Optimization Boosts Decarbonization: Accelerating Net Zero from the Perspective of Carbon Capture and Utilization

Zhimian Hao

Supervisor: Prof. Alexei. A. Lapkin

Department of Chemical Engineering and Biotechnology

University of Cambridge

This dissertation is submitted for the degree of

Doctor of Philosophy

Selwyn College Dec 2021

I would like to dedicate this thesis to my loving parents.

Declaration

This thesis is the result of my own work and includes nothing which is the outcome of work done in collaboration except as declared in the Preface and specified in the text. I further state that no substantial part of my thesis has already been submitted, or, is being concurrently submitted for any such degree, diploma or other qualification at the University of Cambridge or any other University or similar institution except as declared in the Preface and specified in the text. It does not exceed the prescribed word limit for the relevant Degree Committee.

Zhimian Hao

Dec 2021

Acknowledgements

Firstly, I would like to thank Prof. Alexei Lapkin for his supervision and support throughout my PhD study. Thanks to his high-level vision, I have been on the trend of digitalization and net zero for four years. More importantly, he enlightens me on the sense of research: how to formulate research questions. He shows me the magic of interdisciplinary studies by applying machine learning to solving systems engineering problems. Further, his funds in multiple conferences, a research exchange, and my final-year PhD make it possible to expand my horizons over the past four years.

I appreciate the insightful comments and suggestions from my examiners – Dr. Ewa Marek and Dr. Dongda Zhang to help me in improving the quality of the thesis.

Thanks to my excellent colleagues - Chonghuan Zhang, who taught me the programming of neural networks in the initial stage of my research; Dr. Magda Barecka for the suggestions on the detailed analysis for CCU; Dr. Liwei Cao and Kobi Felton for the discussion on machine learning basics; Dr. Jana Weber and Dr. Nicholas Jose for the feedback on my presentation skills and proofreading; It is a fabulous experience to work with lovely colleagues – Perman, Sabine, Ahmad, Danilo, Adarsh, and Peace *et al.* I owe my gratitude to Prof. Alexander Mitsos, who accepted me for an exchange in his group at RWTH Aachen University. Thanks to Dr. Kosan Roh, who offered me the initial guidance of LCA; Also, my respects to many brilliant 'mathematicians' there: Dr. Artur Schweidtmann for patiently explaining optimization basics to me and offering me the hybrid optimization concept, together with Dr. Adel Mhamdi, Dr. Adrian Caspari and Dr. Yannic Vaupel for their assistance in setting up the dynamic optimization framework, which contributed to my first publication.

On my life side, thanks to my tutors – Dr. Anita Faul, Dr. Stewart Cant and Dr. Jörg Haustein for the continuous encouragement. I appreciate the informal chat with college staff, particularly Mrs. Carolyn Taylor (tunnel before the light; in the future, recall the pandemic experience; think about life in a broader time-space scale) and Mrs. Samantha Carr (calm down and things come). I thank my friends for the emotional support! With rowing/tennis friends, I have kept myself physically fit.

I acknowledge the financial support from China Scholarship Council and Cambridge Trust.

Final thanks to my loving parents for their support throughout the ups and downs of my life.

Journey to a PhD degree

How time flies! It is near the end of this PhD! For me, this PhD is really like a Journey, through which I learn, grow and get to know excellent people.

Before the journey, my understanding on optimization was limited to 'maximizing or minimizing a target', while Prof. Johan Grievink (TU Delft) told me that 'optimization is decision-making'. Several colleagues at Imperial College reminded me of the two sub-divisions: modelling and algorithms, but I did not fully understand their difference by that time. After the prior training in numerical mathematics, kinetics and reactor modelling before 2017, I was confident in my mathematic capacity. Then I applied to a PhD position in Prof. Alexei Lapkin's group, mentioning that 'I am keen on process optimization…publication is not the destination…whether continuing academia or industry later, I will keep eager to deliver the acquired knowledge to industrial sectors with an objective to create a sustainable future'. Luckily, I got the position.

Once I started the PhD journey, I tried several side projects on process synthesis and design of experiments. I came to realize that optimization can be used to identify the best configurations and the proper experimental conditions. Yes, optimization is decision-making! RWTH mathematicians emphasized the robustness of calculating rather than searching for the optimal solution. They also showed me the two different types of algorithms, which can be applied to solve the same problem. Additionally, Cambridge colleagues showed me the magic of machine learning to reduce the complexity of process models. Merging all of them leads to the foundation of the methodology in this thesis.

When it comes to the central part of this journey – problem solving, Alexei offered me a flexible system structure of CCU, starting from H_2 and power plant flue gas. However, my limited knowledge assumed that H_2 is produced only from natural gas. Such an assumption confined this system to a chemicals sector, which was probably against the initial perspective of Alexei (because our group was working on electrochemical processes for H_2 production in the meanwhile), but he still encouraged me to move on. Until recently, Alexei told me, 'Zhimian, you are an over-thinking style, which led to slow progress initially'. Once I started with more focus on 'chemicals', the progress has been smooth since then, and an extra focus was given to electrification on the energy demand side. However, the 'chemicals' thinking can only lead to a sub-optimal solution due to the reduced decision space. Consequently, a superior solution

involved with renewables was missed (If I can turn back the clock, I would regard it as an energy system, which can be supplied by diverse energy sources ranging from fossil fuels to renewables.). On the other hand, the ‘chemicals’ thinking brings in benefits: ‘no renewables’ assumption delivers a more robust evaluation on CCU, and hopefully the ‘negative result’ in my work can help raise the awareness and more discussions in the CCUS field.

To the near end of this journey, I worked through the CCU project and then asked Alexei ‘which journal should we consider? The writing style should be consistent with that journal style’. Alexei replied ‘focus on your own logic and write it up first, and then think about journals. Of course, a good work deserves a good publication but do it in a proper sequence’. Before the PhD, I did not care too much about publication, but somehow I have been gradually interested in novelty and publication in the last two years of my PhD. So far, I have no clear answer why and how it happened, but hopefully time will tell me. Whether in academia or industry, question formulation / methodology development / problem solving are transferable skills I learned here, and working with these excellent people boosted the learning process.

In case that someone asks me about similar doubts before his/her journey, I will explain them in plain language as follows,

Modelling: **express** a system or a problem, *e.g.*, reactor/process modelling or optimization modelling (some books do call it this way, while I suppose that optimization formulation sounds more straightforward).

Optimization: make **decisions** to achieve improvement.

Algorithm: a **strategy** for problem-solving, which is normally implemented in an automatic manner, *e.g.*, an optimization algorithm refers to a strategy for decision-making.

Cheers,

Zhimian Hao

20 Dec 2021

Abstract

Achieving a carbon-neutral, or "net zero" society requires a transition from fossil fuels to low-carbon solutions. However, fossil fuels supply approx. 80% of today's worldwide energy demand and are projected to play an indispensable role in the immediate future [1]. Carbon capture may be the most effective way to decarbonize the fossil fuel-based energy sector. Carbon capture consists of two major fields: carbon capture and storage (CCS) as well as carbon capture and utilization (CCU). While CCS is more relevant to electricity production, CCU is compatible with the existing downstream processes of the oil and gas industry – the chemicals sector. CCU will be the focus in this thesis.

Optimization is applied to explore the maximum performance of CCU. CCU contains multiple process options in both the capture and the utilization sections, eventually resulting in a large multi-process system. Optimizing such a multi-process system can be challenging because of the problem scale and its complexity. The problem scale is significantly larger than a single process and would be challenging to most existing optimization approaches; complexity comes from high-level interactions between sub-systems and the nonlinearity of the individual sub-systems. Optimizing a sub-system before extending it to the whole CCU system can lead to a sub-optimal solution due to the reduced decision space. Using one simulation result to represent a sub-system can neglect the complexity/nonlinearity of the individual processes. In this thesis, I intend to: (1) avoid sub-optimal solutions by simultaneously optimizing the CCU sub-systems, and (2) use surrogates to represent sub-systems to keep a certain complexity/nonlinearity of sub-systems.

This thesis is divided into two parts. **Part I** is methodology development, engaged in identifying suitable surrogate types for CCU sub-systems and how to obtain surrogates in an efficient way. The methodology development lays the foundation for an optimization framework for large multi-process systems. The optimization framework consists of three levels. Level 1 decomposes a large system into several sub-systems, which are digitalized by

rigorous process models. Level 2 replaces rigorous process models with machine learning-based surrogates, as to efficiently evaluate mass and energy balances. Level 3 performs surrogate-based optimization. This optimization framework includes the interactions of sub-systems and optimizes sub-systems simultaneously. **Part II** is concerned with problem-solving, focusing on optimizing a CCU system (by the three-level optimization framework), where no renewables are involved. The result shows that CCU may be worse for greenhouse gas (GHG) emissions than the conventional (unabated gas) process, if operating conditions are not properly set. Single-objective optimization enables CCU to effectively reduce GHG emissions, and electrifying heating can further cut GHG emissions. Additionally, multi-objective optimization enables CCU to balance the competing criteria between environmental and economic aspects.

The methodology developed in this thesis can be applied to other multi-process systems. In the long term, net zero needs various low-carbon pathways, which might integrate different sectors and form multi-process systems. While their decarbonization performances are enhanced by optimization, the overall progress of net zero will be accelerated.

Publications

(1) Hao, Z., Caspari, A., Schweidtmann, A. M., Vaupel, Y., Lapkin, A. A., Mhamdi, A. Efficient hybrid multiobjective optimization of pressure swing adsorption. *Chem. Eng. J.* 423 (2021) 130248.

In this work, I was involved in conceptualization (supporting), process modelling, software, writing – original draft and editing. Chapter 3 is reproduced from this publication. Dr. Caspari supported the implementation of dynamic optimization in DyOS, and Dr. Schweidtmann proposed the hybrid optimization strategy.

(2) Hao, Z., Zhang C., Lapkin, A. A. Efficient surrogates construction of chemical processes: Case studies on pressure swing adsorption and gas-to-liquids. *AIChE J.* (2022) e17616.

In this work, I led conceptualization, methodology, process modelling, software, writing – original draft and editing. This publication is the key element for Chapter 5.

(3) Hao, Z., Barecka, M., Lapkin, A. A. Accelerating net zero from the perspective of optimizing a carbon capture and utilization system. *Energy Environ. Sci.* 15 (2022) 2139-2153.

In this work, I led the digitalization methodology, process modelling, software, writing – original draft and editing. Regarding the conceptualization, Prof. Lapkin proposed the concept of an industrial park for CCU, and I initiated the idea of separation of the renewables from CCU. The major part of this publication contributes to Chapter 6.

Conferences and Seminars

(1) Oral (online), “Efficient Surrogates Construction of Chemical Processes: Case studies on Pressure Swing Adsorption and Gas-to-Liquids”, Conference: Mathematics in (Bio)Chemical Kinetics and Engineering 2021, Shanghai, China, Oct 2021.

(2) Research Seminar (online), “Accelerating Net Zero from the Perspective of Optimizing a Carbon Capture and Utilization System”, MIT Energy Initiative, Massachusetts, US, Sep 2021.

(3) Research Seminar (online), “Superstructure Optimization for Process Design – Implementation on Aspen-MATLAB”, Process Engineering Seminar, Cambridge, Nov 2020.

(4) Research Seminar (online), “Kinetic Parameter Estimation for C-H Mechanism *via* ModelBuilder (gPROMS)”, PSE in Synthetic Chemistry, Cambridge, Oct 2020.

(5) Research Seminar (online), “Thermodynamics refresher”, Process Engineering Seminar, Cambridge, Aug 2020.

(6) Research Seminar (online), “Introduction to gPROMS and Aspen Plus”, Process Engineering Seminar, Cambridge, Jul 2020.

(7) Oral, “Optimal Design of Flowsheets *via* Surrogate-based Sub-Process Models: A Case Study of CO_2 Capture and Utilization”, Conference: 2019 AIChE Annual Meeting, Orlando, US, Nov 2019.

(8) Oral, “A Hybrid Stochastic/Gradient-Based Multi-Objective Dynamic Optimization of Vacuum Swing Adsorption”, Conference: 2019 AIChE Annual Meeting, Orlando, US, Nov 2019.

(9) Poster, “Sequential Dynamic Optimization of Multi-Stage Vacuum Swing Adsorption”, Conference: 2019 AIChE Annual Meeting, Orlando, US, Nov 2019.

(10) Poster, “Optimization of Vacuum Swing Adsorption Process *via* Artificial Neural Networks”, Conference: RSC Reactors and Separation, London, UK, Sep 2019.

(11) Poster, “Optimization of Flowsheets *via* Surrogate Models”, Conference: Machine-learning in (Bio-) Chemical Engineering, Cambridge, UK, Jul 2019.

