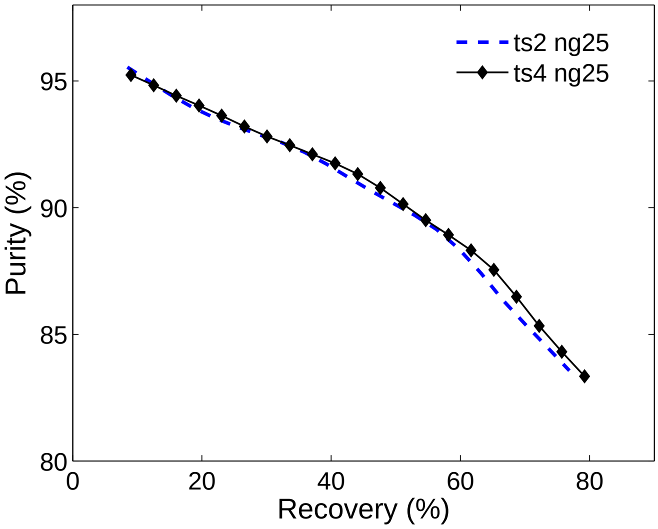


FIGURE 5.12: Average Pareto front produced with tournament size 2, 4 at 25 generations.

The most influential parameter appears to be the size of the population (n) used: in fig. 5.13 as the size of the population increases the Pareto front is pushed further, while the standard deviation of the data decreases (fig. 5.14).

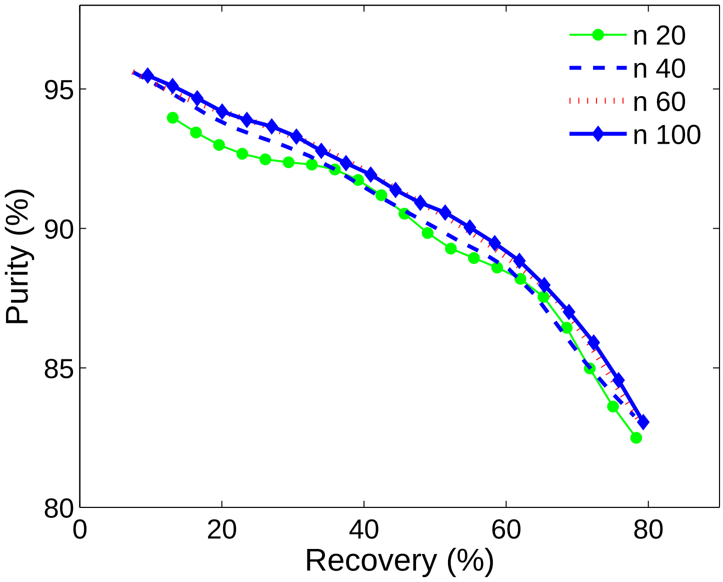


FIGURE 5.13: Influence of the population size (n), with n=20, n=40, n=60, n=100

As shown in fig. 5.15, increasing the size of the population we are able to investigate more accurately the search space: although the same trends are indicated in fig. 5.15 (a) and (b), the clumping of solutions is more evident in fig. 5.15 (a), whereas the solutions found with a larger population are more distributed within the search space.

However, such an improvement is counterbalanced by a higher computational requirement: the number of function evaluations needed grows from an average of 950 with $n = 40$, to 1410 with $n = 60$, to 2480 with $n = 100$ and $ng = 50$.

5.4.3 Comparison between MOGA and NSGA-II

The experience with the MOGA has been a preliminary investigation into the suitability of multi-objective evolutionary algorithms for the design of PSA processes. The success of this preliminary investigation now motivates us to investigate the use of existing and validated evolutionary algorithms, such as NSGA-II. A description of NSGA-II can be found in section 5.2.1.

The overall population, consisting of current population and current offsprings, is sorted again based on non-domination and only the best N individuals are selected, where N is the population size. The selection is based on rank and the on crowding distance on the last front.

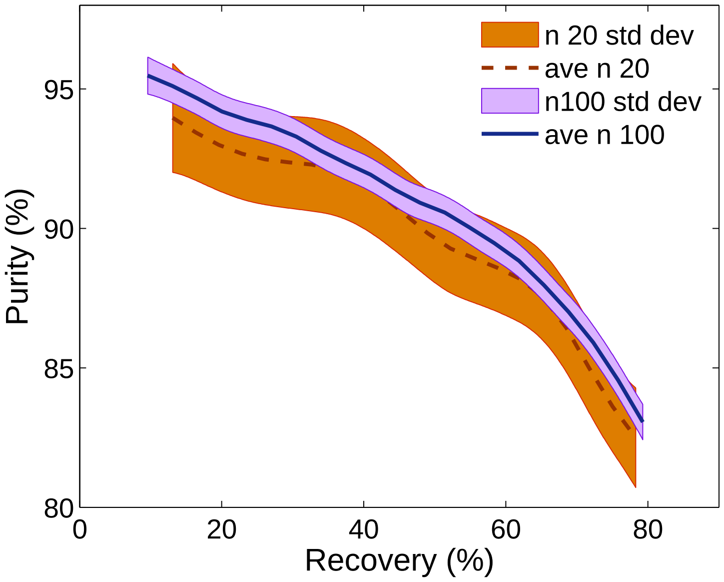


Figure 5.14: Influence of the population size on the standard deviation from the Pareto front

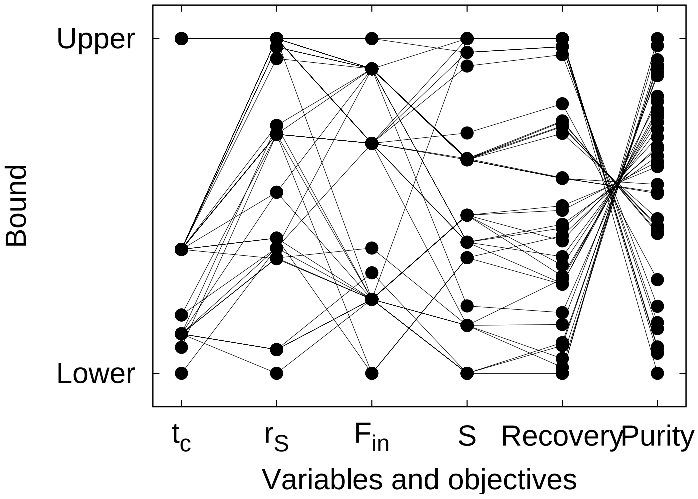
Results shown by Deb *et al.* [27] proved that NSGA-II, in most problems, is able to find much better spread of solutions and better convergence near the true Pareto front compared to Pareto-archived evolution strategy and strength-Pareto EA, two other elitist MOEAs.

The NSGA-II has been implemented with a population size n=50, and ng=50 generations. For the crossover and the mutation probability, we used the default values proposed by [111]. The crossover probability cr = 0.9 and mutation probability is mr= $1/d$, where d is the number of decision variables. The problem as been run 10 times. The number of function evaluation needed to perform the 50 generations is around 3500.

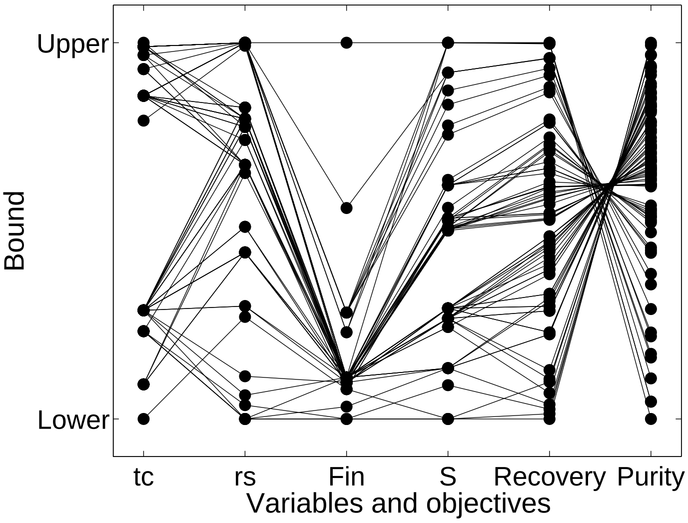
The performance of the two algorithms appear comparable, as shown in fig. 5.16, In particular, NSGA-II appears equivalent or slightly dominant at the extremes of the Pareto, while MOGA is dominant in the middle area of the Pareto.

5.4.4 Design problem analysis

Analysis of the Pareto sets identified in all the runs show clear clumping of the solutions. This manifests itself in a parallel co-ordinate visualisation of the Pareto set shown in fig. 5.15. In 5.15(b), although there are 72 solutions comprising the Pareto set, there



(a) 36 different solutions are represented in this graph, obtained with a population of 40 individuals.



(b) 72 different solutions are represented in this graph, obtained with a population of 100 individuals.

Figure 5.15: Pareto set visualised using a parallel co-ordinate representation. Variable domains and objective function value ranges have been normalised for presentation.

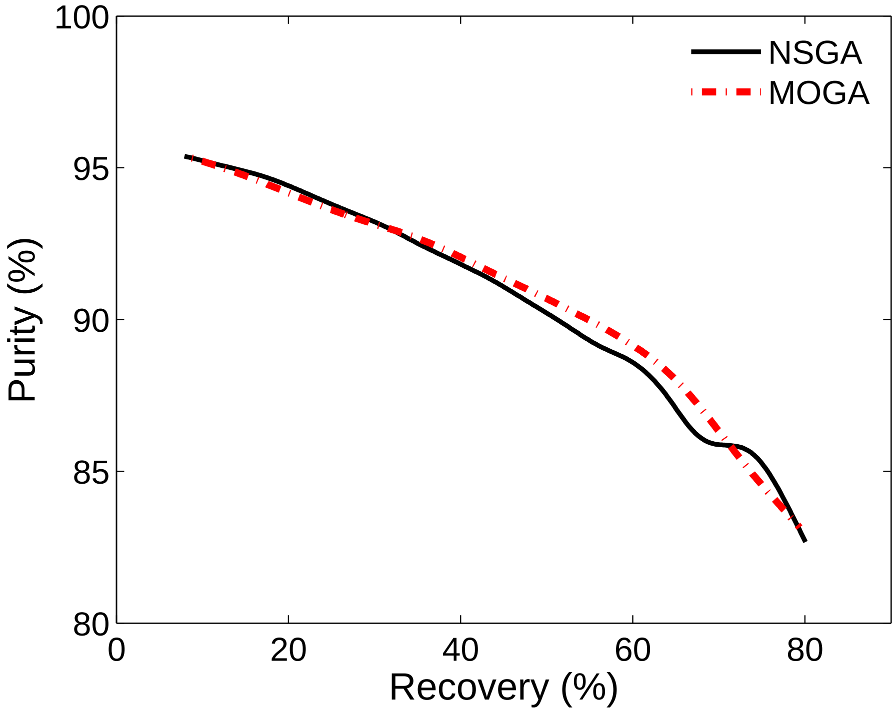


FIGURE 5.16: Comparison between the average performance, obtained via Gaussian regression of 10 Pareto sets, of NSGA-II (using n=50, ng=50, cr = 0.9 and mr= 0.25) and MOGA (using n=50, ng=50, cr=0.9 and mr=0.1).

are only 9 distinct values of the cycle time, t_c, and 8 for the feed flow rate, F_{in}. To a lesser degree, the same clumping is observed for the split ratio, r_S, and schedule, S, design variables. Also apparent from the clustered intersection points for the lines connecting the two criteria is the inverse relationship between these two criteria, as expected. Overall, the results illustrate the complex relationship between the design variables and the objectives for the design problem. There appear to be hyper-slices in the 4 dimensional design space, defined by values of t_c and F_{in}, which correspond to families of good solutions for particular trade-offs between the two criteria. An almost linear relationship between the two criteria is highlighted not only in the parallel co-ordinate visualisation but also in the Pareto graph.

The cycle time ranges in $\in [137, 248]$, $r_S \in [0.33, 0.82]$, $F_{in} \in [30.5, 87.9]$ and the schedule $\in [20, 44.5]$ %. While an almost linear relation holds between purity and recovery, no design variable shows monotonic behaviour as the Pareto front is traversed.

We can compare our results with those presented by Hassan *et al.* [45]. They also considered air separation on a molecular sieve using a 4-step Skarstrom cycle. To summarise, the following trends have been described in[45] and observed in the results of our optimisation runs:

- As the split ratio increases, the recovery increases and the purity decreases. This is as expected. The regeneration of the adsorbent (i.e. lower F_{purge}) is less effective with increasing split ratio. Also, the amount of product withdrawn, F_{prod}, increases with the split ratio. The effect of the split ratio is exploited in a modified Skarstrom cycle commonly adopted for air separation, where the purge step is eliminated (i.e. $r_S = 1$) to maximise recovery, while a pressure equalization step is introduced to gain an optimal regeneration of the bed as well as to save compression work [44, 108].

- Hassan *et al.* [45] explored the effect of the purge to feed ratio. This is equivalent to looking at $(1 - r_S)/F_{in}$ for our model. Again, we find agreement in our results: as the purge/feed ratio increases, we get a lower recovery and a higher purity. This is shown clearly in fig. 5.17.

Hassan *et al.* [45] investigated the effect of the schedule in a restricted range $[75, 90]$ % and they found no appreciable effect on the performance of the operation. In our case, however, the use of detailed modelling and advanced computational tools has allowed a wider range of S to be explored. We have observed that the longer the t_{ads} for a given t_c, the higher the recovery and the lower the purity.

Although we do not observe a monotonic relation between the cycle time and the objectives, we do note that lower t_c values correspond to lower recoveries and higher purities. To some extent, peaks in t_c correspond to valleys in F_{in} as the two variables balance out to allow the pressure to satisfy the operating constraint.

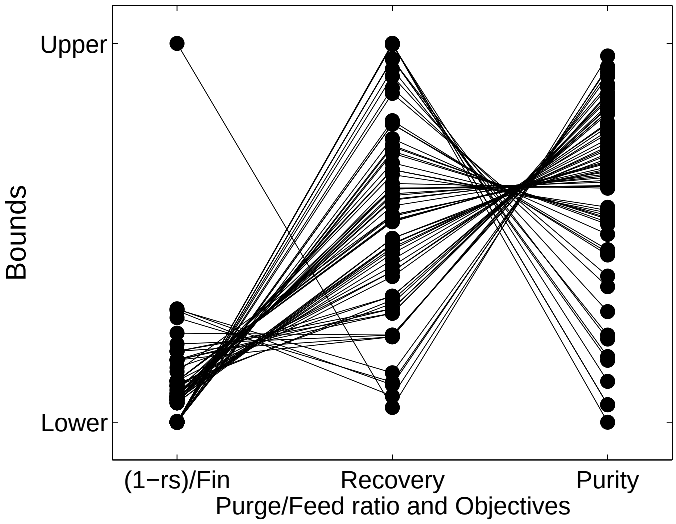


FIGURE 5.17: Influence of the Purge/Feed ratio on the objectives

5.5 CO_2 Case study

The second case study has been the simultaneous optimisation of CO_2 recovery (R) and enrichment (E) in the product, using MOGA. The recovery is evaluated as the number of moles of CO_2 in the product, divided by the number of CO_2 moles entering the bed during the adsorption step. The enrichment is the ratio between the concentration of CO_2 in the product, or purity, and the concentration of CO_2 in the feed (15%): a maximum enrichment of 6.67 can be achieved, value of the ratio 100/15. The enrichment has been used instead of the purity to facilitate the comparison with the results presented by Reynolds *et al.* [101]. The configuration investigated is the 4-bed/4-step PSA cycle introduced in section 3.2, originally proposed by [101]. The only constraint is the evaluation of R and E at cyclic steady state (CSS), which implies the dynamic simulation of the operation as described in chapter 2.

The analysis of the cycle has been made in the 4-dimensional design space defined by the same parameters used in [101]. The parameters are the purge-to-feed ratio (γ), the cycle time (t_c), the pressure ratio ($\pi_T = P_H/P_L$) and throughput of the operation (θ). θ is defined as the amount of feed fed to one reactor during the adsorption step (step I) divided by the cycle time and the mass of adsorbent in one bed. It is worth noting that θ, defined by eq. 5.4, is indeed independent of the cycle time as the feed inlet, F_{in}, is constant.

$$\theta = \int_0^{t_{ads}} F_{in} dt / t_c V_{bed} \tag{5.4}$$

The value of the high pressure has been kept constant, to the value of 1.36atm [101]. Hence, the value of π_T varies only in dependence of P_L.

According to our model, the independent variables that can be accessed are the value of the inlet feed rate (F_{in}), the cycle time t_c, the value of the low pressure P_L and the value of the split ratios. Hence, it has been necessary to express the independent variables of the model as a function of the variables used in [101], i.e. γ, $\pi_T = P_H/P_L$ and θ. The cycle time has been used as an independent variable.

The relations found are:

$$\theta = \frac{\int_0^{t_{ads}} F_{in} dt}{t_c V_{bed}} \tag{5.5}$$

since F_{in} is constant and all the four steps need to have the same duration, $t_c = 4\, t_c$, the equation 5.5 can be simplified as $\theta = F_{in}/4\, V_{bed}$.

According to the definition

$$\gamma = \frac{\int_0^{t_{des}} F_{in,step\,3} dt}{\int_0^{t_{ads}} F_{in} dt} = \frac{r_{s1} \int_0^{t_{des}} F_{out,step\,1} dt}{\int_0^{t_{ads}} F_{in} dt} \tag{5.6}$$

where r_{s1} express the fraction of the output of step I that is sent as purge to step III, according to $F_{in,step\,III} = r_{s1}\, F_{out,step\,I}$.

Equations 5.5 and 5.6 need to be rearranged so that F_{in} is expressed as a function of θ and, similarly, r_{s1} as a function of γ.

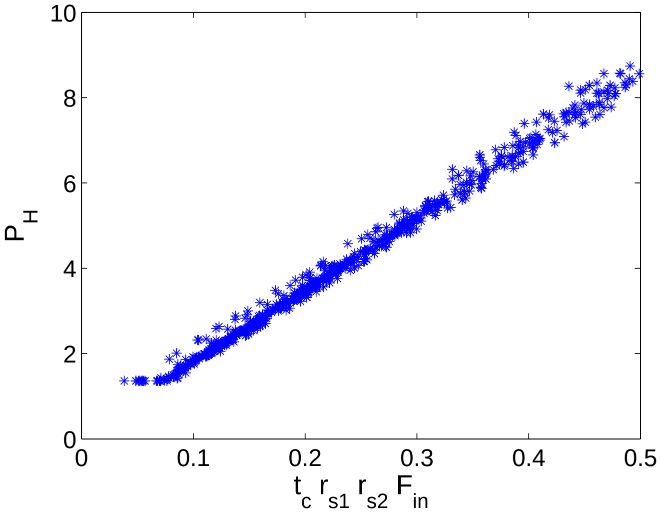


FIGURE 5.18: Relation between the value reached by the pressure and the design variables.

Finally, it has been necessary to control the value of the high pressure P_H reached during the pressurisation step. No analytical expression of P_H is available in the model. However, P_H is a function of the duration of the duration of the pressurisation step and the inlet feed rate of the step, $F_{in,stepIV} = r_{s1}\, r_{s2}\, F_{out,stepI}$, where r_{s2} is the only remaining degree of freedom. As shown in figure 5.18, a strong relation holds between the value of P_H and the product of $t_c\, r_{s1}\, r_{s2}\, F_{out,stepI}$: to keep the value of P_H around 1.36 atm, the value of r_{s2} needs to be tuned so that $t_c\, r_{s1}\, r_{s2}\, F_{out,stepI} \leq 0.08$.

Our simplified model provides a good match with the results of the rigorous model used in [101], as shown in Fig. 5.3. A good qualitative match is achieved. However, our results underestimate the recovery while overestimating the enrichment, so that a reasonable but not perfect qualitative match is achieved. This is expected as our models differ: not only we have modelled each bed as a series of CSTRs, but also we added the assumption that the operation is adiabatic when writing the energy balance. The

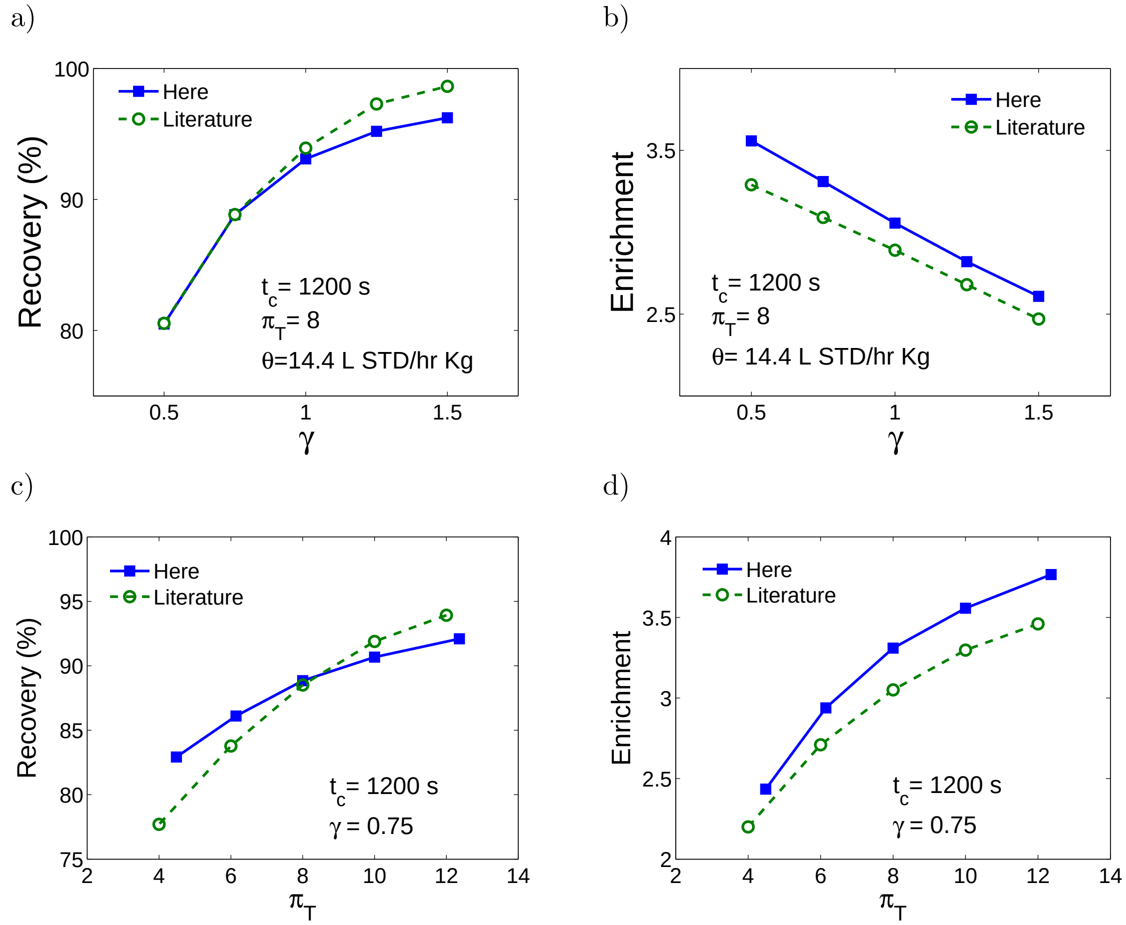


Table 5.3: Effect of γ (a and b) and π_T (c and d) on recovery and enrichment. The comparison with literature data from [101] is shown.

choice of considering the operation adiabatic is due to an observation made by Zhang *et al.* [136] : since industrial operations are usually carried out in bed of more than 3m of diameter, real industrial PSA process are close to an adiabatic performance. Since the relationship between design variables and objectives is correctly described, the model can be considered satisfactory for our purpose.

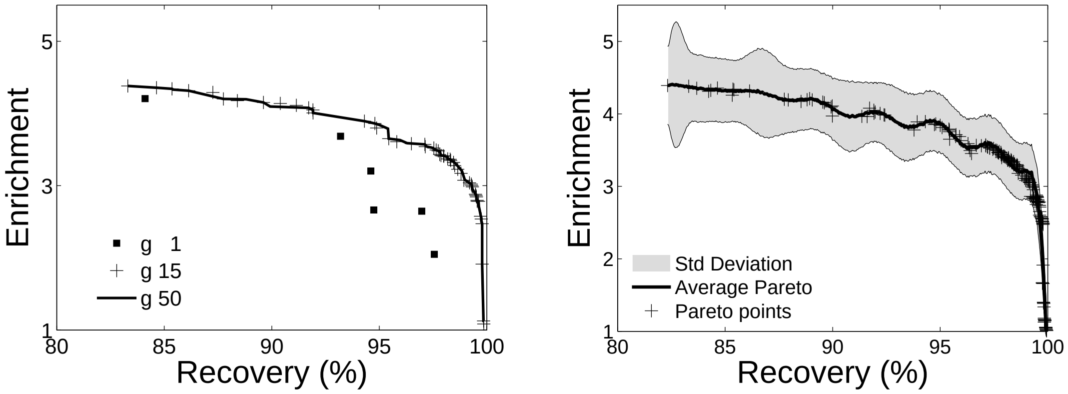
The effect of each process parameters has been previously analysed [101], and confirmed by our simulations. As γ increases, more light product is recycled to the system to purge the bed during step III. This translates into a deeper cleaning of the adsorbent, meaning that more of the adsorbed CO_2 will actually be desorbed. The bed will be more available to adsorb CO_2 in the following feed step, so that less CO_2 will be allowed to breakthrough. Both these aspects contribute to the increase of the recovery. On the other hand, the more light product is used to purge the bed, the more diluted the heavy product will be and the lower the enrichment obtained.

As the value of the high pressure is constant, the value of π_T is increased by a decrease in the low pressure value, P_L. A greater difference between P_H and P_L means a greater difference between the loading of CO_2 in the high and low pressure steps. This contributes to higher recovery and enrichment. Furthermore, the lower the pressure of the purge stream, the lower the molar flow necessary to provide a fixed purge to feed ratio (γ). Less purge gas translates into higher recovery and enrichment in the product. As γ, also the cycle time t_c has opposite effects on the recovery and the purity. As the cycle time increases, the adsorption time increases as well and so the amount of CO_2 fed to the system. This increases the chances for CO_2 to breakthrough the bed during adsorption, which decreases the recovery. Conversely, as more CO_2 is adsorbed to the bed, the product obtained will be more concentrated, e.g. a higher enrichment can be achieved. High values of γ and t_c provide high recovery and low enrichment. A high value of π_T favours both recovery and enrichment. High enrichment is achieved for low values of γ, high values of π_T and low cycle time t_c.

A strong interaction among the design variables, and the conflicting effect of some of them on the two objectives, justify the implementation of the multi-objective optimisation of the process.

5.5.1 Results of the multi-objective optimisation

The optimisation algorithm used has been described in section 5.2. The Pareto fronts have been obtained using a crossover rate of 0.7, a mutation rate 0.01, a tournament size 2, a population size of 70, and 50 generations. The Pareto curves obtained show a good spreads of solutions along the front size n has been chosen to increase reliability of the Pareto front found, as well as to encourage the spread of solution along the front.



(a)

(b)

Figure 5.19: Analysis of the performance of the MOGA in terms of convergence (a) and statistic behaviour of the Pareto front (b).

A statistical analysis is necessary to understand the average performance of the optimiser. Hence, in Fig. 5.19(b), the average front detected via Gaussian regression of 10 Pareto front generated from the same starting point is presented. Convergence is achieved after 25 generations, and a good spread of solutions along the all front can be appreciated.

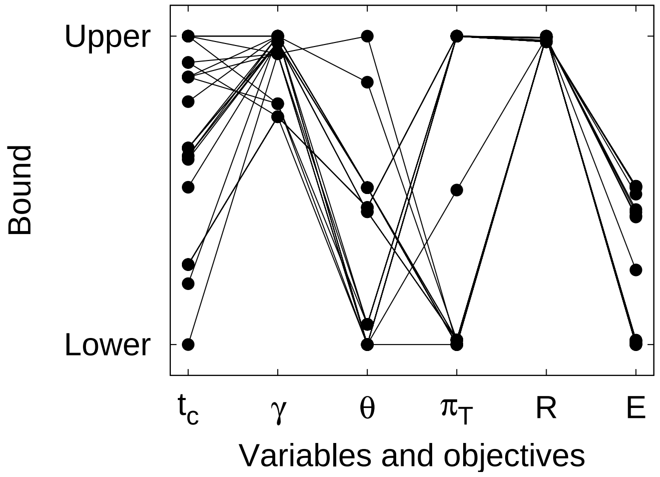
The shape of the Pareto set suggests that we can obtain high values of recovery (over 99.9 %), for values of enrichment up to almost 2.9. After this point, an increase in the enrichment is obtained with a decrease of the recovery. The values of enrichment obtained are higher than those detected manually in [101], as a wider design region has been explored. The bounds of the design space investigated are defined by $t_C \in [200 - 2000]$s, $\gamma \in [0.45 - 3]$, $\theta \in [1.15 - 50]$ L STD/hr Kg and $\pi_T \in [3.4 - 13.6]$. The bounds of the optimal design region are $t_C \in [736.5 - 1869.3]$s, $\gamma \in [0.59 - 2.9]$, $\theta \in [9.1 - 26.9]$ L STD/hr Kg and $\pi_T \in [3.5 - 13.4]$.

The results confirm the complex interaction between the 4 design variables. However, if we isolate the solution at the extremes of the Pareto front (Fig. 5.20), we can recognise some patterns. High enrichment is achieved only at high value of cycle time, low γ and θ values, and high value of π_T, as expected (Fig. 5.20(b)). A more complex relation between design variables and recovery has been noticed (Fig. 5.20(a)). High values of the recovery have been achieved for a wide range of t_c. Only high value of γ can provide a high recovery. High θ and low π_T values can allow high recovery, as well as the opposite situation, with low θ and high π_T.

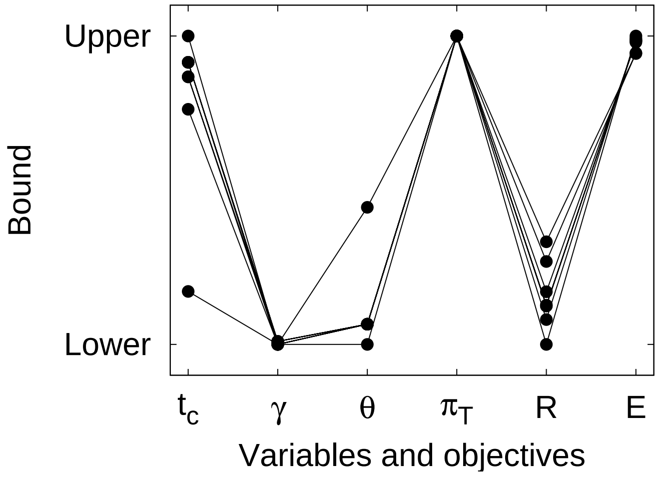
5.5.2 Effect of adsorbent on optimal performance

In section 3.2, the models of CO_2 adsorption on two different adsorbents have been presented: hydrotalcites (HTlcs) and zeolite 13X. In this section, the Pareto fronts obtained by MOGA using the two adsorbents are shown. The cycle configuration used is shown in Fig. 3.1. The comparison is shown in Fig. 5.21.

Different feeds have been used for the two processes: no H_2O is contained in the feed in the case of the zeolite, and it has been replace by N_2, as mentioned in section 3.2.2. Hence, the additional cost of the de-hydration of the feed should be taken into account when comparing the performance of the two adsorbent also under an economical point of view. Furthermore, additional costs would be needed to reduce the the temperature of the feed in the case of the zeolite from 575 K to 375 K, as equilibrium and kinetic data are available only at this lower temperature. Each of the two Pareto fronts is the average found by Gaussian regression after 10 optimisation runs, as described previously. The two Pareto sets show similar solutions in the high recovery region. When the profiles



(a) High recovery solutions



(b) High enrichment solutions

FIGURE 5.20: The two sets are visualised using a parallel co-ordinate representation. Variable domains and objective function value ranges have been normalised for presentation

starts to bend, the solutions obtained by the zeolite appear to slightly dominate. For values of enrichment above 3.8, corresponding to a purity of around 57%, the HTlcs outperform zeolite 13X. A better performance of HTlcs was expected, given the competitive adsorption of N_2 on the zeolite.

5.6 Conclusions

The multi-objective optimisation of two PSA processes has been addressed. The two case studies presented are the separation of air for N_2 production and the CO_2 capture from flue gases.

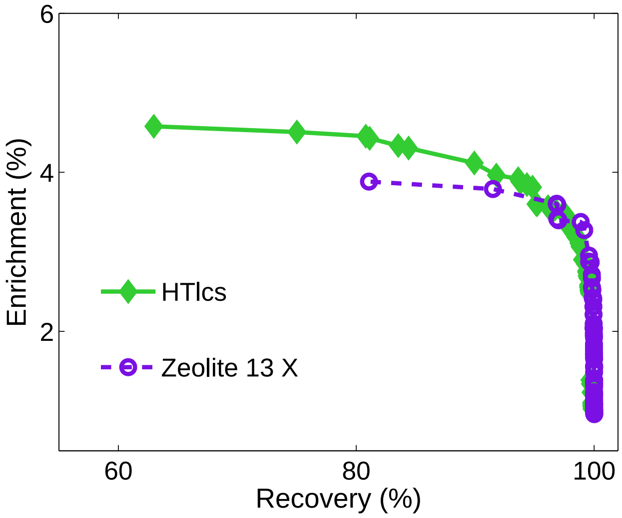


FIGURE 5.21: Comparison of the Pareto sets obtained using HTlcs and zeolite 13X

The design problem proposed is the maximisation of both purity and recovery of the desired product in a 4 dimensional design space. Two classes of optimisers can cope with the non-smoothness of the objective function: evolutionary algorithms and direct search methods (DSMs). The performance of the two approaches has been compared. To this aim, a multi-objective genetic algorithm (MOGA) has been developed. A targeted fitness function has been defined which emphasises not just points close to the current Pareto set but also those which are close to an infinite extension to that set parallel to the criteria axes. This fitness function encourages the evolutionary procedure to broaden the extent of the front. The aim is to help a design engineer identify the trade-offs between the different criteria and thereby choose the appropriate design point or even a region in the design space for further investigation.

A statistical analysis has been carried out to evaluate the average performance of the MOGA. It has been shown the MOGA is more efficient and reliable than DSMs in approximating the Pareto front for the problem of interest. The results of the comparison showed that on average DSMs are not able to detect a good approximation of the Pareto front. Moreover, the success of DSMs is very sensitive to the starting point of the optimisation, so that a pre-knowledge of the design problem would be required to obtain a good approximation of the Pareto set.

The results of the two case studies show good agreement with experimental results in both cases (see sections 5.4.4 and 5.5). Furthermore, the analysis of the performance of the MOGA indicates that it has been successfully applied, generating a Pareto front which has sufficient breadth and diversity to demonstrate this agreement. The comparison with DSMs has shown the MOGA is more efficient and reliable. [27].

The methodology introduced in this chapter allows the optimisation of the performance of a given PSA cycle according to a multi-objective procedure. The highest purity of CO_2 in the product stream achieved by the 4-bed/4-step process proposed by Reynolds *et al.* is not satisfactory. In the next chapter a flowsheet design framework is presented to address the need of synthesising PSA cycle configurations with higher performance.

Chapter 6

A Flowsheet Design Framework for PSA Cycles

The design framework proposed in chapter 5 is able to detect the optimal performance of a given PSA configuration using a multi-objective approach. The focus of the present chapter is to introduce a flowsheet design framework able to synthesise optimal PSA cycle configurations, retaining the multi-criteria point of view. The cycle configuration has a huge impact on the performance of the separation, in particular when the desired product is the more strongly adsorbed [49, 81, 105, 116], as in the case of CO_2 (see section 1.5). The task is particularly difficult due to the high combinatorial nature of the design space: number of steps and number of beds needed to perform them, beds interconnections, inputs and outputs of each step need to be specified to define a single configuration [81, 89].

In section 1.5 an overview of the performance of different cycle configurations for CO_2 capture has been provided. All the configurations proposed in the above studies have been designed according to heuristics. Although this approach allows to introduce some improvement with respect to previously proposed configurations, it does not provide a systematic methodology to synthesise optimal cycles.

Smith and Westerberg [116] developed an approach to determine the best sequence of steps among a set of known operating steps. The optimal schedule was also determined. The schedule consists of the time allocated to each of the steps, and the number of bed needed to perform the required bed interconnections. The objective of the optimisation problem was the minimisation of the number of steps needed, hence, the minimisation of the capital costs. The optimisation problem was formulated as a mixed-integer nonlinear programming (MINLP). No dynamic simulation of the PSA was included at this stage. In a subsequent publication [117] the same authors employed the same optimisation approach but included a simple integral model of PSA processes and applied it to the process of hydrogen purification from a hydrogen/methane waste stream. This approach has been classified among the simplified optimisation methods in section 4.1.1, and the weaknesses of the approach have already been discussed.

An effort to remove the assumption of knowing a feasible set of steps has been made by Agarwal *et al.* [3]. They proposed a superstructure of two-bed PSA processes to optimally synthesise an appropriate cycle for CO_2 capture solving a single-objective optimisation problem. As of now, the information about their work is limited to the abstract published for the AIChE meeting 2008 [3], and it is difficult to assess their method. However, it can be noticed that the super-structure they propose has been developed to simulate only two-beds operations. Expanding the superstructure to configurations with a higher number of beds is likely to be a difficult task.

A recent study by Nicolic *et al.* [89] addresses the optimisation of complex PSA processes including multibed configurations and multilayered adsorbents. The design variables used include the number of beds, PSA cycle configuration, and various operating and design parameters. A detailed PSA simulator is included in the framework. The main idea behind this framework is that the optimisation procedure needs to indicate the appropriate "state transitions" from one step to another: all the cycles proposed need to start from the same step, i.e. pressurisation; then, the optimiser decides the best following step, hence the state transition. The authors present a map of all the possible state transitions, called "State Transition Network" (STN) [89]: for example, from the pressurisation step (either by feed or by light product) it is possible to switch to the adsorption step only. From adsorption step, it is possible to go to many other states such as cocurrent depressurisation, blowdown, pressure equalisation, or purge by strong adsorptive. According to Nicolic *et al.*, the STN concept developed in [89] covers all of the most important states/PSA operating steps. However, no proof of completeness is provided. Despite this framework is an improvement with respect to the work of Smith and Westerberg [116, 117], to our understanding, the framework is still choosing the optimal step sequence among a set of possible already known step sequences, which are included in the STN. Furthermore, a single-objective optimisation approach has been used, despite the authors recognised the existence of trade-offs between different aspects of the performance.

In the present chapter a preliminary investigation is carried out to evaluate the viability of a multi-criteria flowsheet framework for the design PSA cycles based on evolutionary algorithms. The framework is introduced in section 6.1. A key element for the implementation of an automated design framework is a suitable representation of the design space. The representation of PSA cycle configurations used in this study is explained in section 6.2. Three evolutionary algorithms have been developed to be used within the framework, and are illustrated in section 6.3. The three algorithms adopt a *neighbour* search. The procedure developed to detect neighbour configurations is explained in section 6.4. In section 6.5, the ability of the framework to synthesise cycle configurations for CO_2 capture from flue gases is investigated.

6.1 Introduction to the flowsheet design framework: the general idea

The aim of a flowsheet design framework is to identify the best cycle configuration to perform the desired separation according to the criteria specified by the designer. The objectives of the designer are likely to be conflicting, e.g. maximisation of purity and recovery. The designer might express some indications regarding both the relevant cycle configurations (e.g. maximum and minimum number of steps, number of products etc.) and the operative conditions (e.g. maximum and minimum cycle time, feed rates etc.) to be investigated. These indications are necessary to address the search towards cycle configurations with characteristics appropriate for the separation of interest. These requirements define the design space for the cycle configurations, Y, and for the operating conditions, X. Process constraints might apply, such as the maximum pressure level, the minimum purity/recovery requirements and so forth.

In Fig. 6.1, a flowchart of the framework is illustrated. After the designer has expressed his/her preferences, an iterative procedure starts. The first step is the generation of a cycle configuration, y. The configuration generated is "viable" if it satisfies the requirements specified by the designer, i.e. if $y \in Y$. The set of operating conditions, x, which affect the performance of y is identified. The set of design variables depends on the configuration generated. The performance of y is then optimised using multiple objectives, z. The performance of the configuration, represented by its Pareto front, is fed back to the generator of cycles and taken into account when generating new configurations. The process is carried on for a certain number of iterations ($\mathrm{i_{max}}$ in Fig. 6.4).

The resulting problem could be classified as a dynamic max-max multi-objective optimisation problem:

$$
\begin{aligned}
&\max_{y \in Y} z(x,y) \\
&\qquad s.t. \\
&\qquad\qquad x = \arg \max_{x(y)\in X(y)} z(x,y) \\
&\qquad\qquad\qquad s.t. \\
&\qquad\qquad\qquad \text{Evaluation of the model at CSS;} \\
&\qquad\qquad\qquad \text{Process constraints}
\end{aligned}
\tag{6.1}
$$

where z is the vector of objectives, e.g. purity and recovery of the desired product.

The inner optimisation procedure to generate Pareto front of given cycle configurations has been developed in chapter 5. The focus of this chapter is on the development of the outer optimisation loop, whose aim is the synthesis of cycle configurations. A key element for the implementation of the automated procedure is the representation of cycle configurations within the framework. This issue is discussed in the following section.

6.2 Representation of cycle configurations

One of the most critical decisions when developing automated design tools for a particular class of problems is the choice of the search space, or, equivalently, of its representation [43]. The solution of the design problem stated in eq. 6.1 requires the development of a representation of PSA cycle configurations. The representation needs to be flexible enough to describe any possible PSA configuration. The representation also needs to contain enough details to allow the simulation of the configuration it represents. The third requirement is that it has to be managed with ease within the framework.

The key idea motivating the representation proposed here is that to synthesise new, more efficient cycles, we need to think of cycles not as sequences of known steps (e.g. adsorption with feed, desorption etc.), but to allow for all the possible bed interconnections to be explored. As noted by Guan *et al.* [42], PSA is so flexible that we may not find even a suitable word to represent every possible operation step.