(12) Poster, “An Efficient Approach to Kinetic Parameter Estimation through Dynamic-Model-based Design of Experiments” (Best Poster Prize), Conference: ChemReactor-23, Ghent, BE, Nov 2018.

(13) Audience, “CPES Summer School in Decision Making Under Uncertainty”, Imperial College, London, UK, Sep 2018.

(14) Audience, “gPROMS Training”, Process Systems Enterprise Limited. London, UK, Nov 2017.

Contents

List of Figures

List of Tables

Nomenclature

LCP	Low-carbon process
CCUS	Carbon capture, utilization and storage
CCU	Carbon capture and utilization
CCS	Carbon capture and storage
GHG	Greenhouse gas
IEA	International Energy Agency
KKT	Karush-Kuhn-Tucker
SCN	Supply chain network
NLP	Nonlinear programming
MINLP	Mixed-integer nonlinear programming
ANN	Artificial neural network
GP	Gaussian process
SVM	Support vector machines
GS	Grid search
RS	Random search
NG	Natural gas
MEA	Monoethanolamine
MEA process	Monoethanolamine absorption for CO_2 capture
PSA	Pressure swing adsorption
DAC	Direct air capture
FT	Fischer-Tropsch
GTL	Gas-to-Liquids
DCHM	Direct CO_2 hydrogenation to methanol
MS	Methanol synthesis (*via* syngas)

RKS	Redilich-Kwong-Soave
NRTL	Non-random two-liquid
GA	Genetic algorithm
NSGA-II	Non-dominated sorting genetic algorithm-II
MSE	Mean square error
RMSE	Root mean square error
NRMSE	Normalized root mean square error (for multiple outputs)
LHS	Latin hypercube sampling
NGCC	Natural gas combined cycle power plant
NGCC-CCS	Natural gas power plant equipped with CCS

Chapter 1 Introduction

This chapter gives the motivations for (1) problem solving, why this thesis is focused on carbon capture and utilization, and (2) the methodology development, how optimization can be enhanced by machine learning-based surrogates, in order to solve the problem.

To limit global warming to 1.5~2 ℃ above pre-industrial levels, over 130 countries have pledged to cut CO_2 emissions to nearly zero (or 'net zero') by the mid of 21st century [2]. Net zero promotes all *clean* and *efficient* processes. A *clean* process requires the involvement of low-carbon technologies (*e.g.*, renewables, carbon capture, and nuclear). Many low-carbon technologies are still under development, and optimization can help achieve their maximum potentials – achieving *efficient* processes. Notably, process optimization is not a 'one-click' work: an optimizer requires some information from the corresponding process system and makes decisions iteratively.

To better understand how optimization works, this chapter starts with the basic optimization steps and then presents how these steps are related to process systems. Subsequently, I introduce the characteristics of low-carbon process (LCP) systems: large and complex. With these characteristics, carbon capture and utilization (CCU) can be a proper example of LCP to investigate. CCU contains two sections: CO_2 is captured from carbon sources, and then the captured CO_2 is further converted to valuable chemicals. To accelerate the optimization for CCU, machine learning-based surrogates can be an appropriate method. Additionally, limitations of prior research studies are pointed out, and potential solutions are outlined in the thesis structure.

1.1 Optimization

Optimization is a decision-making tool, which recommends the best options according to the given objectives. The systematic workflow of optimization follows three steps: formulation, algorithm selection and optimality check [3].

Optimization formulation is to identify the objective, decision variables and constraints. A typical formulation can be expressed as Eq 1.1 - Eq 1.2. Optimizers can minimize (or maximize) either one objective or multiple objectives, $f(x)$. In order to improve the objectives, the values of decision variables ($x = [x_1, x_2 \dots, x_n]$) can be adjusted during the optimization. Constraints determine the searching space (S) of decision variables. In engineering problems, objectives are commonly nonconvex (multiple local optimal solutions exist, see Figure 1.1) [4-6]; decision variables can contain continuous and discrete variables.

$$\min_{x \in S} f(x) \qquad \text{Eq 1.1}$$

$$S = \{g(x) \leq 0, h(x) = 0\} \qquad \text{Eq 1.2}$$

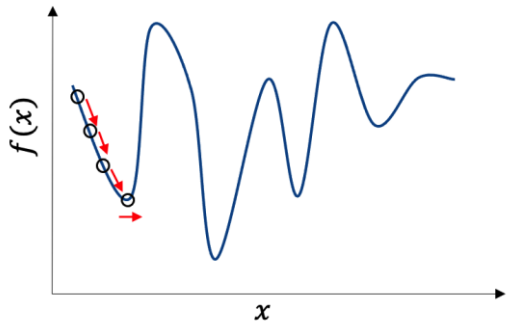


Figure 1.1. A deterministic method searches for an optimal solution for a nonconvex objective function.

Secondly, an optimization algorithm is employed to search for the optimal solution, usually implemented in an automatic manner as a computational solver. Optimization algorithms are iterative: an initial guess is given to x, and then optimization algorithms generate a sequence of improved points (or *iterates*). The optimization algorithms can be equivalent to searching strategies: the strategies, which are used to move from one iteration to the next, distinguish one algorithm from another [3]. Algorithms can be roughly classified into two types: deterministic

and stochastic methods. A deterministic method follows exact directions (*e.g.,* gradients) to improve but tends to get stuck in a local solution (see Figure 1.1). Inspired by nature, scientists have developed a variety of stochastic methods (*e.g.,* genetic algorithm), which use certain randomness to jump out of local basins [7]. As such, stochastic methods can access the entire decision space.

The third step is the optimality check, see Figure 1.1. The optimality check can help identify improvement space and a slight improvement may make a significant difference given the problem scale and time, especially in the industry. Biegler, one of the academic pioneers in process optimization, claims that the optimization study should go beyond the search for improved solutions towards a solution satisfying optimal conditions [8], such as Karush-Kuhn-Tucker (KKT) condition (using the first derivative to ensure the found solution is optimal) [9].

To make informative decisions, systems must offer the required information to the three optimization steps, see Figure 1.2. Ideally, all the equations are available, so that the process systems can provide all the required information (*e.g.,* gradients) to the optimization steps. However, a large process system may contain numerous unit operations, thus resulting in a big modelling task for engineers. The commercial simulation tools (*e.g.,* Aspen Plus, Pro II) facilitate engineers in building and evaluating process systems to a large extent. However, there is ‘no free lunch’ because most commercial simulators limit users’ access to the equations behind the unit operations. In other words, commercial software platforms can only be regarded as black-box simulators to users. The gradient calculation based on black-box simulations can be quite expensive, so the optimization is generally limited to the stochastic methods (then optimality cannot be guaranteed).

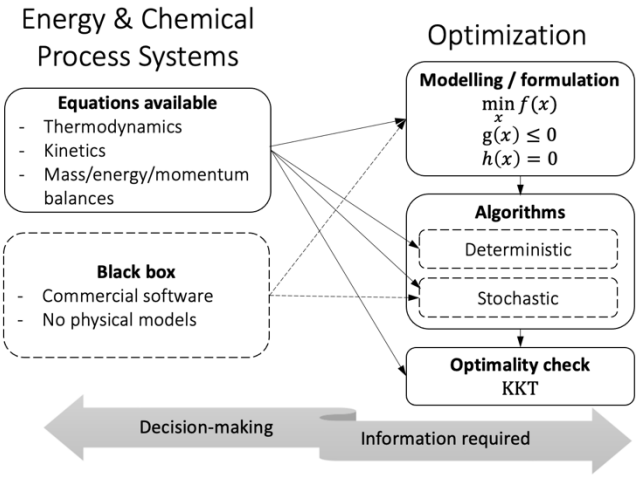


Figure 1.2. Optimization in process systems. Optimization is the decision-making tool to process systems, while process systems must provide the required information for optimization.

1.2 Low-carbon process (LCP) systems

Net zero promotes LCP systems. An LCP system commonly requires an integration of multiple sectors, ranging from energy, chemicals to bioprocess sectors. For example, power plants (energy) can be coupled with carbon capture (chemicals) [10, 11]; renewable energy can be employed to power the conversion of CO_2 to fuels [12-14]; bioenergy can be integrated with carbon capture as a negative emissions system [10, 11].

For LCP systems, I point out two general characteristics regarding their problem scale and complexity. The problem scale of an LCP system is typically larger than either an individual chemical or energy process system, which might only be a sub-system of the LCP. Although an LCP's problem scale cannot be compared to a supply chain network (SCN), the complexity of LCP is considerably higher than SCN. This is because individual nodes in SCN have been well established. Thus, optimization of SCN can be formulated as a linear programming (LP) or a mixed-integer linear programming (MILP) problem. By contrast, most LCP systems are still in the development stage, so the design and operation of LCP's sub-system are likely to be modified for the benefits of overall system. Most of systems are inherently nonlinear between design variables and system performances [15]. As such, an LCP's complexity can come from the nonlinearity of the individual sub-systems and interactions between the sub-systems, thus

resulting in nonlinear programming (NLP) or mixed-integer nonlinear programming (MINLP). Therefore, LCP systems are relatively large and complex.

Many LCP systems have not been commercialized yet, and optimization can help explore their maximum performance. It is desired to optimize all the sub-systems simultaneously, so that the interactions of sub-systems can be included. This thesis uses an LCP example system (*i.e.*, CCU) with the large-complex characteristics and then develops an optimization framework.

1.3 Motivation for surrogate-based optimization for CCU

To optimize a large and complex LCP system (*e.g.*, CCU), the first task is to model the whole system. Using the commercial simulators can accelerate the modelling task but also causes the following issues during the optimization:

(1) Some simulations can be considerably slow when the systems involve multi-scale, multiphase phenomena and dynamic behaviour [16]. Suppose a large system can be divided into several sub-systems and the simulation of sub-system A is more computationally intensive than other sub-systems. To simulate this large system, should other sub-systems just halt before the simulation of sub-system A is completed?
(2) Identifying a suitable simulator would be a challenge. CCU can closely interact with the electricity (generation) source and produce fuels, thus leading to a large system including both electricity and chemicals sectors. Yet, the current industrial practice involves the application of tailored simulators for specific sectors. Thus, sub-systems might be established in different simulators, while interfaces between different simulators are required to be developed.
(3) Due to no access to the equations behind the processes, the gradient information is almost unavailable to optimizers. Without gradient, the searching algorithms are limited to stochastic methods (no guaranteed KKT optimality).

To solve the expensive-simulation issue, data-driven surrogates can be employed to represent the original physical models by building a statistical relationship between process inputs and outputs. Surrogates are cheap-to-evaluate and can be directly employed for optimization. Various surrogate formulations exist, while the significant advances in machine learning have expanded the ‘surrogates family’. For example, artificial neural networks (ANNs) are regarded

as universal approximators [17], and allow to fit multiple output variables simultaneously. Additionally, surrogates can be generated and assembled in a high-level platform (MATLAB, Python), which can interact with low-level simulators for inputs/outputs datasets. As such, an LCP system can be evaluated simultaneously, when all of its sub-systems are represented by surrogates.

Surrogate-based optimization refers to the methodology, where surrogates are used for the function evaluations during the optimization iterations. This method possesses three advantages, as follows:

(1) This method can significantly decrease the computational cost of optimization.
(2) These surrogates of processes from different industry sectors can be assembled in one platform.
(3) Surrogates can be simultaneously accessed, so parallel computing is easy to implement.

1.4 Limitations of prior works

Although extensive prior studies have been performed on the CCU optimization, there are several limitations and important research questions to address.

The first one is related to the methodology regarding optimality. Several works apply surrogate-based optimization method to optimize CCU sub-systems, and conclude that the surrogate-based optimization is much more efficient than the optimization based on rigorous models [18-20]. However, the optimality is hardly discussed. This is probably because the gradient information is difficult to extract, and the optimality cannot be checked in this method. The question here falls on whether optimality check is necessary. In other words, how far the best solution found by surrogate-based optimization is from the optimality.

The second one is related to the methodology regarding surrogates. Prior works apply a specific surrogate to mimic sub-systems of CCU [18-20] but do not compare different surrogate types. Schweidtmann *et al.* review various surrogates types (ranging from linear, Gaussian process, Ensemble tree models to neutral networks) and their potential applications in chemical engineering, but little comments are given to compare their performances [21]. Boukouvala *et al.* explored three surrogate types (Gaussian process, general quadratic, signomial) and applied

them in the optimization of pressure swing adsorption (PSA) [19]. Gaussian process was found the most suitable surrogate type for the regression of PSA. Nevertheless, limited works can be found for comparing the surrogate types for CO_2 utilization processes. The question here falls on whether there exist CCU-specific suitable surrogate types.