A PSA cycle is described by the sequence of steps that the beds undergo. Each step is characterised by the connections that are open/closed among the beds, the feed tank, containing the gas mixture that has to be separated, and the products tanks. Since all the beds undergo the same step sequence, Σ, the whole configuration can be described by using only one bed. The same principle has been previously exploited to reduce the

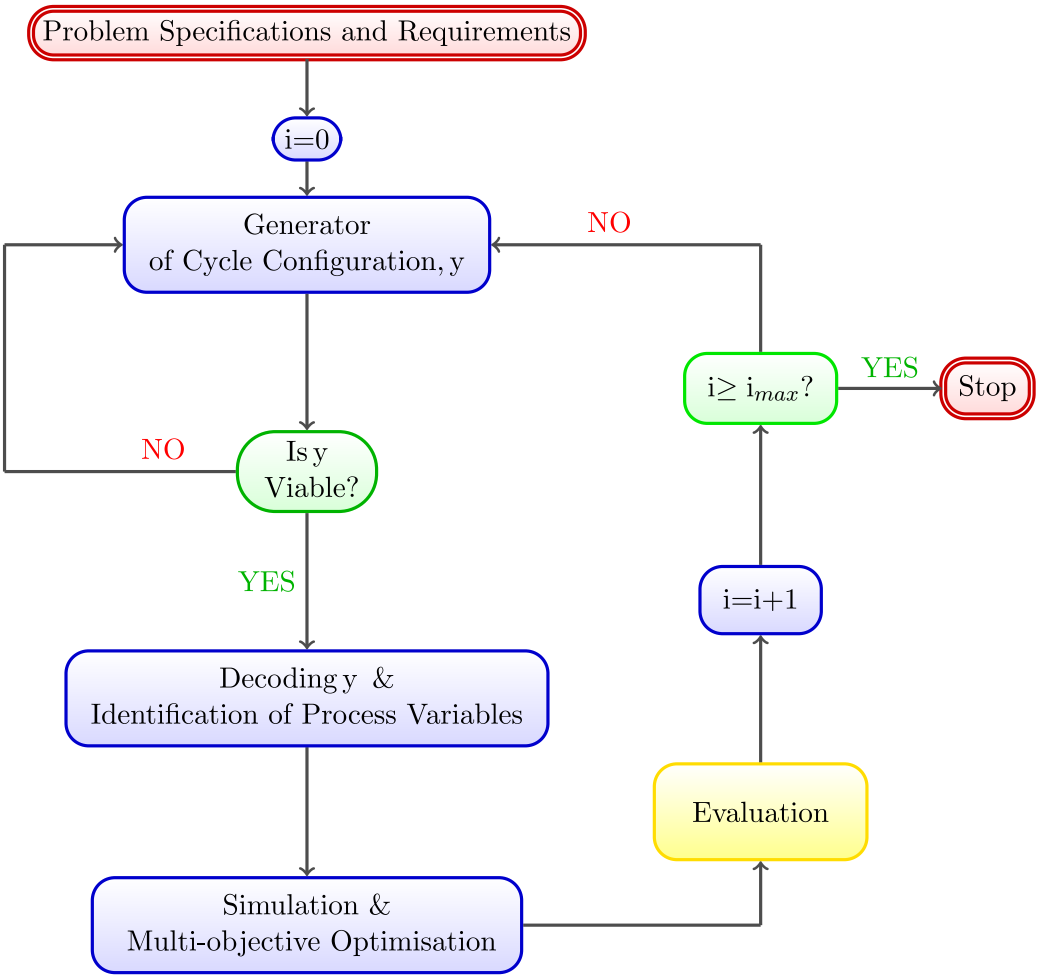


FIGURE 6.1: Flowchart representing the elements of the ideal automated design framework for PSA cycles. According to the specification introduced by the designer, a viable cycle configuration, y, is generated. Then, the operating variables, x, which determine the performance of y are identified. The optimal performance of y, in the design space defined by x, is determined. The performance criteria are then fed back to the cycle generator so as to take it into account when generating new cycles.

computational time of the simulation by adopting a "unibed" approach to reach CSS, as described in section 3.3.1.

The performance of the cycle, in terms of purity and recovery of the product, is defined by the cycle configuration (sequence Σ of steps used, bed interconnections) and operating variables, but not by the number of beds (n_b) used to perform the cycle since, as mentioned, at CSS they all behave in an identical way. On the other hand, the number of beds is necessary to evaluate capital and operating costs of the cycle. Once the sequence of steps Σ which provides the desired product has been chosen, the minimum number of beds needed can be determined using the method proposed by Smith and Westerberg in [116]. The methodology developed in this thesis allow to determine optimal cycle configurations to provide the desired quality product. Once the cycle configuration is determined by the framework, it would be possible to use the algorithm proposed by Smith and Westerberg to evaluate the minimum number of beds necessary to implement the desired cycle configuration and to carry out a cost analysis.

The representation of the cycle configuration, y, is then a tuple containing informations about the number and the sequence of steps each bed undergo during the cycle, the direction of the flow at each step and at which steps the products are withdrawn.

The sequence of steps is represented by the series of bed interconnections activated during the cycle. These connections can be represented by an adjacency matrix, A, which specifies, at each step, the step or steps from which the bed is receiving its inlet; A also has to indicate the steps from which products are withdrawn. Each row of A corresponds to the "receiving step", including product withdrawals, and each column to its "sources". Each elements a_{ij} takes the value 1 if destination i is receiving from source j of the matrix, a_{ij} =0 otherwise. The number of receiving units (n_u) is given by the summation of the number of steps (n_s) and the number of products (n_p). The bed can act as a source at each step, as well as the feed tank. The number of sources (n_{in}) is given by the number of steps plus 1, the feed. If n_c is the maximum number of units feeding a single unit, the summation of the elements of each row of A must be lower or equal to n_c. The dimension of the matrix A is $n_u \times n_{\text{in}}$.

Table 6.1: Example of adjacency matrix. It represents all the possible connections among the destinations and the sources of a PSA cycle where the number of steps is 5 (n_s=5), and the number of products 2 (n_p): the light product (*LP*) and the heavy product (*HP*). In the representation, "-1" corresponds to the feed.

	Sources					
Destinations	-1	1	2	3	4	5
I	1/0		⋯	⋯		1/0
II						
III	⋮		⋱			⋮
IV	⋮			⋱		⋮
V						
LP						
HP	1/0		⋯	⋯		1/0

An example of adjacency matrix is given in Tab. 6.1. The adjacency matrix takes into account all the possible steps that can take place in a PSA cycle, feasible and non feasible ones. In a PSA with 5 steps and 2 product (n_u=7, n_{in} =6), the matrix A has 42 elements. Hence, without any further refinement regarding feasible/infeasible connections, the number of configurations that can be generated is $2^{42} \simeq 4.4 \times 10^{12}$. It is therefore necessary to impose some rules to the interconnections that the matrix A

can describe to reduce the design space and make sure it includes the configuration of interests to the designer. This process will be illustrated in section 6.2.1. It is important to notice that the rules that the configurations generated must comply with depend on the case study of interest; however, the generic nature of A allows the representation proposed in this work to be used for any class of PSA processes.

6.2.1 Definition of a search space

The evolutionary procedure illustrated in Fig. 6.1 requires that every configuration considered satisfy the requirements expressed by the designer. Such requirements define the search space Y. Before the evaluation of a configuration y, proposed either by the cycle generator or by the neighbour generator, the algorithm checks whether $y \in Y$. If $y \in Y$ we will say that y is viable. The introduction of a viability check allows the optimisation procedure to capitalise on the designer's knowledge of the problem, addressing the search to a class of configurations that the designer knows as interesting. The size of design space is therefore resized, and the optimisation problem more manageable. The potential risk is to excessively shrink the search space, excluding potentially optimal configurations to be generated within the search.

Some examples of the requirements that the designer can express are:

- a minimum and maximum number of steps: $n_{s,min}$ and $n_{s,max}$;
- the desired number of products, n_p;
- the maximum number of bed interconnections, n_c;
- which steps can receive the feed;
- maximum number of blowdown steps within the cycle;
- whether pressure equalisation steps are allowed within the cycle, i.e. if a blowdown step can be the source of the pressurisation step;
- the output of a step cannot be recycled as input to the same step;

By applying the rules introduced in this section 6.2.1, the design space is reduced as the number of possible interconnections is limited.

To make sure that the cycle represented corresponds to a PSA operation, some requirements are necessary: during the cycle, the feed needs to be the inlet of at least one step. By convention, in the framework one of the inlet streams of the bed during the first step needs to be the feed. The cycle needs to end with the bed at high pressure, so that it is ready to start again from an adsorption step. This means that either a pressurisation step ends the cycle or that the last pressurisation step is not followed by any blowdown. At least a blowdown step should be included in the cycle, between the adsorption and the pressurisation step, so that the pressure actually swings between a high and a low level.

6.2.2 Implementation

The cycle representation introduced section 6.2 is intended to fully describe any cycle configuration. In this section it is shown how it is possible to implement the simulation of a cycle using the information contained in y.

6.2.2.1 Compact representation of the cycle configuration

The tuple y needs to contain all the necessary information to implement the cycle representation. In the practical implementation of the framework, the information needs to be expressed by a manageable mathematical structure. The information that y needs to contain are the number and sequence of steps (map of interconnections) and a vector f describing the direction of the flow for each step.

The map of interconnections could be expressed by the adjacency matrix introduced in section 6.1. However, to ease the implementation, the map of interconnections has been expressed by a matrix C, where only the activated connections are specified: the element c_{ij} is equal to $t \in 1, \ldots, n_s$, if the output stream of the bed at step t is feeding the bed at step i. The dimension of the matrix C will be $n_u \times n_c$, where n_c is the maximum number of units feeding a single unit. As A, C specifies, at each step, the step or steps from which the bed is receiving its inlet; C also has to indicate the steps from which products are withdrawn. Each row of C corresponds to the "receiving step", including product withdrawals, and each column to its "sources". Any matrix of connections C can be thought as an instantiation of the adjacency matrix A, representing all the possible connections among the destinations and the sources of a PSA cycle.

By convention, the first n_s rows of C corresponds to the steps, and the remaining rows to the products. Since the number of rows of C is equal to $n_u = n_s + n_p$, it is necessary to know the number of steps, n_s, to be able to correctly decode the information contained in C . It must also be specified the position of the desired product within the matrix C.

Table 6.2: Adjacency matrix for a Skarstrom cycle and corresponding compact representation C

	Sources				
Destinations	-1	1	2	3	4
I	1	0	0	0	0
II	0	0	0	0	0
III	0	1	0	0	0
IV	1	0	0	0	0
LP	0	1	0	0	0
HP	0	0	1	1	0

Destinations	Sources	
I	-1	-
II	-	-
III	1	-
IV	-1	-
LP	1	-
HP	2	3

The direction of the flow at each step can be described by a vector f of n_s elements, where f_i is 1 if the flow in step i is cocurrent with respect to the adsorption step, -1 otherwise.

To summarise, each configuration can be described by a tuple, $y = \langle n_s, C, f \rangle$ whose elements are the number of steps n_s, the map of interconnections C and the vector of flow directions f.

The Tables 6.2 are a representations of the simple Skarstrom cycle of Fig. 3.3. The compact representation can be obtained from the other: in both tables each row corresponds to a destination. In the extended table each column corresponds to a source: the element a_{ij} is 1 if destination i is receiving from source j of the matrix. In the compact representation, if destination i is receiving from j, then j will be one of element of row i. "-1" corresponds to the feed. The "-" entries in C corresponds to a 0 value in the actual computer representation. The vector f for the Skarstrom cycle is $f = [1, -1, -1, 1]$.

In Table 6.3, five cycle configurations, originally proposed in [105], are illustrated both by the classic graphical visualisation and by the matrix of connections C.

The generator of configurations needs to provide the three elements of y to define a configuration. The procedure that the generator of configurations applies consists of randomly selecting the number of steps n_s, the maximum number of interconnections n_c, the number of products n_p, each within the respective bounds specified by the designer. Once the above parameters are chosen, the dimensions of C are determined as C has n_s+n_p rows and n_c columns. The generator of configurations can then start to assign random values to each element of C from the set $-1, 1, \ldots, n_s$. Similarly, the n_s elements of vector f are a random series of -1 and 1. Since the first step is always adsorption with feed, as mentioned earlier, the first element of f will always be 1 for consistency.

The possibility of initialising a simulation corresponding to a particular design by extracting the information from the encoded representation is crucial, and yet, a non trivial task. The decoding of the cycle representation is discussed in some detail in the next section.

6.2.2.2 Decoding of the cycle representation to implement the simulation

In section 6.2, the encoded representation of a cycle configuration used in our framework has been introduced. A cycle configuration is fully defined once the number of steps (n_s), the bed interconnections (matrix C) and the direction of the flow (f) at every step have been defined. The above information is collected in one data structure, y. The mathematical encoding of the cycle configuration has been a key element for the implementation of the automated flowsheet design framework. In this section, we illustrate how it is possible to decode the information contained in the mathematical representation of the cycle to implement its simulation and optimisation; for the latter task, it is necessary to detect from y which design parameters need to be assigned by the embedded optimisation procedure (MOGA). The procedure to obtain x from y will be referred to as $x = var(y)$.

After the cycle generator, or the neighbour generator, proposes a viable cycle configuration, the variables x which define the behaviour of configuration y needs to be detected before the performance of y can be optimised by the embedded optimiser. For every cycle configuration, the design variables are the cycle time (t_c), the lower value of the pressure (P_L), the value of the feed flowrate (F_{in}) and the split ratios (r_s). The number of splitters to be used, and consequently the number of split ratios (n_{r_s}) to define, depends on the cycle configuration. This means that the set of operating variables x that affect the performance of y, depend on y itself. In our implementation x is described as a vector whose first elements are always t_c, P_L and F_{in}. The remaining elements of x correspond to the n_{r_s} split ratios to be determined. The number of design variables to be used by the embedded optimiser is equal to n_{r_s}+3. For each step $i \in 1, ..., n_s$, the number of split ratios to be applied to its outlet feed is given by the number of times i appears in the matrix of connection C, diminished by 1: if stream i has to be divided into 5 parts, only 4 split ratios need to be indicated. Summing the number of split ratios that need to be applied to each step we obtain n_{r_s}.

Once the size of the vector x has been determined, the embedded optimisation procedure can be initialised. It will optimise the performance of the given configuration y in the design space defined by $x \in X$. The embedded optimisation procedure requires the evaluation of the objectives at specific design points. To this end, the dynamic simulation of y needs to be implemented. This implies that further information regarding the single steps need to be extracted from $y = \langle n_s, C, f \rangle$.

The number of steps is explicitly provided by n_s, while f describes the direction of the flow at each step. The other information about the steps to be performed are contained by the matrix C. The matrix C describes in a compact way not only the map of interconnections in the cycle, but it also implicitly contains information about pressure profiles and position of mixers and splitters. Once the output/input streams are specified, also the pressure profiles are. This is illustrated in algorithm 6.1. If both input and output streams are present, the pressure will remain constant (as in adsorption and desorption); if no input is fed to the bed, then the pressure will go down as the bed releases its content (as in the blowdown step); if no output stream leaves the bed, the pressure will increase during the step (as during pressurisation). It is necessary to know the pressure profile of each step to correctly implement its simulation: although the model is the same for any step, the unknowns differ between constant and non-constant pressure profiles, as described in section 3.1.

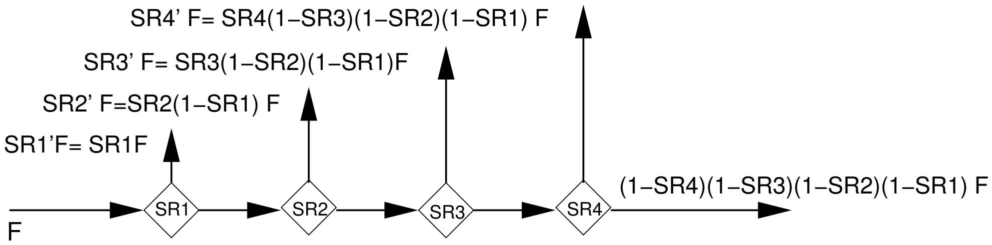
Algorithm 6.1 How to initialise the simulation after the vector x of design variables has been determined.

```
Input: $x$, $y = \langle n_s, C, f \rangle$
Output: $\Sigma$, $S$
  for $j$=1 to $n_s$ do
    if step $j$ appears in matrix $C$, then the step is not a pressurisation, and it is momentarily classified a constant-pressure step then
      $\Sigma(j) \leftarrow a$
    else if the step is pressurization then
      $\Sigma(j) \leftarrow p$
    end if
    if The step is not receiving any feed, $c_j, k$=0, for $k = 1, \ldots, n_c$, then it is a blowdown then
      $\Sigma(j) \leftarrow b$
    end if
  end for
  $S \leftarrow (1/n_s)$ {All the steps are assigned the same duration, $S$}
```

The simulator can also deduct from the information in C when to include mixers (if a unit is receiving more than one stream, the incoming streams need to be mixed before been fed) and splitters (if a stream, or a mixture of streams, feeds more than one unit, splitters are needed).

To detect the inlet flows at each step, a matrix called *Split* needs to accompany C. *Split* is a matrix of the same dimension as C, whose elements $s_{ij} \in 0, 1$ indicate the fraction of the output of step c_{ij} that needs to be fed to step i.

Let us suppose that an output stream of step $t \in 1, \ldots, n_s$, F_t, needs to be split among r receiving units. The splitter used in this thesis splits an incoming stream into two streams, hence, $r' = (r-1)$ splitters will be used and r' split ratios need to be defined, each between 0 and 1. We can indicate the split ratios as $SR_m \in [0, 1]$ where $m = 1, \ldots, r'$. Each splitter m is receiving the fraction of F_t remaining from the previous $m-1$ splitters. Such fraction can be indicated as $F_{t,m-1} = \prod_{l=1}^{m-1} (1 - SR_l) F_t$. Here, the symbol $\prod$ is used to indicate the product of a series. Hence, splitter m will be sending to the unit it is connected to a stream given by $SR_m \, F_{t,m-1} = SR_m \prod_{l=1}^{m-1} (1 - SR_l) F_t = SR'_m F_t$. Below, an example how a stream F is split is given, for r=5.



The elements $s_{ij} \in 0, 1$ of the matrix *Split* correspond to the split ratios SR'_m calculated with the procedure illustrated above from the original value SR_m proposed by MOGA for the splitter. If step t appears for the first time in matrix C in position c_{22}, and for the second time in position c_{41}, then $s_{22} = SR'_1$, and $s_{41} = SR'_2$.

Once Σ, *Split* and S are determined, the cycle described by y can be simulated as described in algorithm 6.2: at each step l, with $l \in 1, \ldots, n_s$, the inlet stream is determined using the information in C and *Split* (see lines 3, 4 and 6). The correct set of equations (M or M_p) is chosen using the information in Σ (e.g. lines 8 and 12). The output stream of step l, F_{out}, is stored in a data structure if F_{out} is to be used as input stream of another step (see line 16).

6.2.3 Validation of the representation

The results of the optimisation of 5 cycle configurations originally introduced in recent literature [105] are presented. The analysis has two aims: firstly, we test the capability

Algorithm 6.2 Dynamic simulation of one cycle of configuration y

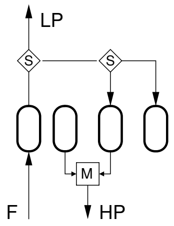
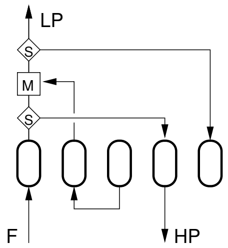
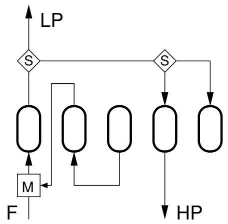
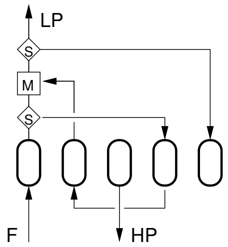
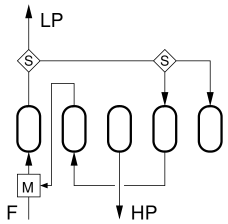
```
Input: $y = \langle n_s, C, f \rangle$, $t_c$, $F_{in}$, $P_{low}$, $Split$, $\Sigma$, $S$, Initial condition of the bed, $B_0$, in
    terms of concentrations and temperature, initial pressure $P_0$
$t_s \leftarrow S \times t_c$ {Step duration}
for $l = 1$ to $n_s$ do
    Sources of step $l$: output streams of the steps $c_{lk} \in C$ where $k = 1, \ldots, n_c$.
    Fractions of the outlet streams of $c_{lk}$ fed to step $l$: $s_{lk} \in Split$ where $k = 1, \ldots, n_c$.

    if step $l$ is fed by more than one source then
        mix the incoming streams to calculate the inlet stream ($IS$)
    end if
    if $\Sigma(l) = a \vee \Sigma(l) = b$ then
        if $\Sigma(l) = a$ then
            $P_f \leftarrow P_0$
        else if $\Sigma(l) = b$ then
            $P_f \leftarrow P_L$
        end if
        [$B(t)$, $F_{out}$, $P(t)$, $B_f$ =M($P_0$, $P_f$, $t_s$, $IS$, $B_0$, $f(l)$ )
        if Step $l$ is a source for other steps then
            Store the outlet stream of step $l$ in a vector for later use
        end if
    else if $\Sigma(l) = p$ then
        [$B(t)$, $P_f$, $P(t)$ =M$_p$($P_0$, $t_s$, $IS$, $B_0$, $f(l)$)
    end if
    $B_0 \leftarrow B_f$ {Update bed conditions}
    $P_0 \leftarrow P_f$ {Update initial pressure}
end for
```

of the cycle representation proposed in 6.2 to describe known cycle configurations. Secondly, the performance of these 5 cycles is evaluated so that it can be used as a reference against which the cycle configurations proposed by the framework can be evaluated. The five cycle configurations, together with the corresponding map of interconnections C, are illustrated in Table 6.3.