The third one is related to problem-solving regarding problem scale and complexity. Extensive works have been done to maximize the performance of the individual sub-systems of CCU, *e.g.*, PSA [18, 22-24], MEA [25, 26], methanol synthesis [20, 27, 28] and Fischer-Tropsch [29, 30]. However, these sub-systems can interact with each other, and, thus, the individual optimal solutions may not simultaneously exist. Further, several research works on CO_2 utilization assumes that the renewables (H_2 or electricity) are the system input, which concludes that the inclusion of renewable energy sources is indispensable to achieve emissions reduction [31, 32], and also the cost of renewables is considered to be the limiting factor for the economic viability [27, 31, 33]. However, the CO_2 source is commonly not mentioned (the CO_2 capture step is neglected), and the assumption of renewables involvement is not realistic in the short future due to the limited availability of renewable sources, and how to deal with the intermittency of renewables are not well considered. Another strategy is to optimize a sub-system before extending to the whole CCU system, but such a strategy is similar to the "onion" strategy, where the optimization search initially goes to a sub-optimal solution due to the reduced decision space [34]. Inspiringly, Roh *et al.* optimize a CCU system as a supply chain network, but the complexity/nonlinearity of the individual sub-systems are not considered [32]. The research question falls on how to efficiently optimize a large CCU system, while its complexity is retained.

The fourth one is related to problem-solving regarding interactions. Such interactions can be the competition for the limited resource, waste recycling and heat integration among sub-systems. However, little research is conducted to contain interactions in the optimization of CCU systems [32]. The research question falls on how to include the interactions between sub-systems inside the optimization formulation.

1.5 Goals and structure of this thesis

This thesis is aimed to solve the above-mentioned limitations of prior works. I develop a surrogate-based optimization methodology and then apply it to solve a problem – optimizing an industrial park of natural gas-based power plants integrated with CCU. Initially, this industrial park contains several but not too many sub-systems, which can be a suitable starting point to testify the optimization methodology. For this industrial park, a surrogate-based optimization workflow can be shown in Figure 1.3. First, the industrial park is decomposed into sub-systems and then digitalized in commercial simulators. Simulators can generate sufficient input-output datasets for sub-systems, which are further digitalized to machine learning-based surrogates, respectively. These cheap-to-evaluate surrogates can then be assembled in one platform for optimization. Eventually, the optimizer can perform a decision-making task for the operating conditions of the industrial park.

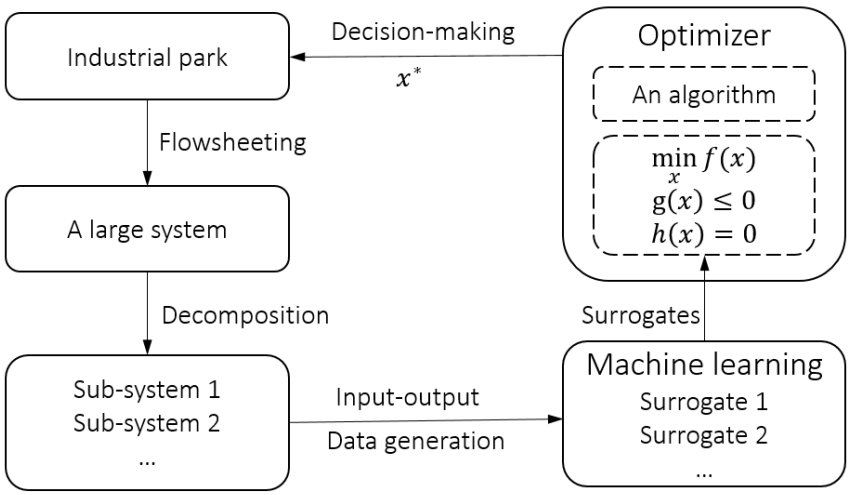


Figure 1.3. Optimization of a large system (on a scale of an industrial park) *via* surrogates.

Following the Introduction (*Chapter 1*) and Literature review (*Chapter 2*), I divide the remaining thesis into two parts. **Part I** is the methodology development, covering the development of the surrogate-based optimization, described in *Chapters 3 - 5*. **Part II** is concerned with problem-solving, focused on optimizing CCU systems, described in *Chapters 6* and *7*. In brief, these chapters include:

Chapter 2 reviews the literature on optimization and decarbonization. In the optimization part, an overview of surrogate formulation as well as optimization algorithms is summarized in this

chapter. For the decarbonization part, I will review the importance of CCU under the scope of net zero. Following this, I survey typical carbon capture pathways and utilization pathways. Further, the evaluation methods for CCU are presented.

Chapter 3 addresses the first limitation related to the optimality check. A hybrid method is applied to optimize pressure swing adsorption (PSA). A stochastic method is initially performed by the evolutionary algorithm combined with Gaussian Process (GP) surrogate; subsequently, the obtained solutions are refined by a gradient-based algorithm combined with the full physical model. This chapter will test how close the results, based on surrogate-based optimization, can reach optimal solutions.

Chapter 4 explores which surrogate type is suitable to regress the process systems. I use a direct CO_2 hydrogenation to methanol process as a toy problem and test various surrogate types, which are then shortlisted based on their fitting performance on a single output. Following this, the shortlisted options are further tested to fit multiple outputs simultaneously and then the surrogate-based optimization. The performance of ANNs is similar to the GP, while ANNs show better flexibility for fitting multiple outputs. Whatever surrogates to use, the hotspot is the time for data generation.

Chapter 5 explores an efficient method for generating surrogates *via* ANN. My strategy is to develop a workflow to reduce the total time spent on data generation by (1) lowering the total number of the required data points; (2) shortening the time per data generation. PSA and Gas-to-Liquid (GTL, including natural gas reforming and Fischer-Tropsch) processes are used as two case studies.

Chapter 6 employs the ANN-based surrogates to optimize the industrial park of power plants integrated with CCU. In this chapter, I intend to: (1) avoid sub-optimal solutions by simultaneously optimizing the whole CCU system, and (2) use surrogates to represent sub-systems to retain the complexity/nonlinearity of sub-systems. Multi-objective optimization is performed for the industrial park regarding emissions reduction and economic gain.

Chapter 7 examines whether surrogate-based sub-systems can capture their interactions. The feasibility is checked by a case study on a reactor-separator-recycle system. Following this, I

re-visit a complex CCU system to check whether the interactions of sub-systems influence process design and synthesis.

Chapter 8 concludes this thesis and provides recommendations for future research directions.

Chapter 2 Literature review

Chapter 2 reviews the literature on optimization and decarbonization. In the optimization part, I survey some surrogate formulations for efficient function evaluation. To employ surrogates in optimization, the stochastic searching method is the common choice for the type of optimization algorithms, but the slow convergence is its drawback. As a complement, I revisit the classical method - the deterministic searching method. For the decarbonization part, I will review the high-level scope of net zero. Notably, fossil fuels are projected to play a dispensable role until 2050, even under the net-zero scenario analyzed by IEA [1]. Net zero requires fossil fuels to be burned in a clean way, which can be achieved by carbon capture technology. The two carbon capture divisions - CCS and CCU - are discussed, and I find that CCU is more suitable to be evaluated in a local region. Following this, I review carbon capture pathways and utilization pathways. Last but not least, the evaluation criteria for CCU are presented.

2.1 Optimization

For a large process system, the optimization efficiency depends on evaluating individual *iteration*s and searching strategy (from one *iteration* to the next). An *iteration* evaluation on a large system can be expensive. Additionally, the difficulty in searching strategies results from:

(1) objectives: nonconvexity, multiple objectives.
(2) decision variables: continuous/discrete, high dimensionality.
(3) constraints: linear/nonlinear function.

To develop an efficient optimization framework, I will review the cheap-to-evaluate modelling (surrogates) and optimization algorithms (searching strategies).

2.1.1 Surrogate-based optimization

The transformation to Industry 4.0 is driven by advancements in digitalization [35]. As a foundation of digitalization, digital twins, referring to surrogates in this work, can represent the physical assets within the cyber domain and play an essential role in evaluating engineering systems [36].

The evaluation of real-world engineering systems through experiments is expensive. An alternative way is to build physical models (often refers to first-principle models) to describe the mechanism of engineering systems. With physical models and inputs, computer simulations can accurately deliver information about systems. Although cheaper than experimental or industrial data, simulations can be considerably slow when the systems involve multi-scale, multiphase phenomena and dynamic behaviours [16]. Data-driven surrogates can represent the original physical models by building the relationship between inputs and outputs. For some systems with complex mechanisms (*e.g.*, formulations in pharmaceuticals and cosmetics), there are even no physical models available, and thus surrogates together with the design of experiments (DoE) seem to be the only choice [37-39].

Surrogate-based optimization refers to the methodology, where surrogates are employed for the function evaluations during the optimization iterations. This type of method has been introduced to solve design problems in many engineering areas, *e.g.*, chemical engineering [40-43], pharmaceutical manufacturing [44], supply chain management [45, 46], and aerospace engineering [47-49].

Long before adopting surrogate-based optimization, chemical engineers applied a coarse-to-fine strategy for process optimization: shortcut methods can be used to identify the bounds and narrow down the decision space, followed by robust optimization using rigorous simulations subsequently [50]. Some reactors can be simplified by modelling them as continuous stirred tank reactors (CSTRs) coupled with equilibrium reaction, while shortcut distillation can be referred to Fenske-Underwood-Gilliland (FUG) method. Such simplified models can accelerate coarse evaluations, but the traditional knowledge in chemical engineering cannot offer shortcut models to all unit operations, such as membranes or crystallizers. Regardless of

the types of unit operations, data-driven surrogate models only correlate the necessary inputs to outputs of the robust simulations, thus simplifying full-order physical models.

Besides the conventional surrogate formulations like polynomials, the booming of machine learning has expanded the 'surrogate family' [16, 41, 42, 51] with more choices, *e.g.*, artificial neural networks (ANNs), Gaussian Process (GP), Support vector machine (SVM) and Ensemble Tree models [52-55].

2.1.2 Surrogate training

To train a surrogate model, rigorous simulations are called to generate sufficient datasets. Active learning (also referred to sequential sampling, adaptive learning, design of experiments in research works) can be used to identify the effective sampling points and reduce the times of required simulations. Active learning iteratively samples data points and refines the surrogate models. There are two strategies for applying active learning to surrogate-based optimization, as shown in Figure 2.1. The first strategy is a more conventional way: surrogate models are trained to reach an acceptable accuracy and then used in optimization. Herein, the adaptive sampling is for the accuracy improvement of surrogate models. The second strategy trains the surrogate models and explores their application in optimization in each iteration (*e.g.*, TSEMO [54]). Notably, the adaptive sampling here is to explore further improvement possibilities in optimization.

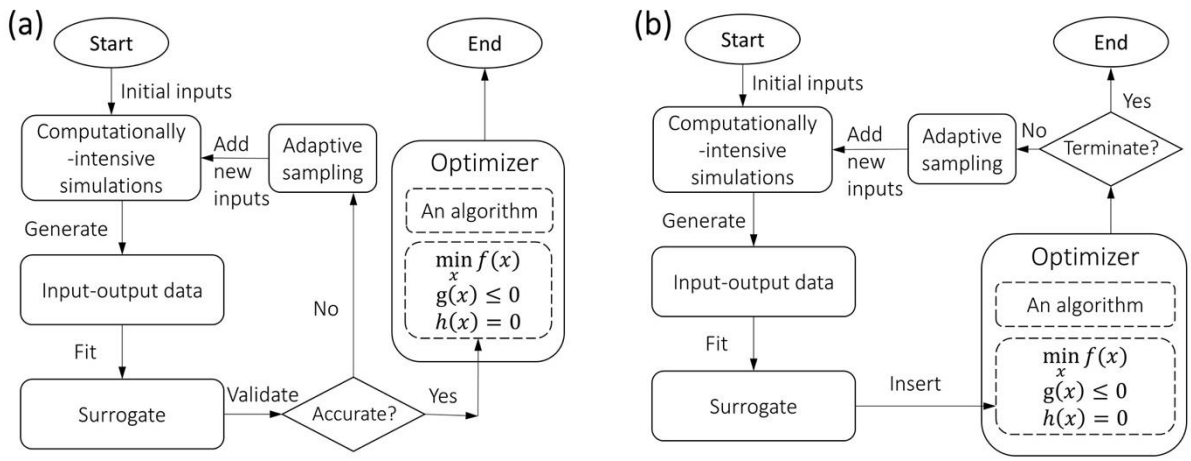


Figure 2.1. Two strategies of applying active learning to surrogate-based optimization. (a) Surrogate models are trained iteratively to be accurate enough, followed by optimization; (b) surrogate models are trained and used in optimization simultaneously in each iteration.

2.1.3 Machine learning-based surrogates

This section introduces several machine learning-based surrogate types.