In all cycles investigated here, the first step is adsorption with the feed and the last step is a pressurisation with the light product. Cycle configuration a has been described earlier. In configurations b to e, the second step is *heavy reflux*, where the bed, at high pressure, receives a purge gas, rich in the heavy component, to the feed end. This heavy purge gas can come from step *III*, which is always a blowdown, or from step *IV*. Step *IV* is a *light reflux* step, during which the CO_2 is desorbed at low pressure using the output stream of step *I* as purge gas. The output of step *II* can either be recycled at the feed end of step *I*, or might be taken as a part of the light product and used to pressurise step *V*. The heavy product is withdrawn from the step, between *III* and *IV*, which is not feeding step *II*. All the steps are counter-current with respect to the first two.

Table 6.3: Illustration of all the cycles investigated in this section. All the configurations have been originally proposed in [105]. In the second column each cycle is represented in a compact way, in the second column the schematics of the cycle. The feed is indicated by "-1". The "-" entries in C corresponds to a 0 value in the actual computer representation.

<table>
<tr><th>Cycle</th><th>Representation</th><th>Scheme</th></tr>
<tr><td>a</td><td>
</td><td><table><tr><th>Destination</th><th colspan="2">Sources</th></tr><tr><td><i>I</i></td><td>-1</td><td>-</td></tr><tr><td><i>II</i></td><td>-</td><td>-</td></tr><tr><td><i>III</i></td><td>1</td><td>-</td></tr><tr><td><i>IV</i></td><td>1</td><td>-</td></tr><tr><td><i>LP</i></td><td>1</td><td>-</td></tr><tr><td><i>HP</i></td><td>2</td><td>3</td></tr></table></td></tr>
<tr><td>b</td><td>
</td><td><table><tr><th>Destination</th><th colspan="2">Sources</th></tr><tr><td><i>I</i></td><td>-1</td><td>-</td></tr><tr><td><i>II</i></td><td>3</td><td>-</td></tr><tr><td><i>III</i></td><td>-</td><td>-</td></tr><tr><td><i>IV</i></td><td>1</td><td>-</td></tr><tr><td><i>V</i></td><td>1</td><td>2</td></tr><tr><td><i>LP</i></td><td>1</td><td>2</td></tr><tr><td><i>HP</i></td><td>4</td><td>-</td></tr></table></td></tr>
<tr><td>c</td><td>
</td><td><table><tr><th>Destination</th><th colspan="2">Sources</th></tr><tr><td><i>I</i></td><td>-1</td><td>2</td></tr><tr><td><i>II</i></td><td>3</td><td>-</td></tr><tr><td><i>III</i></td><td>-</td><td>-</td></tr><tr><td><i>IV</i></td><td>1</td><td>-</td></tr><tr><td><i>V</i></td><td>1</td><td>-</td></tr><tr><td><i>LP</i></td><td>1</td><td>-</td></tr><tr><td><i>HP</i></td><td>4</td><td>-</td></tr></table></td></tr>
<tr><td>d</td><td>
</td><td><table><tr><th>Destination</th><th colspan="2">Sources</th></tr><tr><td><i>I</i></td><td>-1</td><td>-</td></tr><tr><td><i>II</i></td><td>4</td><td>-</td></tr><tr><td><i>III</i></td><td>-</td><td>-</td></tr><tr><td><i>IV</i></td><td>1</td><td>-</td></tr><tr><td><i>V</i></td><td>1</td><td>2</td></tr><tr><td><i>LP</i></td><td>1</td><td>2</td></tr><tr><td><i>HP</i></td><td>3</td><td>-</td></tr></table></td></tr>
<tr><td>e</td><td>
</td><td><table><tr><th>Destination</th><th colspan="2">Sources</th></tr><tr><td><i>I</i></td><td>-1</td><td>2</td></tr><tr><td><i>II</i></td><td>4</td><td>-</td></tr><tr><td><i>III</i></td><td>-</td><td>-</td></tr><tr><td><i>IV</i></td><td>1</td><td>-</td></tr><tr><td><i>V</i></td><td>1</td><td>-</td></tr><tr><td><i>LP</i></td><td>1</td><td>-</td></tr><tr><td><i>HP</i></td><td>3</td><td>-</td></tr></table></td></tr>
</table>

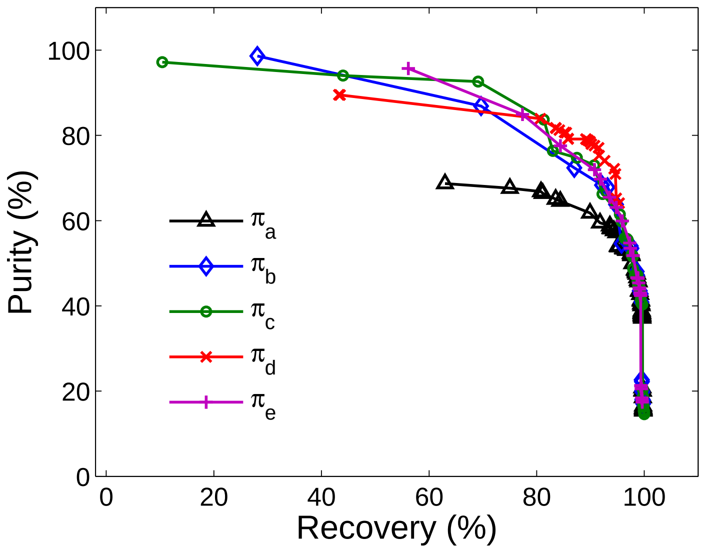


FIGURE 6.2: The Pareto fronts of all the cycle configurations investigated (see Table 6.3) are shown.

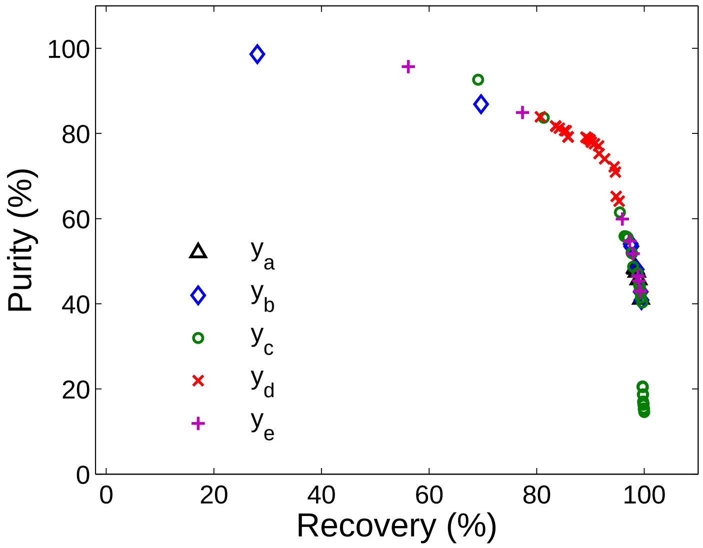


FIGURE 6.3: Overall Pareto Π front obtained as the set of non-dominated points of the population constituted by the Pareto points of each cycle configuration investigated:$\Pi = \mathrm{dom}(\pi_a, \pi_b, \pi_c, \pi_d, \pi_e)$. The aim is to show how the best performance of different cycles compare under a multi-criteria point of view.

Using the tuple y to represent the five cycles, it was possible to initialise the simulation and optimisation of their performance. A Pareto set for each cycle configuration has been generated. All five Pareto fronts, π_j where $j = a, \ldots, e$, are shown in Fig. 6.2. A statistical analysis is necessary to understand the average performance of the optimiser. However, our previous experience with MOGA in chapter 5 showed that with a high population size, there is a very small standard deviation between the Pareto fronts generated with the given set of parameters. Furthermore, the Pareto fronts found with a large population size dominates those obtained with a lower population size (see Chapter 5). Accordingly, a high population size (n=60) has been chosen to avoid multiple optimisation runs.

To ease the comparison among the 5 Pareto fronts, the overall set Π of non-dominated points, from the set of points defined by the union of the five individual Pareto sets, is shown in Fig. 6.3. In mathematical terms, the operation can be described as: $\Pi = \text{dom}(\pi_a, \pi_b, \pi_c, \pi_d, \pi_e)$. The cycle configuration which corresponds to each point in this set is shown in the legend. Our results are qualitatively coherent with the conclusions by Reynolds *et al.* [105]. However, we did not replicate their results quantitatively as our models differ. The effect of the approximation is stressed as the number of beds included in the cycle increases.

All 5-step configurations outperformed the 4-step cycle. The latter could only achieve a product purity of 60%. This result justifies the investigation of cycles where also the heavy recovery step is included (step *II* in all the 5-step cycles).

Comparing the results from cycle b and cycle c, the effect of the management of the output of the heavy recovery step (step *II*) can be deduced: when this is mixed with the output of step *I*, as in cycle b, lower recoveries are achieved as some CO_2 can be lost in the light product. The higher values of purity have been achieved by cycle b. By comparing cycle b and cycle c on one side, and cycle d and cycle e on the other, we can understand the effect of the different arrangements for the input to step *II* and the heavy product. When the purge for the heavy recovery step comes from the blowdown (as in cycle b and cycle c), higher purity and higher recoveries can be achieved, as it becomes evident looking at both extremes of the Pareto set. However, the Pareto set suggests that the best trade-offs between purity and recovery are achieved by cycle d, which dominates the region where the Pareto set bends. This analysis would encourage the cycle-generator in the design framework to propose new cycles which keep the characteristics of cycle d and cycle b, maintaining what they have in common (input to step *I* and *II*), and proposing intermediate arrangements of the input to step *II* and the heavy product, in which the two cycles differ.

6.3 Evolutionary algorithms used within the framework

Three evolutionary algorithms have been implemented for the multi-objective optimisation of PSA configurations: a simple evolutionary algorithm (EA), a simulated annealing procedure (SA) and a population based algorithm (POP). A theoretical background to evolutionary algorithms has been provided in chapter 5. In this section details of the three evolutionary procedures are provided. The evolutionary algorithms used in this chapter adopt a (1+1) evolution strategy: a single parent that generates single offspring [20]. The child is a neighbour of the parent. The procedure implemented to generate a neighbour is illustrated in section 6.4.

At each iteration i, a Pareto front π_i is generated by optimising the cycle configuration y_i. We will refer to this operation as $\pi_i = \text{moga}(y, x)$, where x is the set of operating variables that affect the performance of y; x is detected by the procedure illustrated in section 6.2.2.2, that is indicated as $x = \text{var}(y)$. The overall Pareto front at iteration i, Π_i, is then obtained as the set of non-dominated points of the union of Π_{i-1} and π_i. To summarise the operation in mathematical terms, the following expression has been defined: "$\Pi_i = \text{dom}(\Pi_{i-1}, \pi_i)$". At iteration 1, the overall Pareto front Π_1 coincides with the Pareto front of y_1: $\Pi_1 \equiv \pi_1$. We will say that the configuration y_i has contributed to the overall Pareto front if Π_i is different from Π_{i-1}: some of the points of π_i have been included in the overall Pareto.

6.3.1 Simple Evolutionary Algorithm

A simple evolutionary procedure (EA) has been implemented to search for optimal cycle configurations: at each iteration i a neighbour configuration y_i is generated from the previous investigated configuration y_{i-1} by the neighbouring procedure, and the overall Pareto front $\Pi_i = \text{dom}(\Pi_{i-1}, \pi_i)$ is obtained. The process is repeated for a given number of iterations (i_{max}), as illustrated in algorithm 6.3.

The search carried out by this algorithm is *random* in the sense that every move is accepted (i.e. the new configuration to be investigated is always the neighbour of the previous configuration), as shown in line 7 of algorithm 6.3, regardless of whether y_{i-1} contributed to the overall Pareto front or not. The randomness of the search prevents from capitalising on good solutions previously found, and might proceeds evolving worse configuration than previously found. On the other hand, it prevents from getting trapped in local optima by accepting also moves to inferior neighbouring solutions. An improvement is represented by the simulated annealing search, as illustrated in the next section.

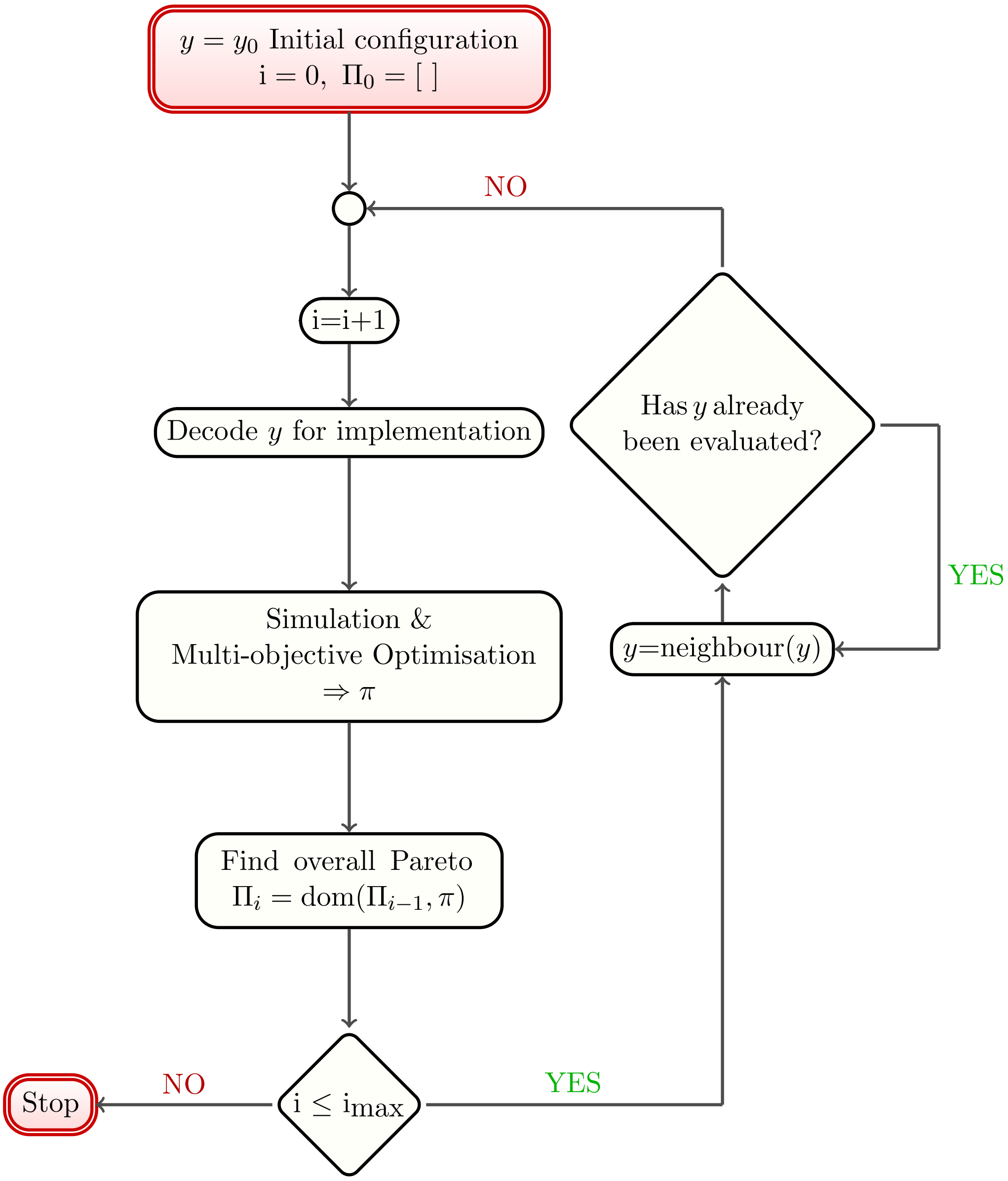


FIGURE 6.4: Simple evolutionary algorithm for the optimisation of a cycle configuration y.

Algorithm 6.3 Simple evolutionary algorithm (EA)

Output: Overall Pareto front Π

Input: y_0 initial configuration, $i_{\max}$ number of iterations

```
$y \leftarrow y_0$, $\Pi_0 \leftarrow [\,]$, $i \leftarrow 0$
while $i < i_{\max}$ do
    $i \leftarrow i + 1$
    $x \leftarrow$var($y$)
    $\pi \leftarrow$moga($x$, $y$) {Pareto front of configuration $y$}
    $\Pi_i \leftarrow$dom($\Pi_{i-1}$, $\pi$) {Overall Pareto front}
    $y \leftarrow$neighbour($y$) {Neighbour generation}
end while
$\Pi \leftarrow \Pi_i$
```

6.3.2 Simulated Annealing

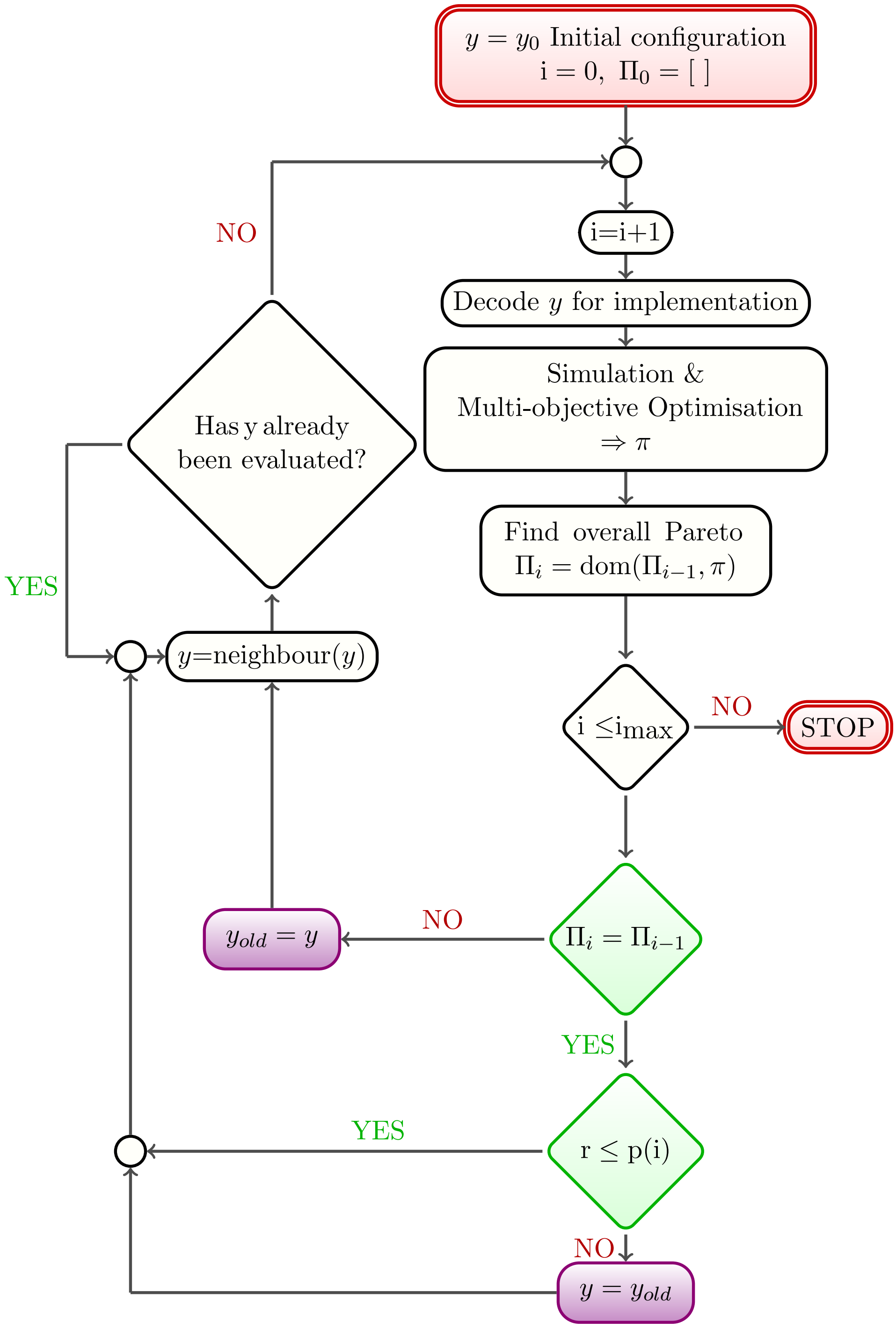


FIGURE 6.5: Simulated annealing procedure for the optimisation of a cycle configuration y.

Simulated annealing (SA) is a neighbour search approach that can provide excellent solutions to single and multiple objective optimisation problems [119]. SA usually is less computationally requiring than population-based (e.g. genetic) algorithms since it finds optimal solutions using a point-by-point iteration rather than a search over a population of individuals [119]. SA starts with an initial solution and iteratively moves towards other existing solutions. It tries to avoid getting trapped in local optima by accepting moves to inferior neighbouring solutions under the control of a randomized scheme [86]. At iteration i, if the current solution y_i brings an improvement of the objectives, the next neighbour will be derived from y_i. If y_i does not improve the objective the next solution will still be derived from y_i if $r < p(i)$, where r is a uniform random number between 0 and 1, and $p(i)$ is a function which takes values between 0 and 1 which decreases as the iterations increases. Hence, at each iteration there is a probability of making an uphill move (e.g. accepting an bad solution to generate a neighbour), so as to prevent the process to get stuck in local optima; the probability of doing an uphill move decreases as the process goes on. The procedure is illustrated in algorithm 6.4 in lines 7 to 17.