2.1.3.1 Artificial neural networks

Artificial neural networks (ANNs) are inspired by how biological neural networks process information. A shown in Figure 2.2a, a typical ANN is established by numerous connected units called neurons, which are arranged into successive layers. A neuron can be regarded as the smallest input-output unit, containing three elements: neuron input, activation function and neuron output [56]. Take an example with the neuron k as shown in Figure 2.2b. The neuron outputs from the previous layer will be the inputs for neuron k; each input is multiplied by a weight; the sum of all weighted neuron inputs is added with a bias; the obtained information is further processed by an activation function (Eq 2.1) to limit the amplitude of the output of neuron k, eventually resulting in neuron output y_k [56].

$$y_k = \varphi(\sum_{i=1}^{m} w_{ki}x_i + b_k) \qquad \text{Eq 2.1}$$

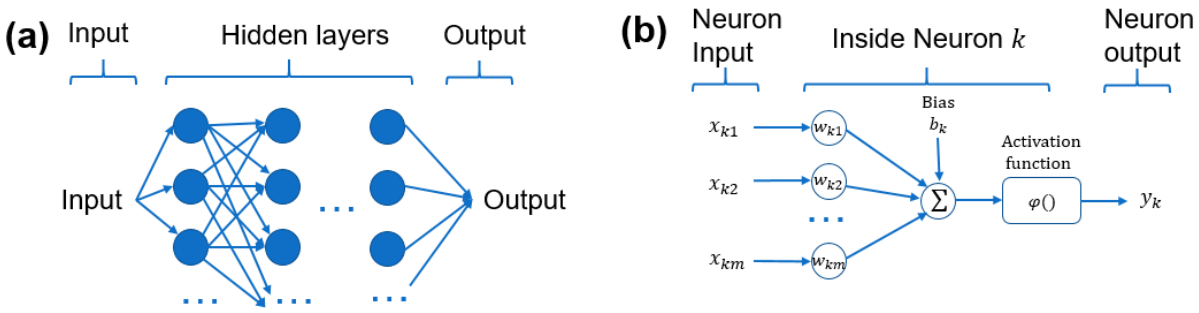


Figure 2.2 (a) Structure of an artificial neural network; (b) structure of an artificial neuron.

In analogy to biological neural networks, one activation function mimics how a neuron transfers the obtained information to a subsequent neuron. In ANN, an activation function determines the output of a neuron, as shown in Eq 2.1. There are numerous activation functions, which can be classified into linear and nonlinear types. For the linear type of activation function, the respond y_k is linearly dependent on the neuron input x_k, but the regression may not perform well for complex nonlinear systems. For the nonlinear activation functions (Sigmoid, tanh, ReLU, Gaussian), one hidden-layer neural network is reported to be a universal function approximator [17].

There are both advantages and disadvantages in employing ANNs. It is easy to implement ANNs to regress multiple outputs simultaneously. The flexible structures and various activation functions enable ANNs to accurately fit any linear or nonlinear relationship of input/output. However, it is complex to identify the best structure of a network because the number of potential structures is infinite. This requires the design of the parameters of networks (activation function types, layer number, neuron numbers) and hyperparameters (learning rate, weights, bias). All of these will make the training of ANNs computationally demanding. Two common methods are grid search (GS) and random search (RS): GS sets up a grid of parameter values, which are trained and evaluated; while RS evaluates random combinations of parameter values. RS is more recommended than GS. This is because different parameters may have different extents of influence on ANNs and some parameter combinations may not be meaningful. In practice, it is commonly unknown which parameter is more critical before optimizing values of parameters; RS offers more possibilities to assess important parameters than GS [57]. To demonstrate it, Figure 2.3 presents an example of searching the values of two parameters. One parameter has a dominated role in affecting the model, as the ‘green filling’,

while the other one has a negligible influence as the ‘yellow filling’. Both GS and RS evaluate the search space by 9 sampling positions. As a result, GS can only evaluate the ‘green filling’ three times, while RS can evaluate it nine times [57].

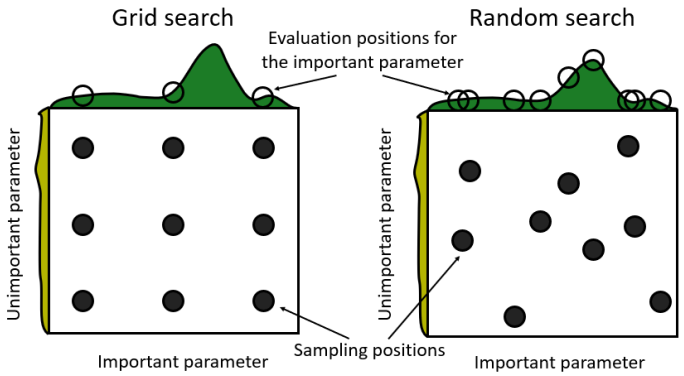


Figure 2.3. Grid search (GS) and random search (RS) for the optimization of ANN parameters. Adapted from [57].

2.1.3.2 Gaussian Process Regression

Gaussian Process (GP) Regression, often referred to as Kriging, is a non-parametric model type, which has excellent regression performance and can estimate the uncertainty of prediction [53]. A general form of GP models consists of two parts: a deterministic term and a noise term [41] as shown in Eq 2.2.

$$Y = \sum_{i=1}^{m} \beta_i f_i(X) + \varepsilon(X) \qquad \text{Eq 2.2}$$

$$cov[\varepsilon(x_i), \varepsilon(x_j)] = k(x_i, x_j) \qquad \text{Eq 2.3}$$

In Eq 2.2, $f_i(X)$ refer to m independent basis functions; β_i refer to unknown parameters; the first term, $\sum_{i=1}^{m} \beta_i f_i(X)$ is used to describe the trend of mean prediction at location X; the second term, $\varepsilon(X)$, is a noise term with the mean value as 0 at location X. The covariance function of the noise term is called kernel (Eq 2.3), which determines the type of GP models. Kernel defines how a GP model generalizes or extrapolates new data points [53].

When using GP models as surrogates, one key issue is to choose the suitable kernel. Bhosekar *et al.* summarize the common kernels used in chemical process systems as shown in Table 2.1

[41]. Within these model equations, m_j refers to the distance between two points; θ_j and p_j are hyperparameters; d is the dimension of the input variables. Specifically, for Matern model, Γ and K_{v_j} are the Gamma function and the modified Bessel function of order v_j, respectively. More information concerned with different kernels can be referred to Duvenaud's thesis [53] and other reviews [41, 42].

Table 2.1. Common kernel models [41].

Kernel types	Kernel models
Exponential	$\exp\left(-\sum_{j=1}^{d}\theta_j\lvert m_j\rvert^{p_j}\right), 0 < p_j < 2$
Squared exponential	$\exp\left(-\sum_{j=1}^{d}\theta_j\lvert m_j\rvert^{2}\right)$
Linear	$\max(0, 1-\sum_{j=1}^{d}\theta_j\lvert m_j\rvert)$
Spherical	$1-1.5\xi_j+0.5\xi_j^3, \xi_j=\min(0, \sum_{j=1}^{d}\theta_j\lvert m_j\rvert)$
Matern	$\prod_{j=1}^{d}\frac{1}{\Gamma(v_j)2^{v_j-1}}(\theta_j\lvert m_j\rvert)^{v_j}K_{v_j}(\theta_j\lvert m_j\rvert)$

One advantage of GP is to estimate the prediction uncertainty in the whole design space. The regions with large model uncertainty correspond to the highly nonlinear/complex regions, which require a higher sampling probability, as to increase the overall model performance [42]. GP is recommended for problems with an input dimension below 20, because fitting a GP model can be computationally intensive when more dimensions are involved [42]. Further, an individual GP is generally implemented for a single output variable, whereas the formulation for regressing multiple output variables is complex [58, 59].

2.1.4 Optimization algorithms

Mathematicians, computer scientists and engineers have developed numerous optimization algorithms. Algorithms can be classified into various subsets, based on the different characteristics, *e.g.*, convex or nonconvex objectives, continuous or discrete decision variables, constrained or unconstrained, single or multiple objectives, static or dynamic. In this thesis, I classify the algorithms based on whether their searching strategies are exact or involve randomness, thus resulting in two types: deterministic and stochastic types of methods (Table 2.2). Most deterministic methods work well for local search, while stochastic methods can take advantage of the randomness to escape from local optima. Additionally, the classification is not that binary, *e.g.,* a hybrid method can have characteristics of both methods. Figure 2.4 shows how these methods move their *iterations.*

Table 2.2. Deterministic *vs.* Stochastic types of searching methods.

Searching types	Deterministic (exact search)		Stochastic (randomness involved)	
Characteristics	Gradient-based Single point	Relaxation Global	Randomness Single point	Randomness Population-based
Algorithm examples	Newton types Steepest descent	Branch-and-bound	Random search Simulated annealing Tabu search GRASP	Evolution algorithms Swarm intelligence
Pros	Efficient in high dimensions Optimality guaranteed	Global optima	Global search Gradient is not required Easy parallel computing	
Cons	Gradient required Local optima	Expensive Need relaxation Gradient required	Inefficient in high dimension	

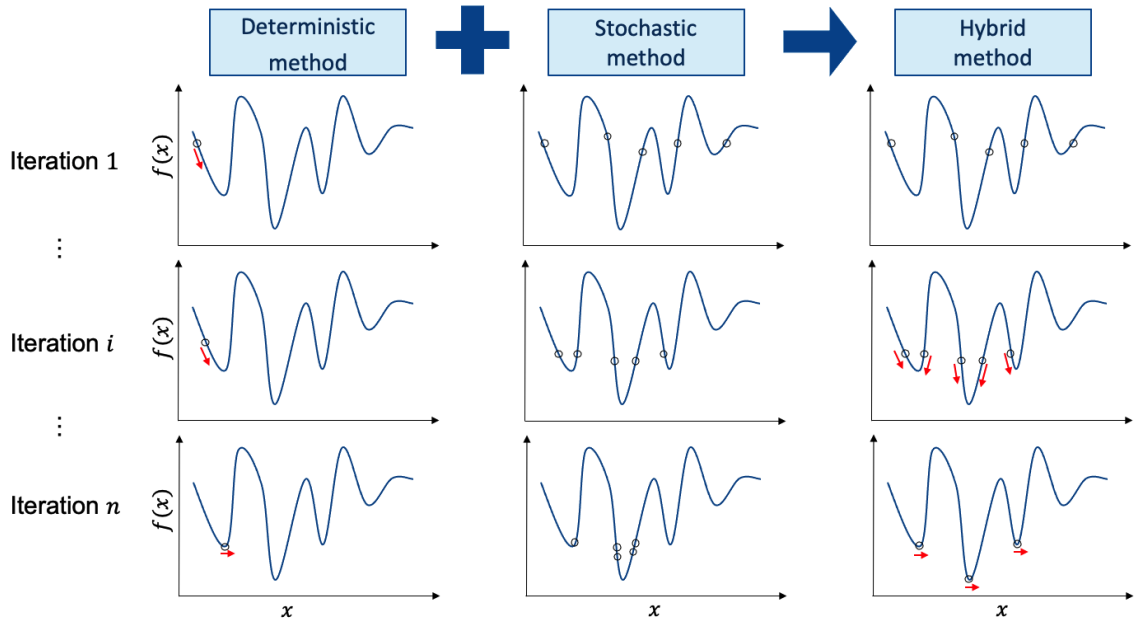


Figure 2.4. Types of optimization methods (searching strategies for the improvement from $Iteration\ i$ to $Iteration\ i+1$): a deterministic method follows a certain path (*e.g.*, gradient) to move the iteration; a stochastic method introduces a certain randomness into searching, and can move the iteration to a better direction or a worse direction; a hybrid method can be developed by combining stochastic and deterministic methods together (a typical example is

to employ a stochastic method for global search, and then switch to a deterministic method for local search).

Many deterministic algorithms are gradient-based, *e.g.*, Newton types. These algorithms employ the gradient information, *e.g.*, the first derivative, to guide the searching direction. They have been successfully applied in linear and nonlinear problems. Nevertheless, the objective functions of engineering problems frequently involve the features of nonconvexity or discontinuities [4-6]. These features may cause the failure of gradient-based algorithms because: (1) the gradient tends to limit the searching in a local basin, (2) discontinuities lead to the failure in calculating the derivative information (not differentiable). Since gradient-based optimization was developed earlier, most software can access several algorithms. They are capable of searching efficiently, even in high dimensions. More details can be found in Section 2.1.4.1.

Besides gradient-based methods, there are other types of deterministic methods. The branch-and-bound method is a deterministic type, and it can be used to determine the global optima. The key is to compute lower bounds for the subsets of decision space and iteratively narrow down the desired decision space. To calculate the lower bounds, relaxations (some also require the gradient information) are needed. Dr. Bongartz applied this method to the global optimization of process systems in his PhD thesis [60], but mentioned this method is computationally expensive. However, this method is beyond the scope of this thesis.

The deterministic methods commonly generate various single-point-based algorithms. Inspired by nature, computer scientists introduced population-based stochastic algorithms for optimization. These algorithms randomly start with a set of initial points (population of solutions), and new points are generated in a stochastic way, eventually pushing these points to move to better solution areas along with iteration [61].

This thesis focuses on gradient-based methods and stochastic methods, which will be reviewed in the following sub-sections.

2.1.4.1 Gradient-based methods

Let us start with a simple case - an unconstrained optimization problem,

$$\min_x f(x) \qquad \text{Eq 2.4}$$

Gradient-based methods employ the gradient to guide the search direction. When the objective function $f(x)$ is differentiable, the gradient can be calculated by its first derivative.

$$\nabla f(x) = [\frac{\partial f}{\partial x_1}(x); \frac{\partial f}{\partial x_2}(x) \dots \frac{\partial f}{\partial x_n}(x)\,] \qquad \text{Eq 2.5}$$

When the objective function $f(x)$ is not differentiable, each partial derivative can be estimated by finite difference methods.

$$\frac{\partial f}{\partial x_i}(x) \cong \frac{f(x_1, .., x_i + \delta, \dots, x_n) - f(x_1, .., x_i, \dots, x_n)}{\delta} \qquad \text{Eq 2.6}$$

where, δ is a very small value.

The gradient, $\nabla f(x)$, is the direction of steepest slope, while $-\nabla f(x)$ is the fastest decreasing direction. The pseudocode of a gradient-based method can be as follows:

Gradient-based methods [62]

Generate a random initial point x_0, set $k = 0$

While-loop (until the termination criteria)

  Calculate gradient $\nabla f(x_k)$

  Adjust the direction p_k (*e.g.*, Inverse Hessian matrix for Newton's method)

  Determine step size α_k

  Obtain the new point $x_{k+1} = x_k - \alpha_k \cdot p_k \cdot \nabla f(x_k)$

  To new iteration $k = k + 1$

End while-loop

The way to compute the direction and step size results in various algorithms. *Steepest descent method*, the earliest approach for numerical optimization [62], follows the steepest descent direction ($p_k = 1$) to search the optima iteratively. *Newton's method* adjusts the direction by

the inversion Hessian matrix ($p_k = H_k^{-1} = \frac{1}{\nabla^2 f(x_k)}$). The use of Hessian can make Newton's method self-scaling, but the computation of Hessian matrix can be expensive [62]. The different approximation methods for the Hessian matrix lead to a set of quasi-Newton methods, such as BFGS, TN, LBFGS [62]. Additionally, the step size is an important factor: a too-small step size can cause slow convergence, while a too-large step size can fail to converge by ending up bouncing around the optima. A dynamic step size can improve the searching efficiency: the step size is initially large and then iteratively decreases. *Linesearch* method determines optimal step size in every iteration (Eq 2.7) [62].

$$\min_{\alpha} f(\alpha) = x_k - \alpha_k \cdot p_k \cdot \nabla f(x_k) \qquad \text{Eq 2.7}$$

Newton's method has a broader application in developing gradient-based algorithms [63]. In fact, *Newton's method* aims at solving the algebraic equation for the necessary condition of optimality (Eq 2.8), and a typical Newton iteration can be expressed in Eq 2.9.

$$\nabla f(x^*) = 0 \qquad \text{Eq 2.8}$$

$$x_{k+1} = x_k - \frac{\nabla f(x_k)}{\nabla^2 f(x_k)} \qquad \text{Eq 2.9}$$

Newton's method can be extended to solve a constrained optimization problem (constrains as shown in Eq 2.11).

$$\min_{x} f(x) \qquad \text{Eq 2.10}$$

$$g(x) \leq 0 \text{ and } h(x) = 0 \qquad \text{Eq 2.11}$$

The Lagrangian function is defined as $L(x, u, v) = f(x) + g(x)^T u + h(x)^T v$, where u and v are multipliers for the inequality and equality constraints, respectively. The necessary condition for optimality: the optimal value (x^*) should satisfy Karush-Kuhn-Tucker (KKT) condition, as shown in Eq 2.12 -Eq 2.14 [63].

$$\nabla_x L(x^*, u^*, v^*) = \nabla f(x^*) + \nabla g(x^*)^T \cdot u^* + \nabla h(x^*)^T \cdot v^* = 0 \qquad \text{Eq 2.12}$$

$$g(x^*) \leq 0 \text{ and } h(x^*) = 0 \qquad \text{Eq 2.13}$$

$$g(x^*)^T u^* = 0 \qquad \text{Eq 2.14}$$

Similarly, *Newton's method* can be used to solve the algebraic equation (Eq 2.12), which is the basis of constrained optimization methods (SQP, Interior point method, nested strategies) [63].

Overall, gradient-based methods require the computation of a first derivative. *Newton's method* (and other quasi-Newton methods) is a powerful gradient-based method but requires the second derivative. If the objective function is differentiable, the CPU time required for the first derivative evaluation is 3 ~ 5 times that required for objective function evaluation, regardless of the number of decision variables (n) [62]. However, the objective evaluation of the process systems may be simulation-based or black-box, and no equation is available, not to mention 'differentiable'. Consequently, the finite difference methods are needed to compute the gradient for process systems, and thus the CPU time of the first derivative evaluation grows to n times that of the objective function. This is because one objective function corresponds to n decision variables, which correspond to n partial differential equations of the first derivative (Eq 2.6,). Further, the second derivative for the Newton-type methods can even require n^2 times the CPU of objective evaluations. Hence, the gradient calculation can be CPU-intensive for a large process system. Last but not least, gradient-based methods get stuck in local optima so that an improper initial point can lead to a local optimum in an unattractive subset of searching space.

2.1.4.2 Stochastic methods

As a clarification, 'stochastic' in this thesis specifically refers to **search methods involving randomness**. 'Stochastic' can refer to metaheuristic methods in the field of computer science, where the metaheuristic methods are high-level and are claimed to be universal optimizers compared to the problem-specific heuristics [7]. In the context of optimization under uncertainty, 'stochastic' means **uncertainty modelling** [3], which is not covered in this thesis.

The most straightforward stochastic method is random search: new points are sampled in a purely random way. Random search can effectively access the entire searching space, but it is not efficient to exploit the local solution. Most advanced stochastic methods maintain a good balance between the exploration on the global search space and the exploitation on the local basins [7].

Stochastic methods can be classified by various criteria, while this thesis classifies stochastic methods into two categories: single point-based methods and population-based methods.

2.1.4.2.1 Single point-based methods

Single point-based methods start with an initial point and randomly move to a potentially better point in an iterative way. 'Potentially better' means a new point (Iteration $i+1$) can be better or worse than the current point (Iteration i), but the overall direction is improvement. The random worse direction can help algorithms jump out of the local basin.

Simulated annealing (SA) is a typical single point-based method. SA is inspired by the annealing technique in metallurgy, where a high-temperature metal cools down to reach an optimal arrangement of solid state [7]. When minimizing an objective function $f(x)$, the pseudocode of SA is as follows:

Simulated annealing (SA)

Generate a random initial point x_0 ($i = 0$) and initialize a high temperature T=T_0

For-loop (until the termination criteria)

  While-loop (until a fixed number of iterations)

    Randomly generate a candidate in the neighbourhood, $x^* = N(x_i)$

    If $f(x^*) \leq f(x_i)$, then $x_{i+1} = x^*$

    Else

    Choose a random $u \, \epsilon [0,1]$

    If $u <$ exp $[-\frac{f(x^*)-f(x_i)}{T}]$, then $x_{i+1} = x^*$, else $x_{i+1} = x_i$

    End If

    $i = i + 1$

End while-loop

Decrease T

End for-loop

The temperature is a parameter to monitor the optimization progress: the higher the temperature is, the more random the search is. SA starts with random search, under a high temperature T_0. Then the temperature gradually decreases. At each iteration, the current point (x_i) and randomly-generated candidate (x^*) are compared regarding their objective values $f(x_i), f(x^*)$. If $(x^*) \leq f(x_i)$, the candidate point is accepted as the new point in the next iteration. Even if $(x^*) > f(x_i)$, the candidate point can also be accepted with a probability $p = \exp\left[-\frac{f(x^*)-f(x_i)}{T}\right]$. The lower T is, the lower p is. In other words, SA can explore the global space in a quite random way in the beginning (equivalent to random search at high T), while the randomness gradually decreases with the reduction of temperature, so SA tends to exploit a local basin eventually. As such, the convergence of SA can be relatively fast [7].

Similarly, other types of single point-based stochastics methods can escape from local optima. More details can be referred to the review papers from Fouskakis et al. [64] and Boussaïd et al. [7]. Table 2.3 lists some methods and their characteristics.

Table 2.3. Summary of single point-based methods [7].

Methods	Characteristics
Random search	Randomly search globally
Simulated annealing	Inspired by annealing, search initially globally and eventually locally.
Tabu search	Inspired by human memory, escape local optima by the search history.
GRASP	Iteratively perform [global-local] search.
Iterated local search	Iteratively [perturb a local optimum and perform local search]

The major drawback of single point-based methods falls in the considerable computational time. Parallel computing can accelerate the search. For example, parallel SA algorithms can be implemented by employing multiple SA independently. However, the parallel SA is criticized for the low efficiency in the initial stages, because all SA solvers perform random search and such independent parallelism is a waste of computational cost [64]. Further, such logic can promote the methods in evaluating multiple interacted points in each iteration, referred to as population-based methods.

2.1.4.2.2 Population-based methods

Population-based methods evaluate a group of candidates in each iteration. Two popular method types are evolutionary algorithms and swarm intelligence.

Inspired by the Darwinian principles of natural selection (species try out to adapt to the environment), Turing proposed the concept of evolutionary search [65]. In the 1960s, scientists from different places implemented evolutionary algorithms (EAs) [65]. Within many variants of EAs, genetic algorithm (GA) is a typical one. The pseudocode of GA is as follows:

Genetic algorithm (GA)

Generate a random initial population;

For-loop (until the termination criteria)

Score individual points in population i;

Select the *parents* based on the scores;

Identify *elite* points based on their high scores;

Parents generate *children* points in *mutation* by making random changes;

Parents generate *children* points in *cross-over* by combining pairs of *parents;*

Elite, mutation and *cross-over* points are passed to the next population $(i + 1)$;

$i = i + 1$;

End for-loop

GA can be easily accessed in the MATLAB built-in function, *ga.* Most EAs follow a similar workflow to improve the populations in an iterative way. Inside each iteration (or population) of EAs, individuals with good performance can evolve to the next population, which also contains a certain percentage of randomly generated points to jump out of local basins. After iterative populations, the best candidates try out to adapt to the problem.

Inspired by the collective behaviour of social insect groups, swarm intelligence algorithms start with a population of points and these points move to better directions iteratively by interacting locally with other points and with their environment. Particle swarm optimization (PSO) algorithm is a typical example [61], with the pseudocode as follows:

Particle swarm optimization algorithm (PSO)

Generate position and velocity randomly for each particle

For-loop (until the termination criteria)

    Evaluate individual particle's fitness values

    Determine individual particle's best positions

    Determine the best position for the entire particles

    Updates individuals' velocities and positions based on their neighbourhoods

End for-loop

Swarm intelligence includes many similar algorithms, such as ant colony optimization, brainstorm optimization, fireworks algorithms, firefly algorithms and artificial bee colony algorithms, *etc.* [7, 61]. PSO is demonstrated here only because it can be easily accessed in the MATLAB built-in function, *particleswarm*.

2.1.4.2.3 Comments on stochastic methods

All stochastic methods can use certain randomness to jump out of local optima, so theoretically, all can explore the global domain. Stochastic methods are universal solvers for any problem because they are not problem-specific and are mainly based on the simulation inputs/outputs. To accelerate optimization, parallel computing can be easily implemented for stochastic methods. Notably, population-based methods may be more efficient than parallel single point-based methods. This is because population-based methods not only allow a certain degree of random, but also consider the iterations among individuals (*e.g.*, cross-over in GA).

Stochastic methods belong to space-fill methods, which inevitably severely suffer from the curse of dimensionality. Theoretically, stochastic methods can reach the optimality only in the limit of infinite number of searches. As a result, stochastic methods cannot guarantee optimality in practice. Slow convergence is a drawback for these algorithms.

2.1.4.3 Hybrid methods

The motivation to develop hybrid methods is that an individual method cannot provide perfect performance. The progress of a stochastic method is rapid initially but flattens out later. For example, I run 50 iterations of GA to optimize a CCU system. The optimization progress can be equally divided into two halves. As shown in Figure 2.5, the improvement in the first half is more significant than the second half.