Naderi *et al.* [86] define $p(i)$ as $e^{(-\Delta/T)}$, where Δ is the change in the objective function due to solution y_i, and T is a control parameter, called *temperature*. Such definition recalls the original idea behind simulated annealing, which was born by analogy to the annealing process used in making spin glasses [77]. Annealing involves heating and controlled cooling of a material to increase the size of its crystals and reduce their defects. A slow cooling schedule is necessary to give the atoms more chances of finding configurations with lower internal energy than the initial one without getting trapped in a local minimum of the internal energy.

In this study the function p has been expressed as $p(i) = p_b \times p_r^{(i-1)}$, where p_b=0.5, p_r=0.4 and i is the number of iterations, so that the first iteration will accept a bad move in the 50% of cases. p_r defines the rate of decay of the probability of accept a bad solution: the higher p_r, the slower the decay. The algorithm is illustrated in algorithm 6.4.

6.3.3 Population Approach

Alternatively, I have considered a population based approach (POP). In this algorithm, at each iteration the parent configuration is selected among the population of all the previously investigated configurations. Accordingly, the size of the population is equal to the number of iterations implemented, and each configuration is considered an individual of the population. As in a genetic algorithm, the selection process is done through a tournament procedure. In our case, a tournament selection of size two has been used: two random individuals are picked up from the population, and the fittest one is used to generate the next member of the population. The definition of the fitness is the most important element of the procedure.

Algorithm 6.4 Simulated Annealing algorithm (SA)

```
Output: Overall Pareto front $\Pi$
Input: $y_0$ initial configuration, $\mathrm{i}_{\mathrm{max}}$ number of iterations
$y \leftarrow y_0$, $\Pi_0 \leftarrow [\,]$, $i \leftarrow 0$;
while $i \leq i_{\mathrm{max}}$ do
    $i \leftarrow i + 1$
    $x \leftarrow \mathrm{var}(y)$
    $\pi \leftarrow \mathrm{moga}(y,\ x)$
    $\Pi_i \leftarrow \mathrm{dom}(\Pi_{i-1},\ \pi)$
    if Configuration $y$ contributed to the overall Pareto front then
        $y_{\mathrm{old}} \leftarrow y$
        $y \leftarrow \mathrm{neighbour}(y)$
    else {Configuration $y$ has not contributed to the Pareto front}
        Generate a random number $r \in [0, 1]$
        if $r \leq p(i)$ then {$p(i)$ is a decreasing function of the number of iterations}
            $y \leftarrow \mathrm{neighbour}(y)$
        else
            $y \leftarrow \mathrm{neighbour}(y_{\mathrm{old}})$
        end if
    end if
end while
$\Pi \leftarrow \Pi_{i_{\mathrm{max}}}$
```

Different definitions of fitness could be considered. For instance, each design point, defined by both a configuration y and a set of optimal operating conditions x, could be considered as a single individual, and be given a value of fitness with a similar procedure as in the MOGA presented in Chapter 5. To select a parent, then, two random design points would be picked and the cycle configuration of the fittest would be used to generate the next configuration. This definition of fitness, however, would not carry information about the overall performance of the configuration but only about one particular design point, thus not reflecting our intention to evolve the overall Pareto front.

Alternatively, each configuration y_i could be considered as an individual, and the value of the fitness could be proportional to the number of points of π_i belonging to the overall Pareto front Π_i. This definition would represent an improvement with respect to the previous, but it would not be able to differentiate among cycle configurations whose Pareto fronts have no point in common with the overall Pareto front. In some sense, it would be a measure of how good a configuration performs in the objective space, but not of “how bad”. Hence, there might be the risk of excessively penalising potentially interesting configurations in the evolutionary process.

To address the above problems, a third definition of fitness has been adopted which allows the fitness to be a function of the Pareto sets. Each configuration y_i is considered as an individual of the population. The fitness needs to be representative of how the configuration y_i performs in the objective space. To this end, I defined the fitness of each configuration of the population, y_j with $j = 1, \dots, i$, as the average of the fitness of the points of its Pareto front π_j with respect to the current overall Pareto front Π_i. The fitness of the single points is evaluated as in the MOGA procedure. If Π_i is different from Π_{i-1}, which means that configuration y_i contributed to Π_i, the fitness of all the configurations in the population (i.e. $\{y_1, y_2, \dots, y_i\}$) is recalculated with respect to Π_i (see line 8 of algorithm 6.5). Conversely, if $\Pi_i \equiv \Pi_{i-1}$, only the fitness of y_i has to be evaluated (see line 11 of algorithm 6.5).

The overall procedure to implement POP is illustrated in Fig. 6.6 and in algorithm 6.5. In Table 6.4 it is shown the comparison between five individual Pareto fronts π_j, for $j = 1, \dots, 5$, and the overall Pareto front at the fifth iteration Π_5. The fitness of each configuration y_j reflects the relative position of π_i with respect to Π_5.

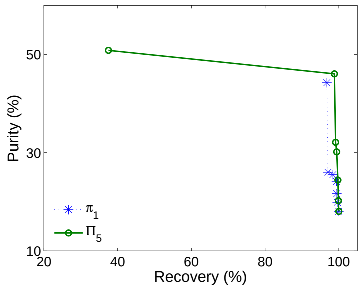
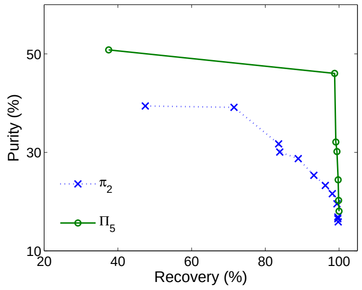
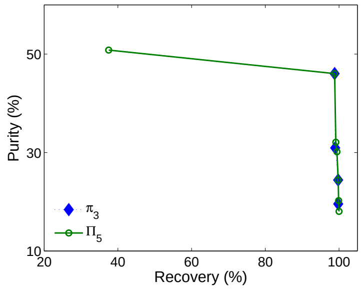
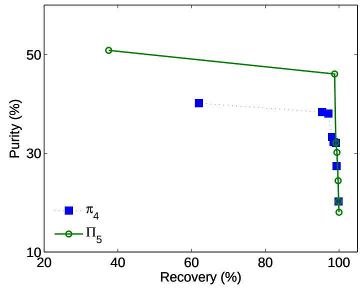
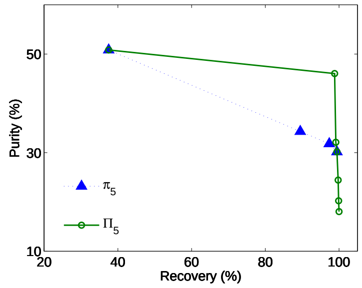
Algorithm 6.5 Population based evolutionary algorithm (POP)

```
Output: Overall Pareto front $\Pi$
Input: $y_0$ initial configuration, $i_{\max}$ number of iterations
 1: $y \leftarrow y_0$, $\Pi_0 \leftarrow [\,]$,pop$_0 \leftarrow [\,]$, $i \leftarrow 0$;
 2: while $i < i_{\max}$ do
 3:     $i \leftarrow i + 1$
 4:     $x \leftarrow$var($y$)
 5:     $\pi \leftarrow$moga($x$, $y$)
 6:     $\Pi_i \leftarrow$dom($\Pi_{i-1}$, $\pi$)
 7:     pop$_i$= $\langle$pop$_{i-1} \cup \langle y, \pi \rangle\rangle$
 8:     if $\Pi_i \neq \Pi_{i-1}$ then {The overall Pareto front changed}
 9:         Evaluate the fitness of all the configurations with respect to $\Pi_i$
10:     else
11:         Evaluate the fitness of configuration $y$
12:     end if
13:     $y_p \leftarrow$select(pop$_i$)
14:     $y \leftarrow$neighbour($y_p$)
15: end while
16: $\Pi \leftarrow \Pi_i$
```

6.4 Generation of neighbour configurations

All the evolutionary algorithms developed here rely on the concept of a neighbour search. The algorithms use different criteria to determine which previously evaluated configuration has to be used to generate the next configuration to be examined. In particular, in the simple evolutionary algorithm, the neighbour configuration is always generated from the previous one, independently of whether this has contributed to the overall Pareto front or not. In the simulated annealing procedure, the previous configuration is surely

Table 6.4: Overall Pareto front obtained using the population based algorithm at the fifth iterations, Π_5. The individual Pareto fronts of the 5 configurations in the population are also shown as π_j for $j = 1, \ldots, 5$. The better the performance of the configuration, the lower the fitness. It is evident how the value of the fitness is representative of how each individual of the configuration $_j$ performs in the 2-dimensional objective space

Iteration	Pareto	Fitness value
1		0.0238
2		0.1080
3		0.0049
4		0.0972
5		0.0312

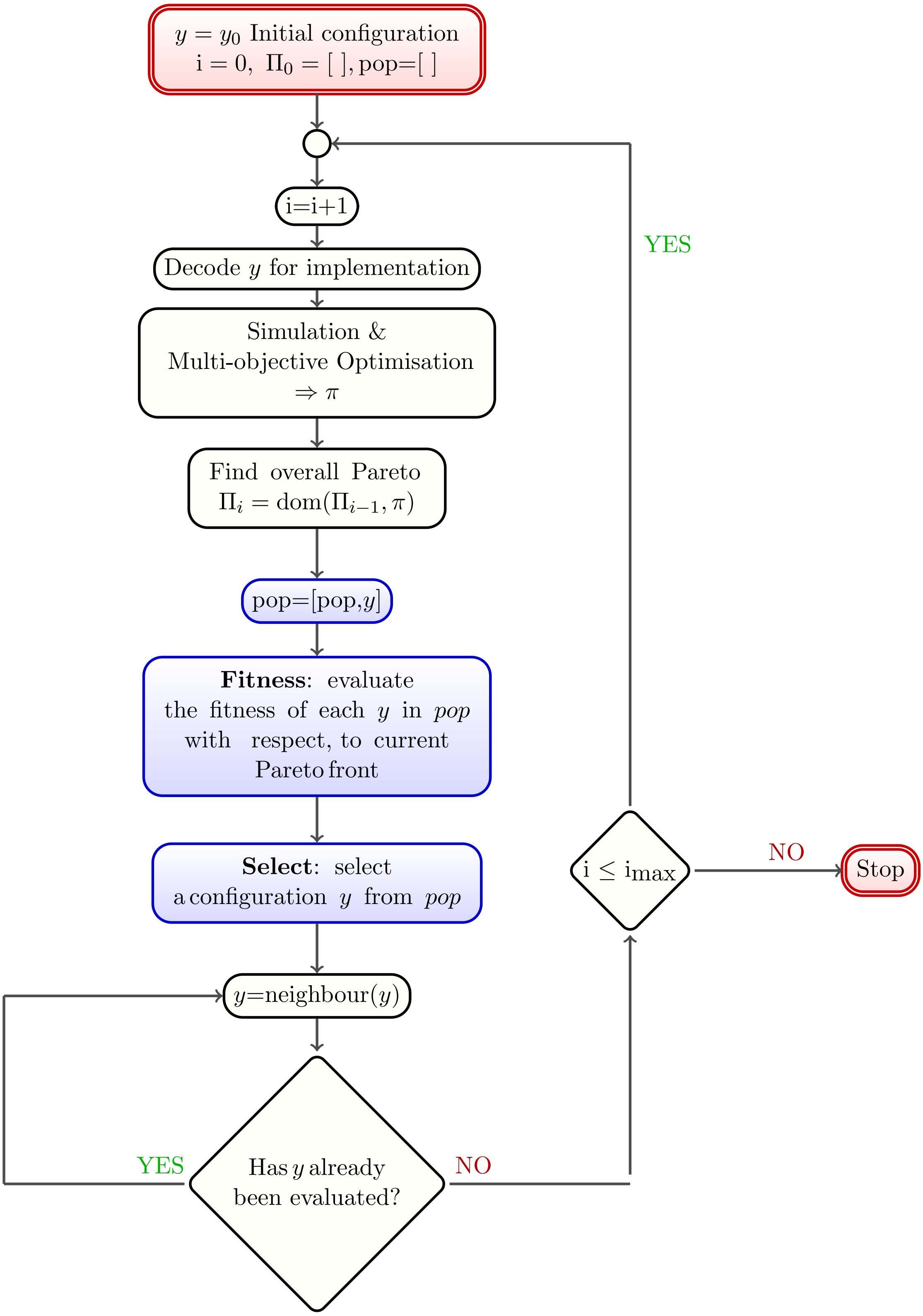


FIGURE 6.6: Population based evolutionary algorithm for the optimisation of a cycle configuration y.

used to generate the next if it has contributed to the overall Pareto. However, if the configuration has not contributed, there is still a probability to use it as a parent. This probability decreases as the iterations progress. In the population based approach, the parent configuration is chosen among all the previously investigated configurations using a fitness/selection procedure.

The definition of a neighbour needs to ensure that all the design space can be explored moving from neighbour to neighbour, starting from any configuration. In fact, as mentioned in section 5.1.5, for an evolutionary algorithm to be convergent, it is necessary to ensure that any solution $x^{'} \in X$ can be produced from every $x \in X$ in a finite number of iterations. The definition of a neighbour is then key to ensure the convergence of the algorithms.

Let us suppose that we need to generate a neighbour from a cycle configuration $y = \langle n_s, C\, f\rangle$. According to our definition, a neighbour y_n of y is a configuration which differ from it by only one step. y_n can be obtained from y by changing one of its existing step (*move a*), adding (*move b*) or subtracting (*move c*) one step. The three options are summarised in Table 6.5.

Table 6.5: Summary of the possible ways of generating a neighbour

Move	A neighbour y_n can be obtained from y by:
a	changing one of the existing step of y
b	adding one step to y
c	subtracting one step to y

Move b is possible only if $(n_s + 1) < n_{s,max}$, while it is necessary that $(n_s - 1) > n_{s,min}$ for move c to be implemented. A stochastic procedure is used to decide which of the three moves is executed, according to the procedure illustrated in algorithm 6.6. A higher probability has been given to *move a* to encourage the algorithm to take into account the effect of small changes to an existing configuration. Equal probabilities have been given to moves b and c. However, an analysis should be carried out to establish the effect of these parameters.

Let us suppose we have to generate a neighbour $y_n = \langle n_{s,n}, C_n, f_n\rangle$ from a configuration $y = \langle n_s, C, f\rangle$, where y_n needs to have the same number of steps as y (move a): $n_{s,n}$=n_s. The first step is the generation of an intermediate configuration, $y^* = \langle n_s^*, C^*, f^*\rangle$, having the same dimensions as y. C_n and f_n are equal to C and f, but for a random row $j \in 1, ..., n_u$, which is substituted by the j-th row of C^* and f^*, respectively.

Algorithm 6.6 Generation of a neighbour configuration y_n from y

```
Input: $y = \langle n_s, C, f \rangle$, $n_{s,max}$, $n_{s,min}$
Output: $y_n = \langle n_{s,n}, C_n, f_n \rangle$
  Repeat
    r=random number;{Two random numbers between 0 and 1 are generated}
    r2=random number;
    if r<0.85 then
      We create a neighbour $y_n$ having the same number of steps as $y$
    else if  r2>0.5 $\vee$ $n_s$=$n_{s,min}$ then
      Create neighbour $y_n$ by adding one step to $y$
    else if r2$\leq$0.5 $\vee$ $n_s$=$n_{s,max}$ then
      Create neighbour $y_n$ by subtracting one step from $y$
    end if
  Until A feasible configuration $y_n$ is found
```

The stochastic procedure assures at the same time a close relation between y and y_n, and that at each iteration any of the rows of C and f can be substituted with another possible step, any configuration can be obtained by another in a finite number of iterations.

6.5 Framework Results: Synthesis of optimal cycle configurations for CO_2 capture

The design problem addressed in this section is the synthesis of optimal VSA configurations for the capture of CO_2 from flue gases. The objective is the simultaneous optimisation of the recovery and purity of CO_2 in the product stream. In mathematical terms, the problem can be expressed as follow:

$$\begin{aligned}
&\max_{y \in Y} \text{Recovery \& Purity of } CO_2 \text{ at CSS} \\
&\; s.t. \\
&\; x = \arg \max_{x(y) \in X} \text{Recovery \& Purity of } CO_2 \text{ at CSS} \\
&\; s.t. \\
&\quad \text{Evaluation of the model at CSS;}
\end{aligned} \tag{6.2}$$

The design space Y is given by cycle configurations, y, which obey to the following rules:

1. the minimum and maximum number of steps: $n_{s,min}$=4 and $n_{s,max}$=6;
2. the number of products is 2 ($n_{p,max} = n_{p,min}$=2), e.g. the light and heavy product;
3. the maximum number of bed interconnections is 2 (n_c=2);

4. step 1 is always receiving the feed (i.e. c_{11}=-1);
5. all the steps but the last one can be sources: the last step is a pressurisation;
6. one, and only one, of the steps must be a blowdown; the first and the last step cannot be blowdowns according to the previous points (i.e. c_{bk}=0 where $b \in 2,\ldots,n_s$-1 and k=1, 2);
7. the blowdown step cannot be the source of the pressurisation step, as we are not interested in saving pressurisation costs by introducing a pressure equalisation step in a vacuum swing adsorption cycle;
8. the light product can be withdrawn from the steps before the blowdown;
9. the heavy product can be withdrawn from the steps after the blowdown minus the pressurisation;
10. the output of a step cannot be recycled as input to the same step;

The rules that define Y have been imposed to include in our search space the cycle configurations proposed by Reynolds *et al.* in [105]. In [105] nine high temperature heavy reflux PSA cycles have been proposed for the capture of CO_2 from flue gases. Despite some of the cycles proved to be efficient, the authors acknowledge the lack of a systematic design strategy on how to best configure complex PSA cycles [81]. I used the class of configurations proposed by Reynolds *et al.* to test the ability of our automated flowsheet design framework to better the results obtained by the heuristic design approach used in [105]. Hence, the rules imposed to our configurations ensures that all the cycles investigated starts with a feed step, contain only a blowdown step and ends with a pressurisation step, as in [105], but a wider range of bed interconnections are allowed than in [105], as any step can be used as source to another.

The rules imposed require some extra care when generating a neighbour with the procedure illustrated in section 6.4: it is necessary to make sure that the neighbouring procedure can change any of the elements of the parent configuration without necessarily generating non-viable configurations. For example, if a blowdown step is added when generating y_n from y using move a, a further row has to be substituted to remove the blowdown step already contained in y. Similarly, if the change implies removing the blowdown step from y, a further row has to be to y_n to make sure that is contains a blowdown step. Before the neighbour configuration is sent to the simulation/optimisation stage, the viability check is carried out to make sure the new neighbour follows the desired rules.

Each configuration y will be optimised by MOGA, introduced in chapter 5 in the design space defined by $x = \text{var}(y)$. For every cycle configuration, the design variables are the cycle time (t_c), the lower value of the pressure (P_L), the value of the feed flowrate (F_{in}) and the split ratios (r_s). The design space of the operating conditions X is defined by the lower and upper bounds imposed to each of these variables. In our case study, the boundaries have been defined by $t_c \in$ [200, 2000] s, $\text{P}_L \in$ [0.1, 0.4] atm, $F_{in} \in$ [10^{-7}, 10^{-5}] kmol/s and $r_s \in$ [0.002, 0.99]. The value of the high pressure has been kept constant, $P_H = 1.36\,\text{atm}$, as in [105]. The only constraint is the evaluation of recovery and purity at CSS, which is identified by dynamic simulation. In sections 6.5.1 and 6.5.2 the results obtained by the framework are presented. Although the model has been modified to reduce computational costs, the simulation and optimisation of a configuration is still time demanding. The computational burden is an even bigger issue within the flowsheet framework, where more configurations have to be evaluated by the embedded MOGA procedure at every run. Hence, in this chapter a preliminary analysis of the results is proposed, rather than a full statistical investigation.

The design problem stated in eq. 6.2 has been firstly solved using the EA algorithm. One run of the algorithm has been completed, allowing 10 iterations of the external loop (i_{max}=10). The computational time required by this run was of 25 days. The results of this run are presented in section 6.5.1.

Due to the excessive computational time required by the previous experiment, the embedded MOGA procedure has been resized so as to be less time requiring, by allowing only 5 generations (ng = 5), and a population size of 40 (n=40). This allowed to do multiple runs for each of the three algorithms proposed, with i_{max}=20. The results are presented in section 6.5.2.

In both cases, the overall Pareto front shown in Fig. 6.3 obtained from the five cycle configurations of Table 6.3 is used as a reference to evaluate the results of the framework.

6.5.1 Results obtained using a non-resized embedded MOGA

The simple evolutionary algorithm EA has been allowed to make 1 run with i_{max}=10, i.e. 10 configurations explored. A low number of iterations has been allowed due to the high computational burden imposed by the embedded optimisation process. Each cycle configuration has been optimised using ng=25, n=60, cr = 0.7 and mr = 0.01 as MOGA parameters.