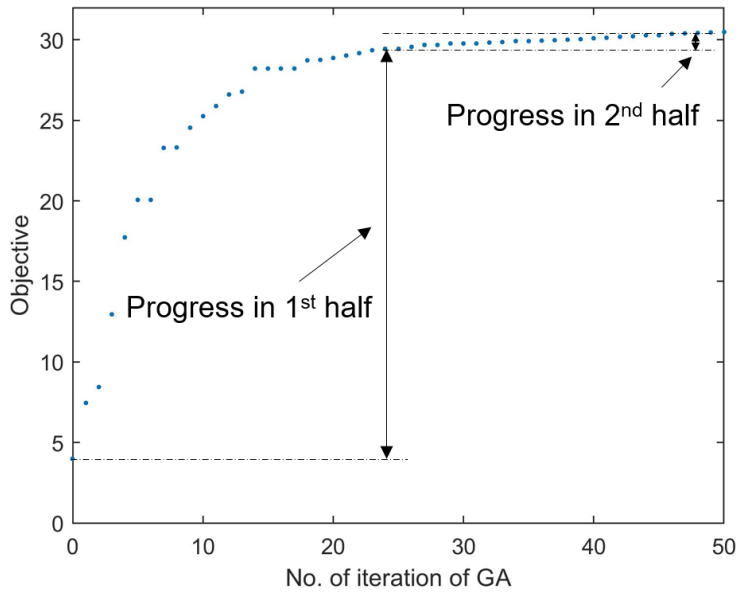


Figure 2.5. Progress of a stochastic algorithm – GA, illustrated by the maximization of an objective of a CCU system.

A stochastic method can rapidly identify good areas of the decision space (exploration). Still, they suffer an inefficiency in exploiting the local basins and cannot guarantee optimality. By contrast, gradient-based methods can use *linesearch* to determine a suitable step size to approach optima in a more efficient way than stochastic methods. A more efficient method might be to search globally by a stochastic method and then accelerate the optimality by a gradient-based method – a hybrid method [24, 65-68].

A hybrid method can integrate the complementary advantages of the individual methods. The concept of hybrid optimization methods – a synthesis of a global solver with a local solver – was initially proposed by computer scientists to solve nonconvex problems many years ago

[66-68]. I apply this concept to solve a process engineering problem and show that a hybrid method can outperform either of the parent algorithms alone, which will be demonstrated in Chapter 3.

2.1.5 Summary

Section 2.1 discusses the challenges in process optimization: (1) the expensive function evaluations; (2) nonconvex issues for objectives; (3) high dimensionality of decision variables; (4) optimality. To solve these challenges, I review potential solutions - surrogate and optimization methods (Table 2.4). However, none of them can solely solve all the challenges. This motivated us to combine the surrogates with some compatible algorithms. The deterministic optimization algorithms favour available equations to obtain the gradient information; but many surrogates are black-box models, so it might be tricky to employ these surrogates within deterministic optimization algorithms. By contrast, stochastic methods are independent of the model types and can be compatible with surrogates. Further, I still need to admit the advantage of deterministic algorithms in the optimality guarantee. I will not leave out deterministic algorithms at this stage, but I will keep in mind the necessity of obtaining optimality. Further, a hybrid concept is proposed to integrate the advantages of stochastic and deterministic methods.

Table 2.4. Optimization challenges and solutions.

Solutions	Function evaluation	Nonconvexity	Dimensionality	Optimality
Surrogate	+	N.A.	N.A.	N.A.
Deterministic	-	-	+ (good initials)	+
Stochastic	N.A.	+	-	-
[Surrogate + stochastic]	+	+	-	-
Hybrid	N.A.	+	+	+

2.2 Decarbonization

To limit global warming to 1.5~2 ℃ above pre-industrial levels, worldwide countries have pledged to cut the CO_2 emissions to nearly zero (or 'net zero') by the mid of 21st century [2]. This section will cover the net-zero trends and why CCU is necessary.

2.2.1 Net zero trend

Net zero requires a complete upgrading for the current energy system, since approx. 75% of GHG emissions result from today's energy sector [1]. In a broad sense, the energy sector contains the electricity generation together with the energy use in transport, industry and building sectors. The remaining 25% GHG emissions come from landfill, industrial emissions, agriculture, forestry and land use [69]. Notably, net zero will not happen spontaneously, and it largely relies on policies. So far, over 130 countries have set out net-zero pledges by 2050 or 2060 [2], but few pledges are backed up by legislation or detailed policies [1]. As shown in Figure 2.6, International Energy Agency (IEA) reports three scenarios: (1) Stated Policies Scenario (STEPS), only including the existing policies by 2021, (2) Announced Pledges Case (APC), where all the pledged targets by 2021 will become policies, and (3) net zero case. STEPS can be regarded as a business-as-usual case, projected to raise the global temperature by 2.7 ℃ in 2100. APC can reduce emissions, but it is still far away from net zero (Figure 2.6a), Transition to net zero requires drastic changes across many industries, which is challenging to realize in a short period [70]. Based on these three scenarios, some trends can be identified as follows.

(1) Electricity generation sector produces nearly 40% emissions in 2020 [1].
(2) The renewable share is plan to grow to 25 ~ 67% of global energy mix in 2050 [1].
(3) The reliance on fossil fuels will decline to 67 ~ 22% of global energy mix in 2050 but not disappear, because fossil fuels are required to produce carbon-embodied products (*e.g.*, specific polymers) [1], which cannot be easily replaced by bio-materials based products.
(4) Even if fossil fuels decline to 22%, the potential market of carbon capture is enormous because around half of fossil fuels are required to equip with carbon capture (4 Gt CO_2 captured in 2035, while 7.6 Gt captured in 2050) [1].

(5) Electrification is a trend across all sectors. Regarding the supply of worldwide heating utility, fossil fuels should gradually be substituted by low-carbon electricity [1], with a share of 20% in 2020 increasing to 49% in 2050. The corresponding electricity can be generated from renewables or power plants integrated with carbon capture.

(6) Significant growth in carbon price should be introduced to regulate emissions [1].

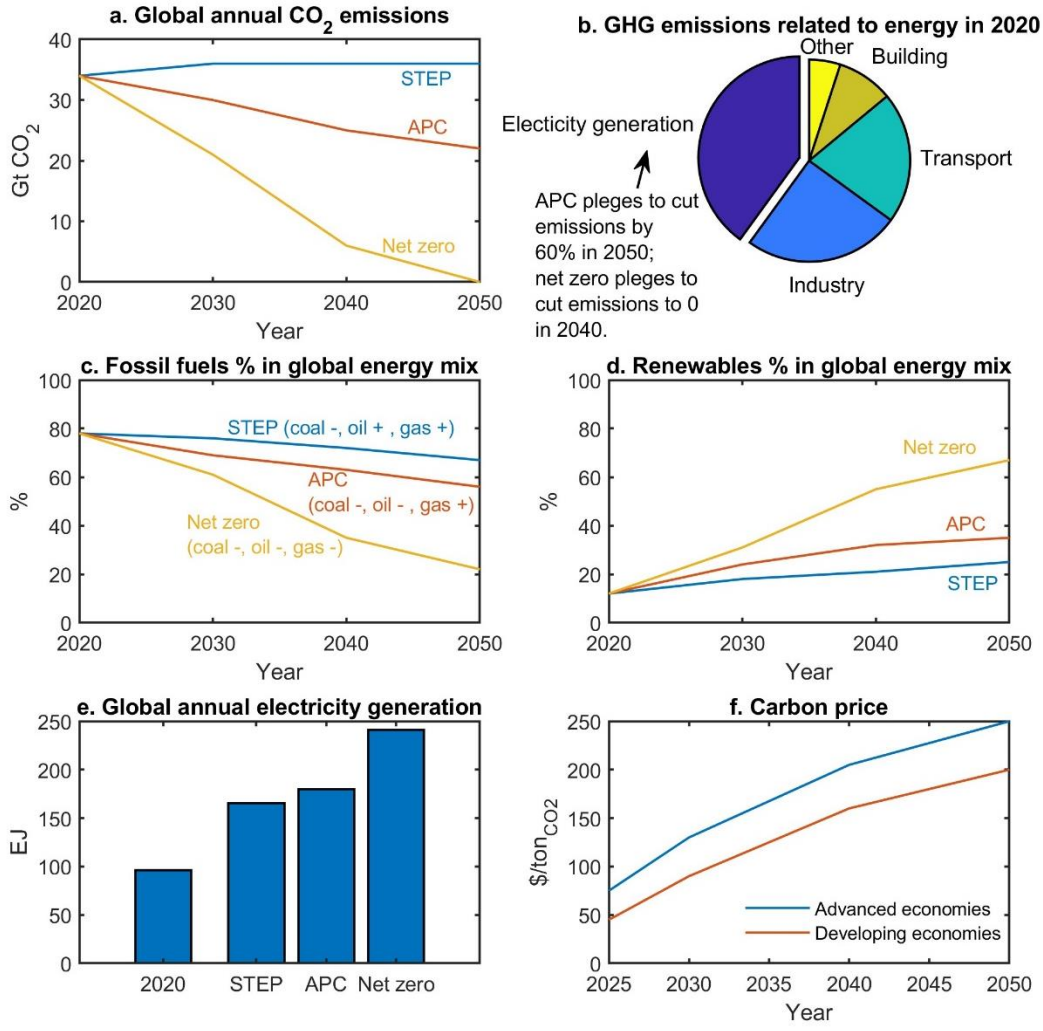


Figure 2.6. Three scenarios for energy transition, predicted by IEA [1]. STEPS only considers the existing policies, which can control the temperature increase within 2.7 ℃ in 2100; (2) APC assumes that all the pledged targets will become policies, which can control the temperature increase within 2.1 ℃ in 2100; (3) net zero case, corresponds to the temperature increase by 1.5 ℃ in 2050 [Note: this net-zero case scenario is proposed by IEA, but not the only scenario path to achieve net-zero emissions.].

2.2.2 Carbon capture

In any above-mentioned scenario, reducing the GHG emissions of power plants is paramount. Fossil fuels may still dominate the supply of the global energy, while the growth rate of renewables may be subject to a high degree of uncertainty in the long term. As such, there is a need for innovation that supports a stepwise transition from the current fossil-fuels-based energy supply to the renewable-based future. Herein, an intermediate solution may be fossil-fuel-based energy supply integrated with carbon capture for GHG emissions reduction. IEA predicts that, from 2030 onwards, every month is projected to see ten more power plants equipped with carbon capture [1]. Furthermore, BP Energy outlook reports that natural gas is more resilient to the pressure under the energy transition than oil or coal [71], so specific attention should be given to gas-fired power plants coupled with carbon capture.

In addition to the application in the electricity sector, carbon capture technology can decarbonize other high-polluting industrial sectors. Financial Times reports that oil and gas major players are facing growing pressure for energy transitions from stakeholders, and many are considering carbon capture [72]. For example, BP plans to produce low-carbon energy through CCUS; Chevron aims to capture 25 million tons CO_2/year by 2030 [72].

2.2.2.1 CCS vs. CCU

'Capture' systems described in the literature usually refer to carbon capture, utilization and storage (CCUS). Carbon capture involves capturing CO_2 from heavy industries, such as power stations, fertilizer production sites, cement factories, steel plants, or directly from the air [10]. CO_2 storage refers to the captured CO_2 being compressed and injected into the underground for permanent storage [10]. CO_2 utilization converts the captured CO_2 to valued products, *e.g.*, fuels and polymers [10]. Depending on storage or utilization, the CCUS is divided into CCS (carbon capture and storage) and CCU (carbon capture and utilization) [73].

When it comes to the relationship between CCU/CCS and environment, CCU aims at a start-of-life problem, *i.e.*, a feedstock problem with CO_2 as a sustainable carbon source, while CCS corresponds to an end-of-life problem, *i.e.*, the endpoint of CO_2 [73]. As shown in Figure 2.7, CCU routes will release CO_2 back to the atmosphere, while CCS offers the destination for CO_2 (~1000 years' storage). Due to their different functions, Bruhn *et al.* claim that CCU should be

separated from CCS, because the combination can lead to confusing conclusions in policymaking [73]. Table 2.5 summarizes the difference between CCS and CCU. CCS can reduce CO_2 emissions for fossil fuels, so oil and gas players can claim that they are producing 'clean coal/gas', but environmentalists criticize the CCS prolongs the usage of fossil fuels. By contrast, CCU reduces fossil fuels as feedstocks, while CCU has little impact regarding climate change because the CO_2-based products will eventually release the CO_2 back to the environment. As such, policymakers may underestimate the environmental benefits of CCS if using a combining term as CCUS. In a broad sense, CCS is tackling a global issue for climate mitigation, while CCU is focused on the regional energy transition, which also determine the demand in CCS is much higher than CCU in the future (2.18 Gt CO_2 for CCU, while 7Gt CO_2 for CCS in 2050) [73]. Another issue is that a combined form 'CCUS' may confuse that policymaker that CCS can gain profits. To make CCS profitable, CO_2 storage is primarily integrated with enhanced oil recovery (EOR). Otherwise, the deployment of CCS can deliver a limited financial return. By contrast, CCU can produce high-value products to deliver economic benefits. Further, the CCU is intensively involved with the chemical conversion of CO_2, which is driven by chemical sector; while fossil-based power plants can be equipped with CCS, which is the interest of electricity sector under the net-zero trend [73].

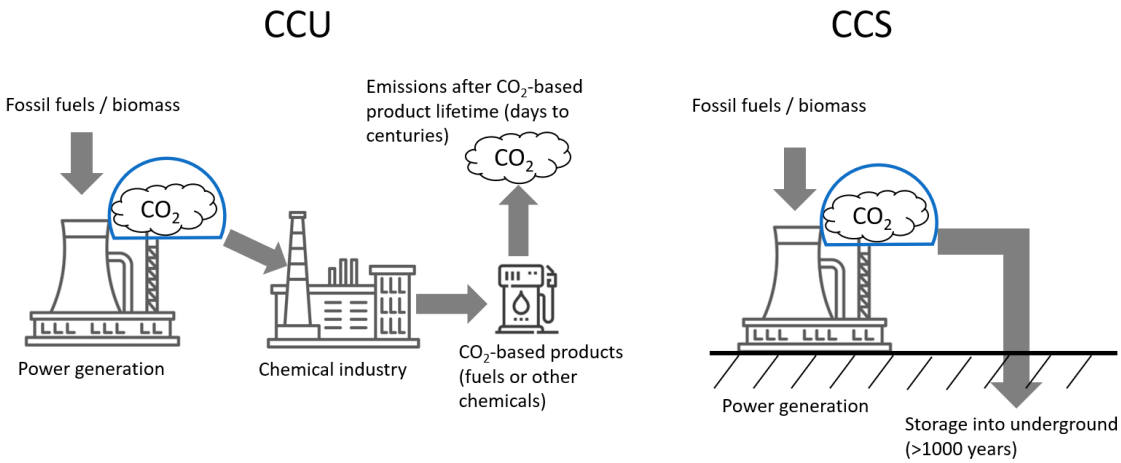


Figure 2.7. Carbon chain for CCU *vs.* CCS: CCU releases the carbon back to atmosphere, while CCS can store the carbon into the underground. CCU can convert the captured carbon to valued products, so the demand for fossil fuels can be reduced. Adapted from [73].

Table 2.5. Differences between CCU and CCS [73].

	CCU	CCS
Motivation	Fossil fuels substitution	Climate change
Driving force	Chemical sector	Electricity sector
Technology	Chemical conversion	Storage
CO_2 storage duration	Fuels (days to weeks); Polymers (years)	~ 1000 years
Sustainability	Circular economy – CO_2 recycling	Combining with bio-energy can achieve negative emissions
Economic aspects (CCU>CCS)	Potential profits	Negative profits in most cases
Climate protection (CCU<CCS)	CO_2 will release back to the atmosphere	CO_2 can be stored underground for ~ 1000 years.
Energy transition (CCU>CCS)	Substitute fossil fuels for carbon sources	Prolong the usage of fossil fuels (because it claims to make ‘clean coal/gas’)
Annual demand in 2050	2 Gt CO_2 for fuels; 0.18 Gt CO_2 for polymers	7 Gt CO_2
Problem scale	Reginal / national level	International / world level

Once the direct air capture (DAC) technology is mature, CCU can be followed by CCS, thus achieving the start-to-end of the carbon chain. However, the concentration of CO_2 in the air is 400 ppm, requiring a considerable amount of energy to concentrate this dilute concentration.

The availability to the cheap renewable energy sources is the key to commercialize DAC [74]. For example, the access to abundant geothermal energy makes it possible for Climeworks to run a DAC plant in Iceland, but such an advantage of renewable energy sources is probably not applicable to other regions. Therefore, DAC is still a long way to commercialization. Herein, this thesis treats CCS and CCU separately.

CCU has several intrinsic advantages over CCS. Most studies assume that the global storage space is unlimited [75]. Also, long-term storage means over thousands of years. Because the space of earth is limited, storage price will dramatically increase when it reaches the storage capacity. When the storage room is shrinking, the storage price will surge to sky-high. CCS is more like the existing linear economy ('take → use → waste'). CO_2 is 'disposed' to the underground. If the circular economy is applied to carbon management, carbon is required to be recycled and reused as starting materials to manufacture valued products – just like CCU. Similarly, in waste management, recycling/reuse is better than landfill. The challenge for CCU falls in that extensive energy is required to activate CO_2. To reduce the emissions by the energy use, renewable electricity is commonly considered in the design of CCU systems [27, 31, 33]. When the surplus renewable energy is supplied to the CCU and CO_2 can be captured from the air, CCU can work as an energy conversion system and potentially a negative-emission technology [76]. Further, CCS has little potential for making profits and requires a considerable investment in the pipeline infrastructure, so it will depend heavily on governmental subsidization [73]. By contrast, CCU regards CO_2 as a carbon source for further conversion, which can be highly profitable [73]. Based on the above-mentioned factors, CCU can be a more promising solution than CCS, so this thesis will focus on CCU.

2.3 Process options for carbon capture and utilization

2.3.1 Carbon capture

CO_2 capture can be achieved by various technologies, ranging from chemisorption, adsorption, membrane to chemical looping combustion, *etc.* [10]. This section will introduce the most mature technology - Chemisorption technology, but this technology depends on the use of amines, *e.g.*, MEA, which might limit its deployment under specific regulations. Here, I also

introduce a non-reactive technology – pressure swing adsorption, based on the physical interaction rather than a chemical reaction.

2.3.1.1 Chemisorption process

In a chemisorption process, aqueous amine solvents are used to absorb the CO_2. Amine absorbents are inexpensive and abundant [10]. Monoethanolamine (MEA) is a typical absorbent, which can achieve over 90% CO_2 removal in the post-combustion capture [10]. MEA process has been commercialized regarding CO_2 capture for coal-based power plants in Canada and US [10]. The first industrial-scale MEA process started its operation in 2014 in SaskPower's Boundary Dam power plant, Canada [77]. The Dam unit (power-CCS) generates 115 MW of low-carbon electricity, which can supply around 1 million households while reducing the SO_2 emissions by 100% and CO_2 emissions by 90% compared to a direct-emission power plant [77].

Figure 2.8 presents the flow diagram of a typical chemisorption process, which contains two main parts – absorber and stripper. The CO_2 sources (*e.g.*, flue gas, containing 4 ~ 20% CO_2) are compressed into the absorber. Within the absorber, CO_2 is captured by the amine solution with fast absorption kinetics (RX 2.1) to form the CO_2-rich stream, whereas the treated gas leaves at the top of the absorber. Following this, the CO_2-rich solvent is heated in the stripper, where CO_2 is separated from the amine solution while amine solvent is regenerated (RX 2.2). Subsequently, the regenerated amine solvent is recycled back to the absorber. A small amount of amine components is lost due to thermal degradation in the stripper, so amine make-up is required to add into the process [78, 79]. The heat usage in the stripper is the hotspot of energy consumption [78, 79]. Typical heat integration employs a heat exchange network, where the outflow of the stripper can be used to pre-heat the inlet flow of the stripper.

Absorber: $$CO_2 + 2RNH_2 \rightarrow RNHCOO^- + RNH_3^+$$ RX 2.1

Stripper: $$RNHCOO^- + RNH_3^+ + (heat) \rightarrow CO_2 + 2RNH_2$$ RX 2.2

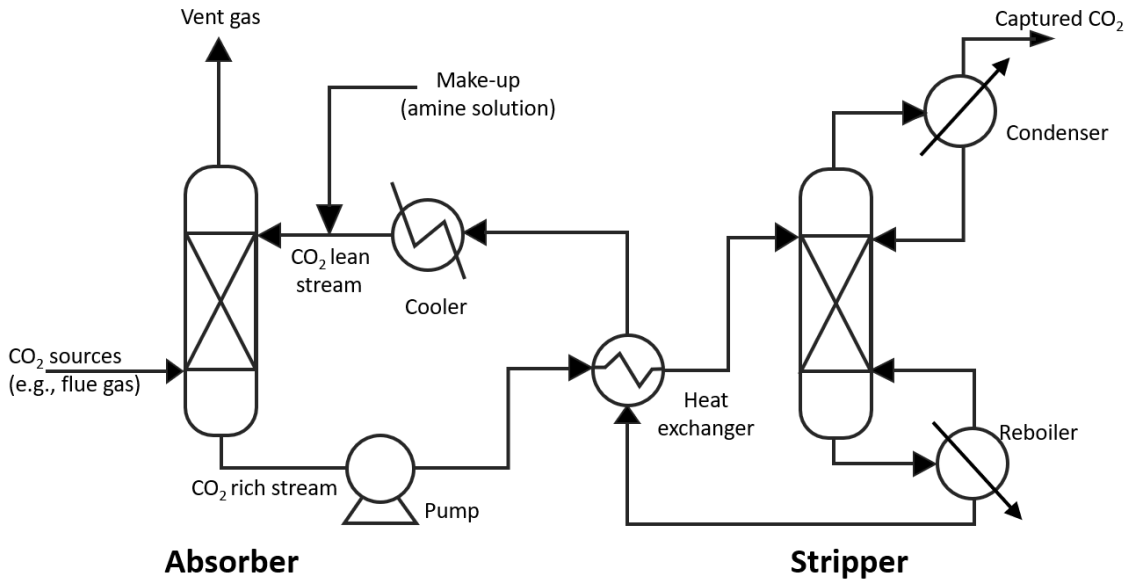


Figure 2.8. Process flow diagram of the amine-based CO_2 capture process. Adapted from [26].

The simulation for the chemisorption process is not an easy task because the convergence needs to close the recycle streams and satisfy the specification for the make-up stream (the system is highly nonlinear). Some relevant works can be referred to [78, 79]. To release the burden of nonlinearity of the Aspen Plus model, Chung *et al.* established a surrogate model to replace the original process model [26].

2.3.1.2 Pressure swing adsorption process

Pressure swing adsorption (PSA) is an energy-efficient gas separation technology [80-82] that has been widely used in the industry for drying [83], air separation [84, 85], and hydrogen production [86, 87]. Over the last two decades, academia has grown interested in applying PSA for CO_2 capture [88, 89]. PSA possesses significant advantages over the conventional amine-based CO_2 capture technology regarding emissions to the environment and energy consumption [23, 82]. Since no amine solvent is involved in the PSA system, no organic waste is disposed to the environment.

Pressure swing adsorption (PSA) is a cyclic dynamic process for gas separation. Through continuously varying pressure, adsorption switches with desorption for all the process periods. Notably, discontinuities are introduced by a sequence of frequent control actions of pressure

levels, thus resulting in multiple discrete stages, *e.g.,* adsorption, blowdown, evacuation, and feed pressurization, while each stage is operated continuously. Hence, the overall process belongs to a class of *combined discrete/continuous* systems, which require additional effort in the model formulation and numerical solution [90]. Eventually, PSA reaches a cyclic steady state (CSS), where consecutive cycles have the same profile. More details and the process models can be found in Chapter 3.

Existing fossil-fired power plants can be equipped with some capture technologies, such as the chemisorption process and PSA. Meanwhile, most of carbon capture technologies require energy input, which then causes the energy penalty for power plants [91].

2.3.2 Carbon utilization

CO_2 utilization has gained increasing attention in the past 20 years. CO_2 is an alternative carbon source, which can substitute fossil fuels. However, the challenge is that CO_2 is a thermodynamically stable molecule and is hard to activate in a chemical reaction. In the current industry, CO_2 is used either for urea production or in the other application fields, where the chemical form of CO_2 does not change, ranging from carbonated soft drinks, dry ice, inert gas, solvent (*e.g.,* supercritical fluid). Yet, the ambition of fighting climate change has driven the breakthrough in catalysis for CO_2 activation [92-94], and the CO_2-based products have expanded to fuels, polymer, concrete, and carbon monoxide [10].

2.3.2.1 CO_2 to synthetic fuels

In all the utilization pathways, CO_2-to-fuels is predicted to accounts for a considerable percentage ~50% [95]. CO_2 is considered to partially substitute fossil fuels to produce liquid fuels, such as gasoline/diesel (through Fischer-Tropsch) and methanol (further processed to DME) [14]. Additionally, CO_2 is considered a feedstock to produce gas fuels, *e.g.*, methane [96]. Liquid fuels have more advantages over gas fuels regarding high energy density, available infrastructure, and low-cost storage/transportation. Hence, priority should be given to developing liquid fuels in the short future.