Given the low number of iterations allowed, the design space has been further restricted to help the algorithm to better explore the design space: it has been imposed that in all

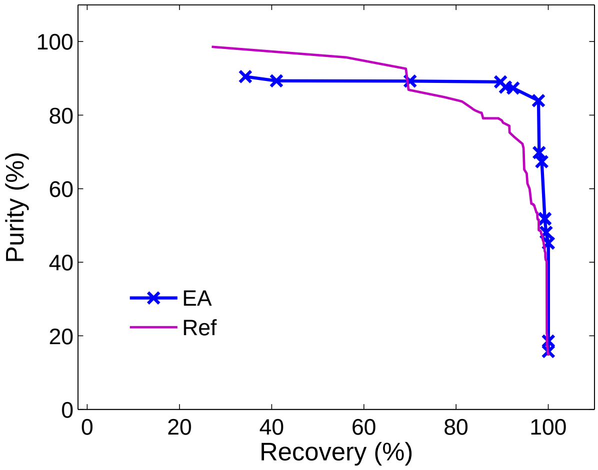


FIGURE 6.7: Pareto front obtained by the simple evolutionary algorithm (EA). The comparison with the overall Pareto front, *Ref*, obtained by the configurations in Table 6.3, shows that EA suggests many solutions with a better performance.

the configurations generated by the framework the inlet streams of the first step, aside the feed, is the outlet stream of the second step, as in configurations c and e in Table 6.3: c_{11}=-1, $c_{12} = 2$.

In Fig. 6.7 the comparison between the results of EA and the overall Pareto fronts obtained by the configurations in Table 6.3 is shown. The results are directly comparable as the MOGA has been used with similar parameters. The overall Pareto front obtained by the EA matches the reference Pareto in the high recovery region, and dominates where the two Pareto fronts bends, hence, in the area of the most desirable performance.

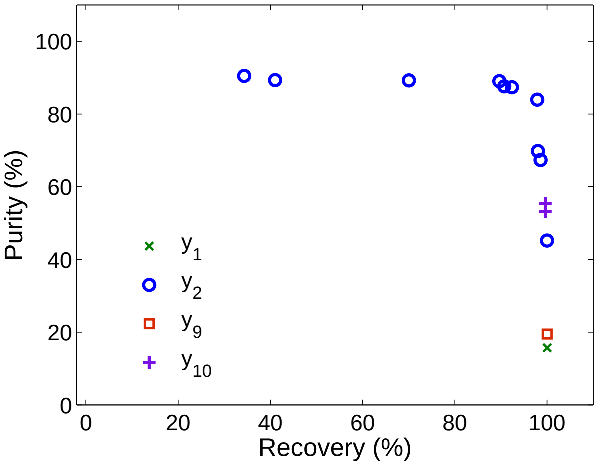


FIGURE 6.8: The final Pareto front obtained by EA using the full MOGA procedures consists of design points belonging to different configurations.

For recovery values under 65%, the reference Pareto dominates, but the improvement of the purity in this region would not justify the huge loss in the recovery. This is an important result, which shows the ability of the framework to propose cycle configurations which outperform the configurations in Table 6.3.

Fig. 6.8 shows how different configurations contribute to the overall Pareto front. While more than one configuration occupies the region of high recovery/low purity, only one provides high purity solutions. A schematic of this configuration is shown in Fig. 6.9. Interestingly, the product is not withdrawn from a blowdown step, and the inlet feed of the step which produces the heavy product is a mixture of the CO_2 enriched stream coming from the blowdown, step *II*, and the stream coming from step *IV*, a light product reflux step. The feed to the product step, step *III*, is a highly enriched CO_2 stream from the feed end of step *II*, which is operated countercurrently, and the outlet stream of step *IV*. During step *IV*, the bed is purged with the light product, which allows to push the remaining CO_2 out of the bed. Since the pressurisation step is operated countercurrently, the heavy product used to pressurise it does not decrease the capacity of the bed: the CO_2 contained in the feed is mainly adsorbed by the feed end of the bed.

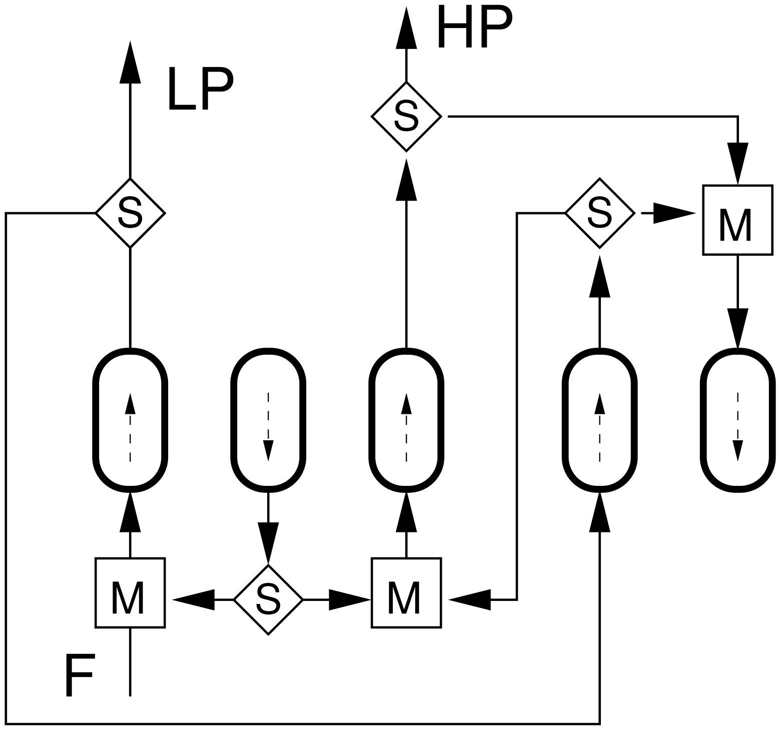
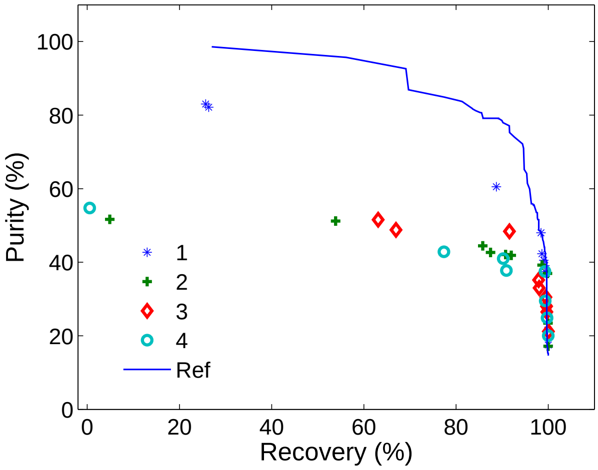
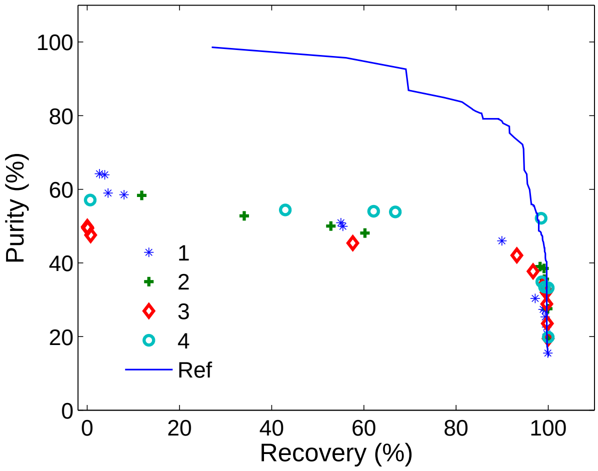


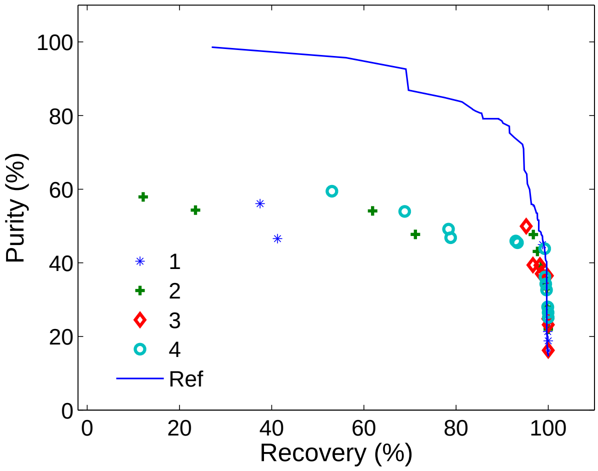
Figure 6.9: Configuration suggested by the simple evolutionary algorithm that dominates in the high purity region of the Pareto front in Fig. 6.8.



(a) Simple Evolutionary Algorithm



(b) Simulated Annealing



(c) Population Based Approach

Figure 6.10: Pareto fronts have been generated using 10 iteration of the external evolutionary procedure and a resized MOGA procedure. The overall Pareto front, *Ref*, obtained by the configurations in Table 6.3 is also shown for comparison.

6.5.2 Results obtained using a resized embedded MOGA

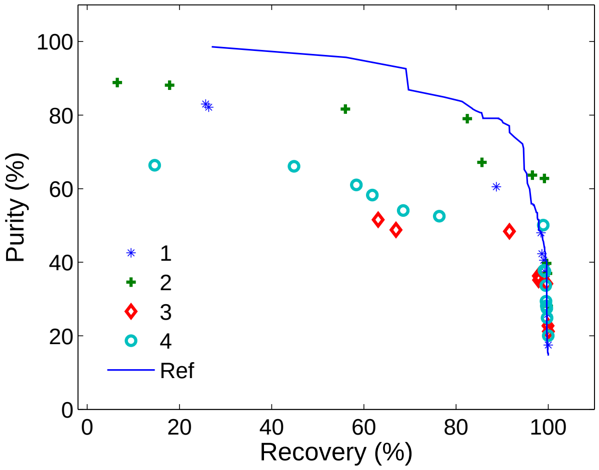
The design problem stated in eq. 6.2 has been solved using the three algorithms described previously. Each algorithm has been initialised 4 times, and a number of 20 iterations per run has been allowed: 20 cycle configurations per run have been considered. To reduce computational time, the embedded MOGA procedure has been resized so as to be less time requiring, by allowing only 5 generations (i.e. $ng = 5$), and a population size of 40 (i.e. n=40). No further restriction to the design space has been imposed in this case. Despite the reduction of the embedded MOGA, each run took approximately 8 days to complete.

The analysis carried out in this section aims at proving whether the three algorithms are able to evolve the overall Pareto front as the number of iterations progresses (comparison between Fig. 6.10 and 6.11), and whether the algorithms are able to propose cycle configurations whose performance dominate the overall Pareto front obtained by the 5 cycle configurations illustrated in Table 6.3. The comparison allows to understand how good the results proposed by the framework are. In all the figures with the results, this reference Pareto front is shown to help carrying out the comparison. However, while comparing the results, it is worth noting that the Pareto fronts obtained by the framework have been achieved using a resized MOGA procedure.

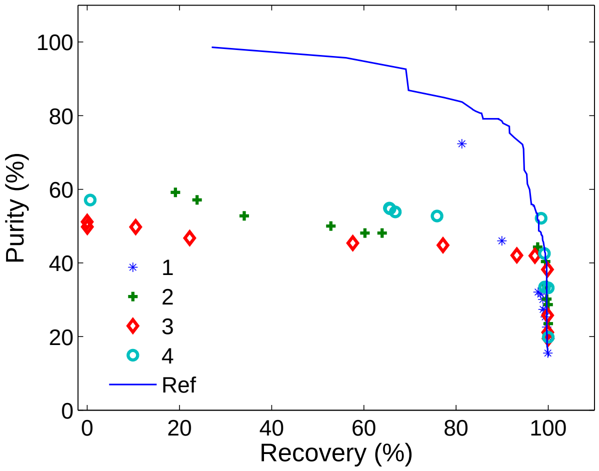
In Fig. 6.10, the Pareto fronts obtained by the three algorithms after the first 10 iteration are shown. The best results have been achieved by EA, whose first run obtained solutions with up to 80% purity and 20% recovery. Solutions obtained by SA and POP seems to reach a plateau in the value of the purity at around 60%. Some of the solutions show a recovery of less than 20%, and even close to 0%, which would make the configurations proposed by the framework unfeasible. As expected, the rules imposed to the configurations (see section 6.2.1) does not exclude unfeasible solutions from our search space.

In Fig. 6.11, it is shown what happen to the same runs showed in Fig. 6.10 after 10 more iterations. In all three cases, comparing the results at 10 and 20 iterations, a higher number of points in the high recovery region can be found, thus showing an overall improvement of the solutions found. In the case of EA, Fig. 6.11(a), runs 1 and 3 do not show a big improvement; on the other hand, the Pareto front of run 2 improves greatly, and some of the points in the high recovery region dominates the reference Pareto front. Improvement can be observed in runs 1 and 4 of the SA algorithm. For the POP algorithm, the main improvement has been achieved by run 2.

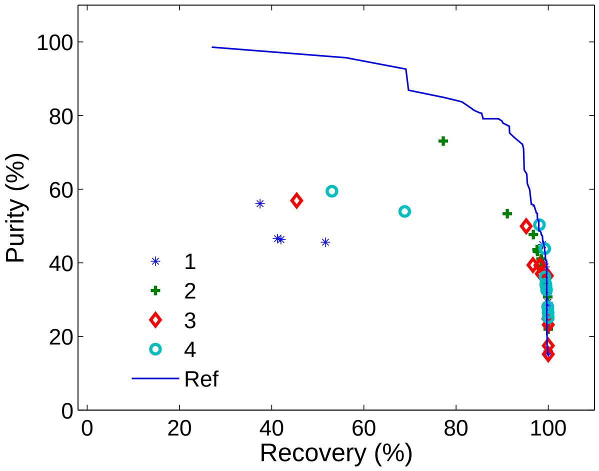
The next set of experiments, shown in Fig. 6.12, aims at investigating the performance of the algorithms in a reduced design space: as in section 6.5.1 it has been imposed to all



(a) Simple Evolutionary Algorithm



(b) Simulated Annealing



(c) Population Based Approach

FIGURE 6.11: Evolution of the Pareto fronts of Fig. 6.10 after ten more iterations and comparison with the reference Pareto front, *Ref*. In average, all the fronts have a higher number of solutions in the high recovery region.

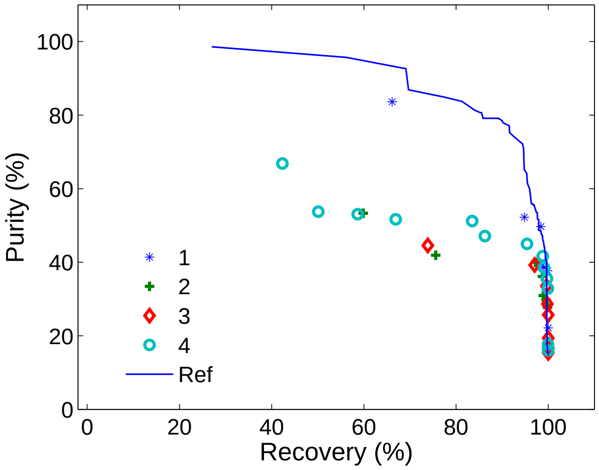
the configurations that c_{11}=-1 and $c_{12} = 2$. For this analysis, each algorithm has been allowed 10 iterations (10 configurations investigated), and the resized MOGA procedure has been implemented for the optimisation of the single configurations.

Within this reduced design space, the three algorithms obtained better solutions than in the previous set of experiment. No solution with a recovery lower than 20% has been found. An average improvement in the purity is observed in the solutions, especially in the region with recovery higher than 70%, meaning that a higher number of viable trade-off has been found. In particular, the Pareto of the second run of POP suggests two outstanding solutions, that dominate the reference Pareto front in the region of the elbow, the most appealing to the designer. As shown in Fig. 6.12(c), the best trade off corresponds to a recovery of 94.47% and a purity of 93.85%. The two solutions have been achieved by the same cycle configuration, shown in Fig. 6.14.

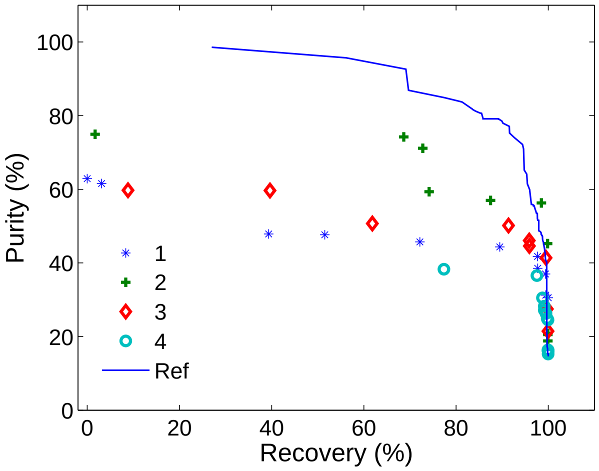
In Fig. 6.13, it is shown how different configurations contribute to the overall Pareto front obtained by POP in run 2 shown in Fig. 6.12(c). The configuration shown in Fig. 6.14 dominates the high purity area of the Pareto; many configurations, on the other hand, dominate the high recovery region. All the results presented in this section, suggest that high recovery solutions are easy to achieve, and the design challenge concerns the high purity region.

The complexity of the configuration in Fig. 6.14 demonstrates how the framework is able to synthesise configurations where unusual connections are allowed, and this would not have been possible within the framework proposed in previous studies [89, 116]. Despite in principle all the interconnections could be implemented, it is difficult to say at this stage whether the connections suggested in Fig. 6.14 can be implemented in practice. It would be interesting to interact with a manufacturer of PSA equipment to gain a deeper understanding of practical issues linked to the cycle configuration. The configuration in Fig. 6.14 requires connections more complicated than the one in Fig. 6.9, and it is therefore less likely to be practically viable.

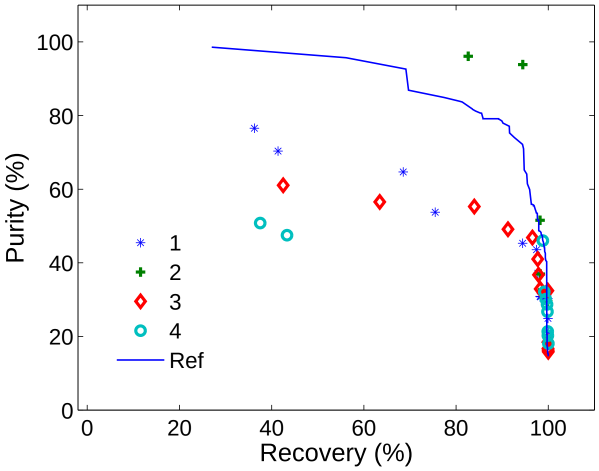
An interesting aspect of the configuration in Fig. 6.14 is that the output of step *II*, a blowdown, is sent both to the the inlet of step *I* and to the light product, thus “mixing” the corresponding characteristics of the configurations in Table 6.3. When step *II* is performed cocurrently, as in this case, the product withdrawn is not heavy, as the CO_2 adsorbed is stored in the feed end of the bed. Since the outlet stream of step *II* is sent to 5 units, only a low amount of the CO_2 it contains is lost in the light product, so it does not impact negatively on the recovery. The heavy product is withdrawn from a small fraction of step *II* and step *III*. Since the upper end of the bed has been purged in step *II*, a high purity product can be obtained.



(a) Simple Evolutionary Algorithm



(b) Simulated Annealing



(c) Population Based Approach

FIGURE 6.12: In a restricted design space, all the algorithms improved the quality of the solutions found. In its second run, POP obtained two outstanding solutions dominating the reference Pareto front, *Ref*, obtained by the configurations in Table 6.3.

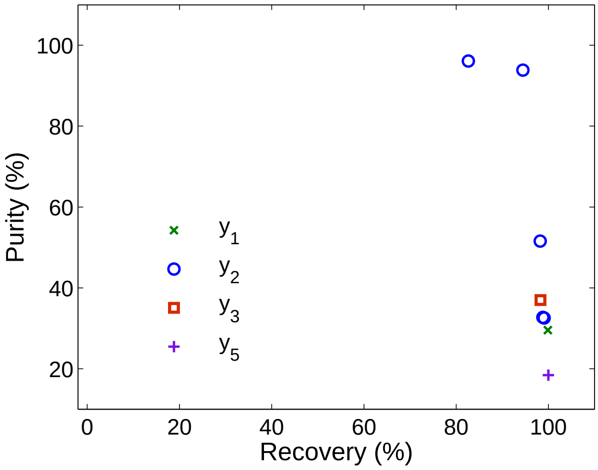


FIGURE 6.13: Different cycle configurations contribute to overall Pareto front obtained by POP in run 2 (Fig. 6.12(c)). For each point, the legend specifies by which configuration the point has been produced, e.g. the points with high purity have been provided by the second configuration, y_2.

The results discussed so far show the ability of the three algorithms to evolve, and potentially to find better solutions than the one proposed in literature. Given the low number of runs and the reduction of the embedded MOGA procedure, the average behaviour of the single algorithms should not be deduced from these results.

6.6 Conclusions

The performance of a PSA cycle, both in terms of quality of the products and cost of the operation, is linked to the cycle configuration used for the separation. Complex multi-bed/multi-step PSA cycles are often necessary, for instance when the desired product is the most preferably adsorbed species of the gas mixtures, as in the case of CO_2 capture from flue gases. The design of such complex cycles is a difficult task and the computational burden imposed by the simulation of PSA processes has hindered the development of appropriate automated tools for the synthesis of these cycles. Furthermore, the high computational nature of the problem makes the development of design tools for PSA cycle conceptually challenging.

A flowsheet design framework for the multi-criteria design of PSA cycle configurations has been presented in this chapter. The framework is based on an evolutionary procedure that, at each iteration, evaluates a cycle configuration, y. The performance of the configuration, represented by the Pareto front it generates via multi-objective optimisation, is fed back to the generator of cycles and taken into account when generating new

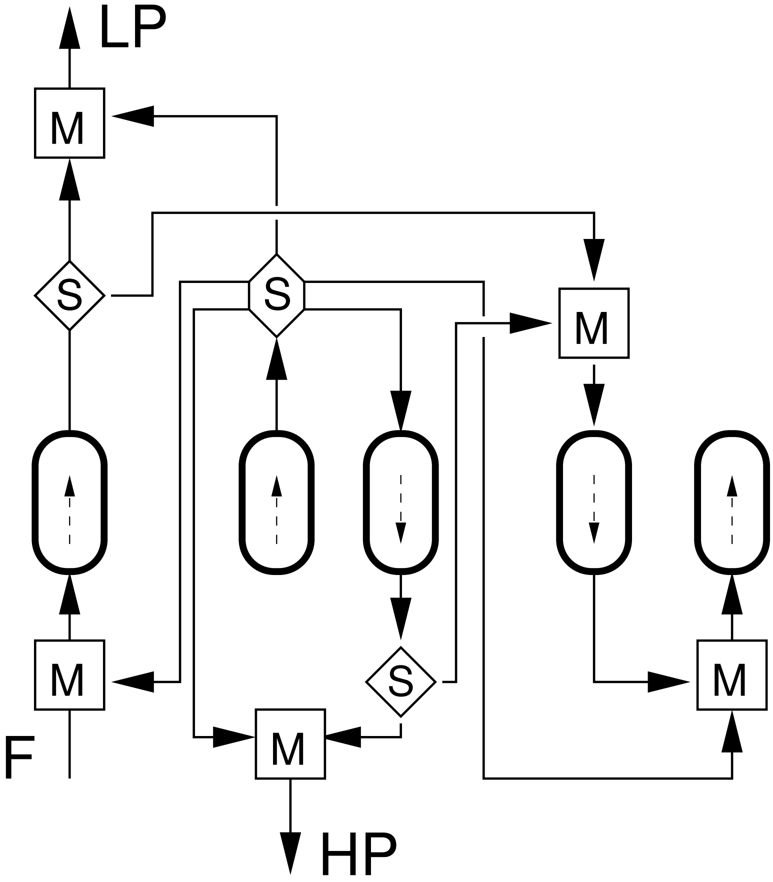


FIGURE 6.14: Configuration suggested by the population based evolutionary algorithm whose performance dominates the points of the reference case study.

configurations. The key element of the framework is the representation of the cycle configurations. In section 6.2 a representation of cycle configurations has been proposed, which allows for all the possible bed interconnections to be explored. The flexibility of the proposed representation allows to overcome the limitations of previous studies, which were based on the idea of selecting the sequence of steps in the cycle among a set of known steps (e.g. adsorption with feed, desorption etc.).

Three evolutionary algorithms have been proposed: a simple evolutionary algorithm (EA), a simulated annealing procedure (SA) and a population based algorithm (POP). The details of the three evolutionary procedures are provided in section 6.3. The evolutionary algorithms used in this chapter adopt a (1+1) evolution strategy: a single parent that generate single offspring [20]. The child is a neighbour of the parent. The procedure implemented to generate a neighbour is illustrated in section 6.4. The main difference among the three algorithms is in the way the parent configuration is selected at each iteration among the previously investigated configurations.

The performance of the framework has been investigated for a particular case study:

the synthesis of optimal cycle configurations for CO_2 capture from flue gases. Firstly, the ability of the cycle representation to represent known cycle configurations has been explored in section 6.2.3. A preliminary analysis of the results is proposed, rather than a full statistical investigation, due the high computational time required.

The results show the ability of the three algorithms to evolve the overall Pareto front and to propose cycle configurations that outperform cycle configurations proposed in literature by Reynolds *et al.* [105]. Furthermore, more than one configuration contribute to each of the overall Pareto fronts obtained. This proves that there is not a unique solution to the multi-objective problem investigated. In all the Pareto fronts, a high number of configurations contribute to the high recovery region of the Pareto, while only one or two appear to dominate the high purity region. This demonstrates that the high purity solutions are more challenging to obtain.

The simplifications introduced in chapter 2 allowed to perform in a manageable time the analysis carried out in this chapter. However, a full statistical analysis could not be carried out, and a reduced-size MOGA has been used, together with a low number of iterations per each run. Hence, the promising results obtained in this chapter can still only be considered as a preliminary analysis of the performance of the three algorithms. Overcoming the computational burden is a key element for the development of automated tools for the synthesis of PSA cycles. Suggestions on how to shorten the computational requirements of the framework are provided in the following chapter.

Chapter 7

Conclusions and directions for future work

7.1 Conclusions

The research project presented in this thesis focuses on the development of a multi-criteria design framework for complex Pressure Swing Adsorption (PSA) cycles. PSA is a cyclic separation process, whose main steps are adsorption, at high pressure, and regeneration of the adsorbent, at low pressure. Despite PSA is a well established industrial process for gas separation, the design of complex PSA cycles is still mainly an experimental effort. This is due to the challenges posed by the complexity of the simulation and by the need to detect the performance at cyclic steady state. However, automated tools for the design of PSA processes would be beneficial for the development of the technology due to the complex behaviour of the performance with respect to the design variables. The operation is characterised by trade-offs between different measures of the performance, but only few previous studies focus on the multi-objective optimisation of PSA cycles. The interest on PSA is due not only to its wide employment in industry, but also to the role it can play as an option for CO_2 capture from flue gases, as discussed in chapter 1.

A multi-criteria design framework for complex PSA cycles has been developed, able to provide useful information about trade-offs involved in the characterisation of PSA cycles. Case studies of interest have been separation of air for N_2 production and separation of CO_2 from flue gases.

The analysis of previous attempts to optimise PSA processes put in evidence some of the gaps in the existing literature regarding design tools for PSA processes:

- cumbersome simulations have been embedded within the optimisation framework, hence limiting the dimension of the PSA processes which could be optimised; the problem has often be avoided by the adoption of too simplified models, or the restriction to small case studies;
- a detailed description of the diffusion process has often been avoided, as the inclusion of diffusion equation would have implied an unmanageable simulation;
- no justification has been provided for the choice of the optimisation tool adopted;
- despite PSA performance is characterised by conflicting criteria, only two studies have addressed the optimisation of PSA cycles as a multi-objective design problem;
- few design strategies for the synthesis of PSA cycles have been proposed.

The design framework that has been proposed in this thesis aims at overcoming the above limitations. The key aspects of the design framework are:

- reduced computational costs imposed by the complexity of PSA models. Computational time has been reduced by introducing a simplified yet reliable model;
- both the detailed diffusion model and the linear driving force approximation have been taken into account to describe the mass transfer within the solid phase. This allows the optimisation of fast cycles, useful to reduce capital and operating costs;
- multi-criteria optimisation approach which allows to detect optimal trade-offs between conflicting parameters of the performance. A multi-objective genetic algorithm has been implemented to generate approximations of the Pareto front for the different design problems;
- capability of simulating and optimising complex multi-bed/multi-steps operations;
- Ability to syntesise novel PSA cycle configurations.

Some details on how the above points have been achieved are discussed in the remainder of this section.

7.1.1 Simulation and modelling

The first step of the project has been the development of appropriate simulation tools for PSA processes, presented in chapter 3. The computational requirements of the dynamic simulation of PSA cycles represents the bottle-neck for the development of automated design tools. This is due to the fact that PSA processes are described by a set of coupled PDAEs equations, and that the performance of the cycle needs to be evaluated at cyclic steady state (CSS), as discussed in chapter 2.

The simulations proposed in this thesis address the need of decreasing computational time without compromising accuracy. Each of the case studies poses a specific challenge to the development of the design framework: in the case of N_2 production, a detailed diffusion model needs to be adopted to capture the dynamics of this kinetically controlled separation. On the other hand, CO_2 capture is an equilibrium driven separation which requires the development of PSA cycles capable of providing a high purity of the strongly adsorbed species.

In both cases, the computational time has been cut by simulating each bed involved in a cycle as a series of CSTRs. A unibed approach has been adopted to speed up the conversion to CSS. An investigation has been carried out to detect the appropriate numerical strategies to solve the equations describing the two operations both accurately and time efficiently.

7.1.2 Characterisation of the objective function and single objective optimisation

In chapter 4 a preliminary investigation is carried out to characterise the objective function and determine the most appropriate class of optimiser to be used to design a PSA configurations. The example taken into account is the maximisation of the recovery obtained from a simple 2-bed/4-step Skarstrom for N_2 production from air.

The effect of the choice of the numerical method on the shape of the objective function has been investigated. Time-adaptive integration schemes have been introduced to reduce the time of the simulation: ode23 and ode45, both based on the Runge-Kutta method. It has been found that to a regular objective function profile, as for the concentration profiles, the recovery has to be calculated by the integrator itself, and it cannot be calculated from the concentration profiles by a quadrature scheme.

The investigation showed that the profile of the objective function is non-smooth. Direct Search methods are able to cope with non-smooth objective functions, and the performance of some of these method has been investigated. Among the optimisation algorithms used, the Nelder-Mead method, with a right-angled simplex, gave the more robust and efficient performance.

It was found that a hybrid optimisation technique, combining the Nelder-Mead and the multidirectional search methods, is the most successful and robust optimisation strategy.

DSMs proved to be reliable method for the single objective optimisation of the PSA process. However, as the recovery is increased, the purity of N_2 in the product stream decreases. Hence, it would be desirable to take into account the effect of the design variables on the conflicting criteria to properly design the operation.

7.1.3 Multi-objective optimisation

PSA operations are characterised by intrinsically conflicting criteria. In chapter 5 the multi-objective optimisation of two PSA processes has been addressed: the two case studies presented are the separation of air for N_2 production, and the CO_2 capture from flue gases.

Due to the non-smoothness of the objective function and the non-convexity of the feasible region, the ability of a multi-objective genetic algorithm (MOGA) to generate approximations to the Pareto front has been investigated. The key element of the algorithm is the definition of a targeted fitness function, deigned to encourage the evolutionary procedure to broaden the extent of the Pareto front. The aim is to help a design engineer identify the trade-offs between the different criteria and thereby choose the appropriate design point or even a region in the design space for further investigation. A statistical analysis showed that the MOGA proposed successfully compared with other optimisers, such as DSMs and NSGA-II. The effect of the MOGA parameters has also been investigated statistically.

7.1.4 A multi-criteria flowsheet design framework for PSA cycles

In chapter 6 a flowsheet multi-objective framework is introduced, able to synthesise optimal cycle configurations for CO_2 capture via PSA. The framework is based on an evolutionary procedure that, at every iteration, evaluates a cycle configuration, y. The performance of the configuration, represented by the Pareto front it generates via multi-objective optimisation, is fed back to the generator of cycles and taken into account when generating new configurations. The key element of the framework is the representation of the cycle configuration. In section 6.2 a representation of cycle configurations has been proposed, which allows for all the possible bed interconnections to be explored. The flexibility of the representation allows to overcome the limitations of previous studies, where cycle configuration could only be obtained by selecting the sequence of steps in the cycle among a set of known steps. The framework proposed constitutes a valuable tool to support the design and implementation of new cycles, addressing the needs of the many industries which are currently building new PSA plants. Three evolutionary algorithms have been developed for the task.

The preliminary results prove the ability of the three algorithms to propose cycle configurations that outperform cycle configurations proposed in literature by Reynolds *et al.* [105]. Despite the simplifications introduced in the models, the high computational requirements of the simulations prevented a full statistical analysis of the performance of the three algorithms. Overcoming the computational burden is a key element for the development of automated tools for the synthesis of PSA cycles. Suggestions on how to shorten the computational requirements of the framework are provided below.

7.2 Directions for future work

The results obtained showed the ability of the framework to synthesise novel PSA configurations and propose Pareto fronts whose points belong to different cycle configurations, thus showing the clear impact of the cycle configuration on the performance. However, the results presented appear to be partial under some aspects: despite the model for CO_2 capture has been validated for a 4-bed/4-step PSA cycle in section 5.5, it cannot accurately reproduce the data of the more complex configurations investigated in section 6.2.3. The high computational time required by the evaluation of the objectives limited the number of experiments that could be run.

The above observation motivates the adoption of detailed PSA models within the framework, together with more efficient techniques to reduce computational time. In particular, improvement to the framework should include:

- development of detailed models for PSA operations for a correct description of the kinetics and thermodynamic aspects of the process, which are vital for a correct evaluation of the process performance both in term of the quality of the product and cost of the operation;
- including the choice of the adsorbent material among the design variables available to the optimiser;
- exploring the implementation of surrogate models (i.e. response surface models, metamodels or emulators) that mimic the behaviour of the simulation model as closely as possible while being computationally cheaper to evaluate. This would reduce the number of times the optimiser needs to call the detailed simulators, thus reducing computational costs;
- parallel implementation of the framework, to further reduce computational costs;
- considering also other separation options (such as membrane, cryogenic, absorption) as an alternative or hybrid/complementary procedure to separate gases.

In particular, surrogate modelling and parallel computing would be precious techniques to enable the flowsheet design framework to properly explore the design space in a reasonable time span. The high computational time required by the evaluation of the objectives would become particularly stringent when a detailed column model is introduced in the framework. Surrogate models, or response surface models (RSM), mimic the behavior of a detailed simulation model, or the data obtained experimentally, as closely as possible while being computationally cheaper to evaluate. The detailed model is evaluated at a number of points, and the surrogate model is built by fitting the response of the detailed model with a function: by running the simulations at a set of points (experimental design) and fitting response surfaces to the resulting input-output data, fast surrogates models can be obtained for the objective and constraint functions used for optimisation [57].

Surrogate modelling has been recently used by Agarwal *et al.* [4], who considered “model reduction” as a valuable approach to reduce the computational efforts needed to simulate PSA processes within an optimisation framework. They developed reduced-order models (ROM) based on proper orthogonal decomposition (POD) as a low-dimensional approximation to a detailed PSA model. The ROM was used to describe the concentration profiles during a cycle. The ROM has been successfully used to optimise the recovery of H_2 from CH_4 in a simple PSA Skarstrom cycle. However, the ROM was able to mimic the behaviour of the PSA process only in the neighbourhood of the experimental conditions used to generate it. Hence, the ROM could not be used as a model to optimise the process in a wide design space. This first study shows that surrogate modelling is a viable option to simulate PSA cycle, however effort is needed to produce a surrogate model capable of representing the behaviour of a PSA in a sufficiently large design space. The higher the number of calls to the exact model, the more accurate the corresponding surrogate model. Hence, using parallel computing to evaluate the exact model would help to build a reliable surrogate model in a time-efficient way.

The development of detailed models, especially in the case of novel nanoporous materials, would require a close interaction between modelling and experiments. In developing our model for CO_2 capture on HTlcs in chapter 3, a limit to the reliability of the model depended on the lack of diffusivity data. It would be then necessary to carry out experiments to determine the equilibrium and kinetic data needed by the framework to implement a robust simulation of PSA processes. It would be interesting to explore the application of response-surface methodology (RSM) to experiment design. Statistical design of experiments reduces the number of experiments to be performed, considers interactions among the variables and can be used for optimization of the operating parameters in multivariable systems. RSM can be used to identify which input variables are the most important (highest contribution to variance of the output) so that the relationship between input-output can be detected and visualised. This might be an efficient approach towards the optimisation of the structure of nanoporous materials for the separation of CO_2 from the other components.

In the long term, other separation processes could be investigated, where adsorption and membrane separation are considered a valuable options. Interesting applications are related to both biogas upgrading and biogas reforming. In the first case, the biogas is purified via PSA up to natural gas grade, and then used as car fuel or gas substitute. In biogas reforming, the biogas is converted into very pure hydrogen which can then be used as a feed for fuel cells. PSA is involved in the final step of the purification process. The novelty of both the operations requires an effort to reduce the costs involved and improve the efficiency. Including these PSA applications within the framework developed would be beneficial to the development of biogas employment in the quest for greener fuels. In the same way, it would be possible to capitalise on the optimisation tools developed for the design of CO_2 capture processes to look at other industrial processes which involve computationally expensive simulations.

Bibliography

[1] Carbon dioxide capture and storage - Summary for policymakers and technical summary. IPCC special reports, Cambridge University Press, UK, 2005. Bert Metz, Ogunlade Davidson, Heleen de Coninck, Manuela Loos and Leo Meyer (Eds.).

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

[3] Agarwal, A., Biegler, L. T., and Zitney, S. E. Optimal synthesis of a pressure swing adsorption process for CO_2 capture. In *Proceedins of 08AIChE Annual Meeting* (2008), no. 505e. http://aiche.confex.com/aiche/2008/techprogram/P129362.HTM.

[4] Agarwal, A., Biegler, L. T., and Zitney, S. E. Simulation and optimization of pressure swing adsorption systems using reduced-order modeling. *Ind. Eng. Chem. Res. 48*, 5 (2009), 2327–2343.

[5] Ahari, J. S., Pakseresht, S., Mahdyarfar, M., Shokri, S., Zamani, Y., pour, A. N., and Naderi, F. Predictive dynamic model of air separation by pressure swing adsorption. *Chem. Eng. Technol. 29*, 1 (2006), 50–58.

[6] Ahn, H., and Brandani, S. Dynamics of carbon dioxie breakthrough in a carbon monolith over a wide concentration range. *Adsorption 11* (2005), 473–477.

[7] Alpay, E., and Scott, D. M. The linear driving force model for short-cycle adsorption and desorption in a spherical particle. *Chem. Eng. Science 47* (1992), 499–502.

[8] Alpay, E., and Scott, D. M. The linear driving force model for fast-cycle adsorption and desorption in a spherical particle. *Chem. Eng. Science 47*, 2 (1993), 499–502.

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

[10] Arvind, R., Farroq, S., and Ruthven, D. M. Analysis of a piston PSA process for air separation. *Chem. Eng. Science 57* (2002), 419–433.

[11] Banerjee, R., Narayankhedkar, K. G., and Sukhatme, S. P. Exergy analysis of pressure swing adsorption processes for air separation. *Chem. Eng. Science 45*, 2 (1990), 467–475.

[12] Berlin, N. Us patent: 3280536., January 1996.

[13] Biegler, L. T., Jiang, L., and Fox, V. G. Recent advances in simulation and optimal design of pressure swing adsorption systems. *Separation and Purification Reviews 33*, 1 (2004), 1–39.

[14] Brunauer, S., Deming, L. S., Deming, W. E., and Teller, E. On a theory of the van der waals adsorption of gases. *Journal of the American Chemical Society* (1940).

[15] Center, R. Capture and storage of CO_2. Tech. rep., IEA Geenhouse Gas, 2001.

[16] Chlendi, M., Tondeur, D., and Rolland, F. A method to obtain a compact representation of process performances from a numerical simulator: example of pressure swing adsorption for pure hydrogen production. *Gas separation & purification* (1995).

[17] Choi, W., Kwon, T., Y.Yeo, Lee, H., Song, H. K., and Na, B. Optimal operation of the pressure swing adsorption (PSA) process fo CO_2 recovery. *Korean J. Chem. Eng. 20*, 4 (2003), 617–623.

[18] Chou, C., and Chen, C. Carbon dioxide recovery by vacuum swing adsorption. *Separation and Purification Technology 39* (2004), 51–65.

[19] Chue, K. T., Kim, J. N., Yoo, Y. J., Cho, S. H., and Yang, R. T. Comparison of activated carbon and zeolite 13X for CO_2 recovery from flue gas by pressure swing adsorption. *Ind. Eng. Chem. Res. 34* (1995), 591–598.

[20] Coello, C. A. C., and Becerra, R. L. Evolutionary multiobjective optimization in materials science and engineering. *Materials and Manufacturing Processes 24*, 2 (2009), 119–129.

[21] Cruz, P., Alves, M. A., Magalhães, F. D., and A.Mendes. Solution of hyperbolic pdes using a stable adaptive multiresolution method. *Chem. Eng. Science 58*, 9 (2003), 1777–1792.

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

[23] Cruz, P., Magalhães, F. D., and Mendes, A. Generalized linear driving force approximation for adsorption of multicomponent mixtures. *Chem. Eng. Science 61* (2006), 3519–3531.

[24] Cruz, P., Santos, J. C., Magalhães, F. D., and Mendes, A. Cyclic adsorption separation processes: Analysis strategy and optimization procedure. *Chem. Eng. Science 58* (2003), 3143–3158.

[25] Cruz, P., Santos, J. C., Magalhães, F. D., and Mendes, A. Simulation of separation processes using finite volume method. *Comp. & Chem. Eng. 30*, 1 (2005), 83–98.

[26] Deb, K. *Multi-objective optimization using evolutionary algorithms.* J. Wiley & Son Inc., Chester,UK, 2001.

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

[28] Diagne, D., Goto, M., and Hirose, T. New PSA process with intermediate feed inlet position operated with dual reflux: Application to carbon dioxide removal and enrichment. *Journal of Chemical Engineering of Japan 27*, 1 (1994), 85–89.

[29] Diagne, D., Goto, M., and Hirose, T. Experimental study of simultaneous removal and concentration of CO_2 by an improved pressure swing adsorption process. *Energy Convers. Mgmt 36*, 6-9 (1995), 431–434.

[30] Diagne, D., Goto, M., and Hirose, T. Parametric studies on CO_2 separation and recovery by a dual reflux process consisting of both rectifying and stripping sections. *Ind Eng. Chem. Res 34* (1995), 3083–3089.

[31] Diagne, D., Goto, M., and Hirose, T. Numerical analysis of a dual refluxed PSA process during simultaneous removal and concentration of carbon dioxide dilute gas from air. *J. Chem. Tech. Biotechnol. 65* (1996), 19–38.

[32] Ding, Y., and Alpay, E. Equilibria and kinetics of CO_2 adsorption on hydrotalcite adsorbent. *Chem. Eng. Science 55* (2000), 3461–3474.

[33] Ding, Y., and Alpay, E. High temprerature recovery of CO_2 from flue gases using hydrotalcite adsorbent. *Trans IChemE 79*, Part B (2001), 45–51.

[34] Ding, Y., Croft, D. T., and LeVan, M. D. Periodic states of adsorption cycles IV. direct optimization. *Chem. Eng. Science 57* (2002), 4521.

[35] Ebner, A. D., Reynolds, S. P., and Ritter, J. A. Understanding the adsorption and desorption behaviour of CO_2 on a K-promoted hydrotalcite-like compound (HTlc) through nonequilibrium dynamimc isotherms. *Ind. Eng. Chem. Res. 45* (2006), 6387–6392.

[36] Ebner, A. D., Reynolds, S. P., and Ritter, J. A. Nonequilibrium kinetic model that describes behaviour of CO_2 in a K-promoted hydrotalcite-like compound. *Ind. Eng. Chem. Res. 46* (2007), 1737–1744.

[37] Ebner, A. D., and Ritter, J. A. Equilibrium theory analysis of rectifying PSA for heavy component production. *AIChE Journal 48*, 8 (2002).

[38] Ebner, A. D., and Ritter, J. A. Equilibrium theory analysis of dual reflux PSA for separation of a binary mixture. *AIChE Journal 50*, 10 (2004).

[39] Finlayson, B. A. *Nonlinear Analysis in Chemical Engineering.* Ravenna Park Publishing, Inc., Seattle, Washington USA, 2003.

[40] Fletcher, A. J. Porosity and sorption behaviour. `http://www.staff.ncl.ac.uk/a.j.fletcher/adsorption.htm`, 2008.

[41] Glueckauf, E., and Coates, J. I. Theory of chromatography. Part IV. The influence of incomplete equilibrium on the front boundary of chromatograms and on the effectiveness of separation. *Journal of Chemical Society* (1947), 1315–1321.

[42] Guan, J., aand C. Long, R. Z., and Hu, X. Dynamic simulation of pressure swing adsorption system with the electrical network. *Chem. Eng. Science 60*, 16 (2005), 4635–4645.

[43] Hamda, H., Roudenko, O., and Schoenauer, M. Multi-objective evolutionary topological optimum design. In *Adaptive Computing in Design and Manufacture V* (2002), pp. 121–132.

[44] Hassan, M. M., Raghavan, N. S., and Ruthven, D. M. Pressure swing air separation on a carbon molecular-sieve. 2. Investigation of a modified cycle with pressure equalization and no purge. *Chem. Eng. Science 42*, 8 (1987), 2037–2043.

[45] Hassan, N. M., Ruthven, D. M., and Raghavan, N. S. Air separation by pressure swing adsorption on a carbon molecular sieve. *Chem. Eng. Science 41*, 5 (1986), 1333–1343.

[46] Herrera, F., Lozano, M., and Sánchez, A. M. A taxonomy for the crossover operator for real–coded genetic algorithms: an experimental study. *International Journal of Intelligent Systems 18*, 3 (2003), 309–338.

[47] Herrero, J. M., Martinez, M., Sanchis, J., and Blasco, X. *Computational and Ambient Intelligence*, vol. 4507 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2007, ch. Well distributed Pareto Front by Using the $\not\epsilon$-MOGA Evolutionary Algorithm, pp. 292–299.

[48] Higham, N. J. Optimization by direct search in matrix computation. *SIAM J. Matrix Anal. Appl. 14*, 2 (1993), 317–333.

[49] Ho, M. T., Allinson, G. W., and Wiley, D. E. Reducing the cost of CO_2 capture from flue gases using pressure swing adsorption. *Ind. Eng. Chem. Res. 47* (2008), 4883–4890.

[50] Horn, J., Nafpliotis, N., and Goldberg, D. E. A niched pareto genetic algorithm for multiobjective optimization. In *Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence* (1994), pp. 82–87.

[51] Intyre, J. A., Holland, C. E., and Ritter, J. A. High enrichment and recovery of dilute hydrocarbons by dual-reflux pressure swing adsorption. *Ind. Eng. Chem. Res. 41* (2002), 3499–3504.

[52] Ishibashi, M., Ota, H., Umeda, S., Tajika, M., Izumi, J., Yasutake, A., Kabata, T., and Kageyama, Y. Technology for removing carbon dioxide from power plant flue gas by the physical adsorption method. *Energy convers. Mgmt 1996*, 6-8 (1996), 929–933.

[53] Jain, S., Moharir, A., Li, P., and Wozni, G. Heuristic design of pressure swing adsorption: A preliminary study. *Separation and Purification Technoloogy 33*, 1 (2003), 25–43.

[54] Jiang, L., Biegler, L. T., and Fox, V. G. Simulation and optimization of pressure swing adsorption system for air separation. *AIChE Journal 49*, 5 (2003), 1140.

[55] Jiang, L., Biegler, L. T., and Fox, V. G. Design and optimization of pressure swing adsorption systems with parallel implementation. *Computers and Chemical Engineering 29* (2005), 393–399. Short Communication.

[56] Jiang, L., Fox, V. G., and Biegler, L. T. Simulation and optimal design of multiple bed pressure swing adsorption systems. *AIChE Joutnal 50*, 11 (2004), 2904–2917.

[57] Jones, D. R. A taxonomy of global optimization methods based on response surfaces. *Journal of Global Optimization 21*, 4 (2001), 345–383.

[58] Jones, R. L., Keller, G. E., and Wells, R. C. Us patent 4194892, 1980.

[59] Kearns, D. T., and Webley, P. A. Application of an adsorption non-flow exergy function to an exergy analysis of a pressure swing adsorption cycle. *Chem. Eng. Science 59* (2004), 3537–3557.

[60] Kelley, C. T. *Iterative Methods for Optimization.* Frontiers in Applied Mathematics. SIAM, 1999.

[61] Kelley, C. T. A brief introduction to implicit filtering. `http://www.ncsu.edu/crsc/reports/ftp/crsc-tr02-28.ps.gz.`, September 2002.

[62] Kikkinides, E. S., Yang, R. T., and Cho, S. H. Concentration and recovery of CO_2 from flue gas by pressure swing adsorption. *Ind. Eng. Chem. Res. 32* (1993), 2714–2720.

[63] Ko, D., and Moon, I. Multiobjective optimization of cyclic adsorption processes. *Ind. Eng. Chem. Res. 42*, 2 (2002), 93–104.

[64] Ko, D., Siriwardane, R., and Biegler, L. T. Optimization of a pressure-swing adsorption process using zeolite 13X for CO_2 sequestration. *Ind. Eng. Chem. Res. 42*, 2 (2003), 339–348.

[65] Ko, D., Siriwardane, R., and Biegler, L. T. Optimization of pressure swing adsorption and fractionated vacuum pressure swing adsorption processes for CO_2 capture. *Ind. Eng. Chem. Res. 44*, 21 (2005), 8084–8094.

[66] Kopaygorodsky, E., Guliants, V. V., and Krantz, W. B. Predictive dynamic model of single-stage ultra-rapid pressure swing adsorption. *AIChE Journal 50*, 5 (2004), 953–962.

[67] Kostrosky, K. P., and Wankat, P. C. High recovery cycles for gas separations by pressurre swing adsorption. *Ind. Eng. Chem. Res. 45* (2006), 8117–8133.

[68] Krishna, R., and Wesselingh, J. A. The Maxwell-Stefan approach to mass transfer. *Chem. Eng. Science 52*, 6 (1997), 861–911.

[69] Kumar, R., Fox, V. G., Hartzog, D. G., Larson, R. E., Chen, Y. C., Houghton, P. A., and Naheiri, T. A versatile process simulator for adsorptive separations. *Chem. Eng. Science 49*, 18 (1994), 3115–3125.

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

[71] Lagarias, J. C., Reeda, J. A., Wright, M. H., and Wright, P. E. Convergence properties of the nelder-mead simplex method in low dimensions. *SIAM Journal on Optimization 9*, 1 (1998), 112–147.

[72] Laumanns, M., Thiele, L., Thiele, K., Deb, K., and Zitzler, E. Combining convergence and diversity in evolutionary multiobjective optimization. *Evolutionary Computation 10*, 3 (2002), 263–282.

[73] Laumanns, M., Zitzler, E., and Thiele, L. *Evolutionary Multi-Criterion Optimization*, vol. 1993 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2001, ch. On the Effects of Archiving, Elitism, and Density Based Selection in Evolutionary Multi-objective Optimization, pp. 181–196.

[74] Levenspiel, O. *Chemical Reaction Engineering*. J. Wiley & Son Inc., 1962.

[75] Lewandowski, J., Lemcoff, N. O., and Palosaari, S. Use of neural networks in the simulation and optimization of pressure swing adsorption processes. *Chemical Engineering & Technology 21*, 7 (1998), 593–597.

[76] Lewis, R. M., Torczon, V., and Trosset, M. W. Direct search methods: Then and now. *Journal of Computational and Applied Mathematics 124*, 1-2 (2000), 191–207.

[77] Lundy, M., and Mees, A. Convergence of an annealing algorithm. *Mathematical Programming 34* (1986), 111–124.

[78] Malek, A., and Farooq, S. Study of a six-bed pressure swing adsorption process. *AIChE Journal 43*, 10 (1997), 2509–2523.

[79] Marsh, W. D. Us patent: 3142547, July 1964.

[80] McKinnon, K. I. M. Convergence of the nelder-mead simplex method to a nonstationary point. *SIAM Journal on Optimization 9*, 1 (1998), 148–158.

[81] Mehrotra, A., Ebner, A. D., and Ritter, J. A. Pressure swing adsorption cycles for carbon dioxide capture. In *Proceedins of 08AIChE Annual Meeting* (2008), no. 340f. `http://aiche.confex.com/aiche/2008/techprogram/P134719.HTM`.

[82] Meza, J. C., Judson, R. S., Faulkner, T. R., and Treasurywala, A. M. A comparison of a direct search method and a genetic algorithm for conformational searching. *Journal of Computational Chemistry 17*, 9 (1996), 1142–1151.

[83] Mhlenbein, H., and Schlierkamp-Voosen, D. Predictive models for the breeder genetic algorithm I. Continuous parameter optimization. *Evolutionary Computation 1*, 1 (1993), 25–49.

[84] Miller, B. L., and Goldberg, D. E. Genetic algorithms, tournament selection, and the effects of noise. http://citeseer.ist.psu.edu/cache/papers/cs/4086/http:zSzzSzgal4.ge.uiuc.eduzSzpubzSzpaperszSzIlliGALszSz95006.pdf/miller95genetic.pdf, July 1995.

[85] Murata, T., and Ishibuchi, H. Moga: multi-objective genetic algorithms. In *International Conference on Evolutionary Computation, IEEE* (1995), pp. 289–924.

[86] Naderi, B., Zandieh, M., and Roshanaei, V. Scheduling hybrid flowshops with sequence dependent setup times to minimize makespan and maximum tardiness. *Int J Adv Manuf Technol 41* (2009), 1186–1198.

[87] Nakao, S., and Suzuki, M. Mass transfer coefficient in cyclic adsorption and desorption. *Chem. Eng. Science 57* (2002), 4227–4242.

[88] Nelder, J. A., and Mead, R. A simplex-methos for function minimization. *Computer Journal 7*, 4 (1965), 308–313.

[89] Nikolic, D., Kikkinides, E. S., and Georgiadis, M. C. Optimization of multibed pressure swing adsorption processes. *Ind. Eng. Chem. Res. 48*, 11 (2009), 5388–5398.

[90] Nilchan, S., and Panthelides, C. C. On the optimization of periodic adsorption processes. *Adsorption 4*, 2 (1998), 113–147.

[91] Octave-Forge. Extra packages for GNU Octave. http://octave.sourceforge.net/, August 2009.

[92] Park, J., Beum, H., Kim, J. N., and Cho, S. H. Numerical analysis on the power consumption of the PSA process for recovering CO_2 from flue gas. *Ind. Eng. Chem. Res.* (2002).

[93] Parmee, I. C. *Evolutionary and adaptive computing in engineering design.* Springer-Verlag, 2001.

[94] Perry, R. H., Green, D. W., and Maloney, J. O. *Perry's Chemical Engineers' Handbook.* McGraw-Hill, 1997.

[95] Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. *Numerical Recepies in FORTRAN.* Press Sindacate of the University of Cambridge, 1986.

[96] Pugsley, T. S., Berruti, F., and Chakma, A. Computer-simulation of a novel circulating fluidized-bed pressure-temperature swing adsorber for recovering carbon-dioxide from flue-gases. *Chem. Eng. Science 49*, 24A (1994), 4465–4481.

[97] Raghavan, N. S., Hassan, M. M., and Ruthven, D. M. Numerical simulation of a PSA system using a pore diffusion model. *Chem. Eng. Science 41*, 11 (1986), 2787–2793.

[98] Raghavan, N. S., and Ruthven, D. M. Numerical simulation of a fixed bed adsorption column by the method of orthogonal collocation. *AIChE Journal 9*, 6 (1983), 922–925.

[99] Rajasree, R., and Moharir, A. S. Simulation based synthesis, design and optimization of pressure swing adsorption processes. *Computers & Chemical Engineering 24*, 11 (2000), 2493–2505.

[100] Rasmussen, C. E., and Williams, C. K. I. *Gaussian Processes for Machine Learning.* The MIT Press, 2006. Online at http://www.gaussianprocess.org/ with links to code as well.

[101] Reynolds, S. P., Ebner, A. D., and Ritter, J. A. New pressure swing adsorption cycles for carbon dioxide sequestration. *Adsorption 11* (2005), 531–536.

[102] Reynolds, S. P., Ebner, A. D., and Ritter, J. A. Carbon dioxide capture from flue gas by pressure swing adsorption at high temperature using a K-promoted HTlc: Effects of mass transfer on the process performance. *Environmental Progress 25*, 4 (2006), 334–342.

[103] Reynolds, S. P., Ebner, A. D., and Ritter, J. A. Enriching PSA cycle for the production of nitrogen from air. *Industrial & Engineering Chemistry Research 45*, 9 (2006), 3256–3264.

[104] Reynolds, S. P., Ebner, A. D., and Ritter, J. A. Stripping PSA cycles for CO_2 recovery from flue gas at high temperature using a hydrotalcite-like adsorbent. *Ind. Eng. Chem. Res. 45* (2006), 4278–4294.

[105] Reynolds, S. P., Mehrotra, A., Ebner, A. D., and Ritter, J. A. Heavy reflux PSA cycles for CO_2 recovery from flue gas: Part I. Performance evaluation. *Adsorption 14* (2008), 399–413.

[106] Rouse, A., and Brandani, S. A new LDF approximation for cyclic adsorption processes. *AIChE Journal, Separation Technology Topical Conference 2* (2001).

[107] Ruthven, D. M. *Principles of adsorption and adsorption processes.* John Wiley & Sons, New York, 1984.

[108] Ruthven, D. M., Farooq, S., and Knaebel, K. S. *Pressure Swing Adsorption.* VCH, 1993.

[109] Sankararao, B., and Gupta, S. K. Multi-objective optimization of pressure swing adsorbers for air separation. *Ind. Eng. Chem. Res. 46*, 11 (2007), 3751–3765.

[110] Schaffer, J. D. Multiple objective optimization with vector evaluated genetic algorithms. In *Proceedings of the 1st International Conference on Genetic Algorithms* (1985), Lawrence Erlbaum Associates, Inc., pp. 93–100.

[111] Seshadri, A. NSGA-II: A multi-objective optimization algorithm. `http://www.mathworks.com/matlabcentral/fileexchange/10429`, August 2009.

[112] Silverman, B. W. *Density estimation for statistics and data analysis.* London: Chapman and Hall, 1986.

[113] Singer, S., and Singer, S. Complexity analysis of nelder-mead search iterations. *Applied Mathematics and Computation* (1999), 185–196.

[114] Singer, S., and Singer, S. Efficient implementation of the nelder-mead search algorithm. *Applied Numerical Analysis & Computational Mathematics 1*, 3 (2004), 524–534.

[115] Skarstrom, C. W. Method and apparatus for fractionating gaseous mixtures by adsorption, 1960. http://www.freepatentsonline.com/2944627.htm.

[116] Smith, O. J., and Westerberg, A. W. Mixed-integer programming for pressure swing adsorption cycle scheduling. *Chem. Eng. Science 45*, 9 (1990), 2833–2842.

[117] Smith, O. J., and Westerberg, A. W. The optimal design of pressure swing adsorption systems. *Chem. Eng. Science 46*, 12 (1991), 2967–2976.

[118] Subramanian, D., and Ritter, J. A. Equilibrium theory for solvent vapour recovery by pressure swing adsorption: analytical solution for process performance. *Chem. Eng. Science 52*, 18 (1997), 3147–3160.

[119] Suman, B., and Kuman, P. A survey of simulated annealing as tool for single and multiobjectiva optimization. *Journal of the Operational Research Society 57* (2006), 1143–1160.

[120] Sumathi, S., Hamsapriya, T., and Surekha, P. *Evolutionary intelligence: an introduction to theory and applications with MATLAB.* Springer-Verlag, 2008.

[121] Sundaram, N. Training neural networks for pressure swing adsorption processes. *Ind. Eng. Chem. Res. 38* (1999), 4449–4457.

[122] Suzuki, M., Suzuki, T., Sakoda, A., and Izumi, J. Recovery of carbon dioxide from stack gas by piston-driven ultra-rapid PSA. *Journal of Chemical Engineering of Japan 30*, 6 (1997), 1026–1033.

[123] Talbi, E. G., Mostaghim, S., Okabe, T., Ishibuchi, H., Rudolph, G., and Coello, A. C. C. Parallel approaches for multiobjective optimization. In *Lecture noted in Computer Science* (2008), J. Branke, K. Deb, K. Miettinen, and R. Slowinski, Eds., vol. 5252, pp. 349–372.

[124] Teague, K. G., and Edgar, T. F. Predictive dynamic model of a small pressure swing adsorption air separation unit. *Ind. Eng. Chem. Res. 38* (1999), 3761–3775.

[125] The MathWorks. MATLAB Getting Started Guide. www.mathworks.com/access/helpdesk/help/techdoc/, August 2009.

[126] Todd, R., Buzzi-Ferraris, G., Manca, D., and Webley, P. A. Improved ODE integrator and mass transfer approach for simulating a cyclic adsorption process. *Comp. & Chem. Eng. 27*, 6 (2003), 883–899.

[127] Todd, R. S., and Webley, P. A. Mass-transfer models for rapid pressure swing adsorption simulation. *AIChE Journal 52*, 9 (2006), 3126–3145.

[128] Torczon, V. On the convergence of the multidirectional search algorithm. *SIAM J. Optimization 1*, 1 (1991), 123–145.

[129] Wagner, J. L. Us patent: 34304181, March 1969. http://www.freepatentsonline.com/34304181.html.

[130] Webley, P. A., and He, J. Fast solution-adaptive finite volume method for PSA/VSA cycle simulation; 1 single step simulation. *Comp. & Chem. Eng. 23* (2000), 1701–1712.

[131] Woods, D. J. *An Interactive Approach for Solving Multi-Objective Optimization Problems.* PhD thesis, Rice University, Houston,Texas, USA, 1985.

[132] Yong, Z., Mata, V., and Rodrigues, A. E. Adsorption of carbon dioxide onto hydrotelcite-like compunds (htlcs) at high temperatures. *Ind. Eng. Chem. Res. 40* (2001), 204–209.

[133] Yong, Z., and Rodrigues, A. E. Hydrotalcite-like compounds as adsorbents for carbon dioxide. *Energy Conversion and Management 43* (2002), 1865–1876.

[134] Yu, W., Hidajat, K., and Ray, A. K. Application of multiobjective optimization in the design and operation of reactive SMB and its experimental verification. *Ind. Eng. Chem. Res. 42*, 26 (2003), 6823–6831.

[135] Zhang, J., and Webley, P. A. Cycle development and design for CO_2 capture from flue gas by vacuum swing adsorption. *Environ. Sci. Technol. 42* (2008), 563–569.

[136] Zhang, J., Webley, P. A., and Xiao, P. Effect of process parameters on power requirements of vacuum swing adsorption technology for CO_2 capture from flue gases. *Energy Conversion and Management 49* (2008), 345–356.

[137] Zhang, Z., Mazzotti, M., and Morbidelli, M. Multiobjective optimization of simulated moving bed and varicol processes using genetic algorithm. *Journal of Chromatography A 989*, 1 (2003), 95–108.

[138] Zitzler, E., Deb, K., and Thiele, L. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation 8*, 2 (2000), 173–195.

[139] Zitzler, E., Laumanns, M., and Bleuler, S. A tutorial on evolutionary multiobjective optimization. In *Metaheuristics for Multiobjective Optimisation* (2004), Springer-Verlag, pp. 3–38.

[140] Zitzler, E., and Thiele, L. Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *Evolutionary Computation, IEEE Transactions 3*, 4 (1999), 257–271.

[141] Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C. M., and da Fonseca, V. G. Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transaction on Evolutionary Computation 7*, 2 (2003), 117–131.