Thermodynamically, CO_2 is too stable to be easily converted to high-value chemicals. To produce fuels, steam/NG/CO_2 can be converted to an intermediate – syngas by reforming

approaches, generally in high temperature or with other energy input (electricity, plasma). The ratio of H_2 over CO in the syngas is directly related to which type of fuel to produce. One advantage of using syngas as an intermediate is that this ratio can be easily adjusted by varying the ratio of steam/NG/CO_2 or separation. Following this, syngas is converted to fuel products *via* Fischer-Tropsch and methanol synthesis (Figure 2.9).

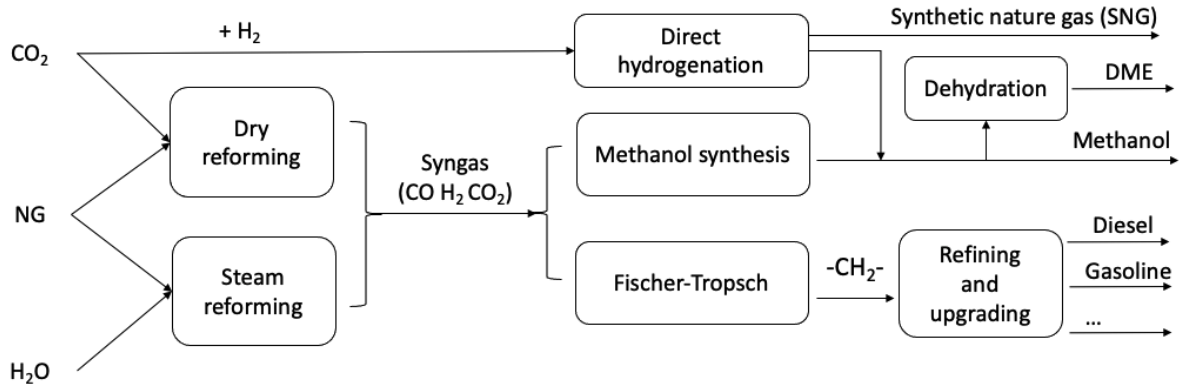


Figure 2.9. CO_2 utilization pathways to fuels.

The reforming approaches (syngas production) range from partial oxidation reforming, autothermal reforming, steam reforming to dry reforming [97]. However, none of the individual reforming technology can provide the desired syngas ratio for the synthesis of fuels. Thus, separation technology is required to remove the excessive component. Alternatively, these reforming options can be combined to adjust the ratio between H_2 and CO. Baltrusaitis *et al.* compared five combination configurations and delivered the conclusion that the combination of steam reforming (NG + steam) and dry reforming (NG + CO_2) is economically favoured over other options (RX 2.3 - RX 2.4) [98]. Also, such a combination can reduce carbon footprint by 67% over conventional steam reforming [99]. Meanwhile, the existence of steam is preferred because the dry reforming suffers severe coke formation [100]. Another novelty of combined reforming (NG + CO_2 + steam) is that CO_2 is regarded as one of the raw materials. Thus, unlike the conventional process (autothermal reforming or steam reforming), a CO_2 separation unit is not required before the reforming unit in the process design. Based on the thermodynamic analysis, the combined reforming has the potential to fully convert the methane while maintaining a high yield of H_2 at 88% [101].

$$CH_4 + H_2O \leftrightarrow 3H_2 + CO \qquad \Delta_r H = 206\ kJ/mol \qquad \text{RX 2.3}$$

$$CH_4 + CO_2 \leftrightarrow 2H_2 + 2CO \qquad \Delta_r H = 247\ kJ/mol \qquad \text{RX 2.4}$$

Fischer-Tropsch (FT) is the most important route for utilizing syngas because it can generate the essential fuels – gasoline and diesel - for the existing transport sector. FT has a long industrial history of producing high-quality fuels (S-free and N-free) [102, 103]. FT can be classified into two types: low-temperature FT (LTFT) based on cobalt catalysts and high-temperature FT (HTFT) based on iron catalysts. The FT reaction is highly exothermic, and a lower temperature can improve the final conversion regarding the thermodynamic equilibrium. By contrast, high temperatures terminate the carbon chains, resulting in shorter hydrocarbons [104]. In recent years, LTFT has been more popular because it can produce long-chain hydrocarbons and is less energy-intensive [103]. Additionally, recent research works focus on the cobalt catalysts [103, 105-109], which are more active and widely used in industry. Hence, priority will be given to LTFT in this work. CO_2 is reported not to react on the cobalt-based catalyst and can be regarded as an inert gas in the FT reaction [108, 110]. Thus, no reaction associated with CO_2 is listed in the FT section. The desired syngas ratio falls in the range of 2 – 2.2 for H_2 : CO [102, 103, 109], and the reaction temperature is reported to range from 200 to 270 ℃ [29, 102, 111] as well as pressure in 15 – 50 bar [29, 111, 112].

$$nCO + (2n+1)H_2 \rightarrow C_nH_{2n+2} + nH_2O \qquad \Delta_r H = \sim -170\ kJ/(mol\ CO) \qquad \text{RX 2.5}$$

Methanol synthesis has attracted increasing attention in academia and industry. Methanol can serve as a component in the new type of gasoline, and methanol can work as an intermediate for further processing to dimethyl ether (DME), which can be used in DME fuel cells and diesel engines [103]. Besides the application in fuel production, methanol can be converted to olefins, which is more economically favoured [103]. Methanol synthesis (*via* either syngas or direct hydrogenation) is an exothermic reaction. A low reaction temperature is desirable for a high conversion but results in a slow rate. Hence, the industry initially employed a high pressure ranging from 50 to 100 bar to enhance the reaction rate and shift the equilibrium to the right. Later, new reactor configurations were developed to remove the heat. For instance, Imperial Chemical Industries (ICI) developed a quench reactor, allowing lower operating pressure. A

quench reactor supplied the syngas in a distributed way by feeding cold unreacted syngas along the catalyst bed, as to lower reaction temperature for higher conversion. Another method for heat removal is applying a multi-tubular reactor, where parallel tubes are installed in a column filled with cooling water. Additionally, the existence of CO_2 can also help produce methanol. The optimal syngas ratio is set as $(2CO + 3CO_2): H_2 = 1:1$. To avoid the carbon loss in the intermediate step, the syngas step might be skipped and there is a research trend for the direct CO_2 hydrogenation to methanol (RX 2.7), as indicated in excessive research works in catalyst development [92-94] and process design [14, 27, 113]. However, these works assume that H_2 can be generated from renewable electricity, whereas it is still far from commercialization due to the limited capacity and intermittent availability of renewables.

$$CO + 2H_2 \leftrightarrow CH_3OH \qquad \Delta_r H = -91.0\ \text{kJ/mol} \qquad \text{RX 2.6}$$

$$CO_2 + 3H_2 \leftrightarrow CH_3OH + H_2O \qquad \Delta_r H = -49.2\ \text{kJ/mol} \qquad \text{RX 2.7}$$

$$2CH_3OH \leftrightarrow CH_3OCH_3 + H_2O \qquad \Delta_r H = -23.4\ \text{kJ/mol} \qquad \text{RX 2.8}$$

2.3.2.2 CO_2-to-chemicals

CO_2 can be converted to ‘chemicals’. The phrase ‘chemicals’ in this specifically section refers to the carbon-based solid materials, which can keep carbon inside ‘chemicals’ for a long period (~ 100 years for polymers and ~ 1000 years for metal carbonates). As fuels release carbon back to the atmosphere in a short time (carbon storage from days to weeks), fuels are not considered as ‘chemicals’ in this section. Assen *et al*. use CO_2 to partially substitute fossil fuels to produce polyurethane, reducing ~ 15% fossil fuels exploitation and ~ 15% GHG emissions [114]. Some studies related to mineral carbonation look promising because these routes are thermodynamically favoured, where carbon keeps the +4 oxidation state [115]. Bui *et al*. reported an inspiring business story for a CO_2-to-chemical start-up company: Carbon8 converts steel waste together with CO_2 to construction materials, and it gets two incomes: selling the construction materials and charging the waste suppliers, who need otherwise pay more for landfill [10]. CO_2-to-chemicals will not be covered in this thesis work, and more details can be referred from the review paper of Bui *et al.* [10].

Different pathways are hard to compare. Ruud *et al.* criticized that some works did not define clear system boundaries, and thus it is still controversial to state which utilization pathway is most viable [103]. Burre *et al.* has developed a systematic way to compare different process configurations for producing dimethoxymethane regarding system boundaries and multiple deciding criteria [31], and their evaluation methods may be borrowed in the evaluation of different CCU pathways.

2.4 Process evaluation methods

There are various pathways for the capture step and utilization step, respectively. To make the right decisions among the multiple pathways, Zimmermann *et al.* have developed a systematic methodology of the CCU evaluation: two evaluation methods are considered: techno-economic assessment (TEA) and lifecycle assessment (LCA) [76]. TEA evaluates how competitive the technology is in the current market, while LCA accounts for the environmental impacts of a product through its entire life cycle. A good process solution should balance economic and environmental aspects [76].

2.4.1 Life cycle assessment

Life cycle assessment (LCA) evaluates the environmental impacts of a product throughout its entire life cycle stages ranging from raw material, manufacturing, distributing, use, to recycling/disposal. In each stage, the product system can interact with the environment by consuming raw materials and utilities as well as emitting wastes [76]. Such interactions of each stage will be quantitively assessed by LCA. LCA can be directly applied in optimizing product manufacturing, policy-making, and marketing [76].

The framework of LCA follows systematic steps: goal and scope definition, inventory analysis, and impact assessment.

Goal and Scope: **Goal** tends to specify the application of the study. Normally, LCA is used in comparative assessments. Specifically for CCU, the common goal is (1) to compare a CCU-based process with a conventional process derived from fossil fuels; (2) to identify the hotspots for improvement on the reduction of emissions; (3) to compare different CCU pathways. Most LCA studies compare the CCU with a reference process. **Scope** defines the system to be

investigated, its functions and system boundaries, *etc*. A CCU system can achieve multiple functions, such as the co-production of various products (fuels, polymers) and generating low-carbon power. For a proper comparison, the reference process also needs to achieve the equivalent functions. System boundaries define the life cycle stages required to compare different processes. Evaluating the entire life cycle of a product (cradle-to-grave) requires excessive workloads, because various process options exist and relevant data is required for each life cycle stage. The manufacturing stages of different CCU pathways generate significantly different emissions, while the downstream emissions are identical [76]. As such, it is not necessary to assess the downstream emissions. As shown in Figure 2.10, a cradle-to-gate approach will quantify the GHG emissions of raw materials and utility and CO_2 emissions for a CCU system (*e.g.*, uncaptured CO_2 and CO_2 in purge streams). Hence, the cradle-to-gate is sufficient to perform a comparative assessment.

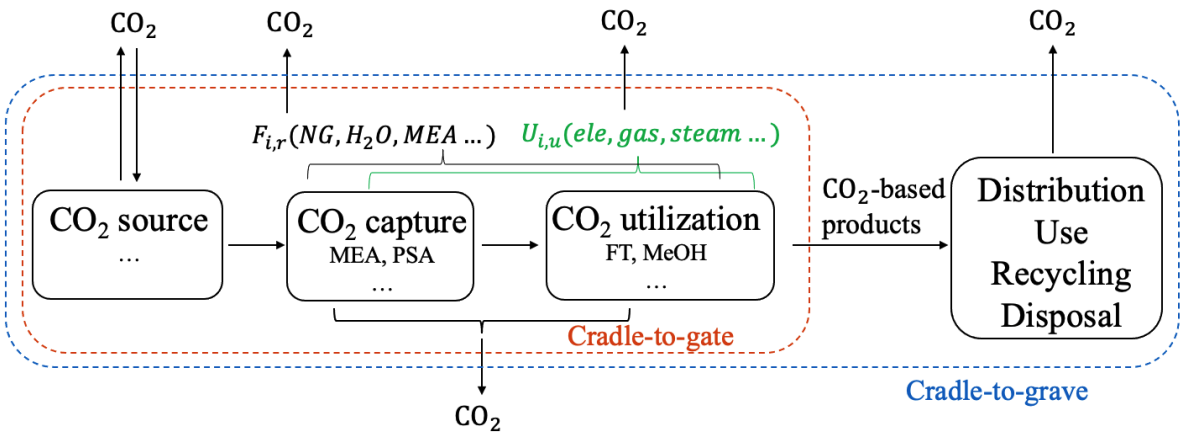


Figure 2.10. System boundaries of cradle-to-gate *vs.* cradle-to-grave approach for the emissions of a CCU system.

Inventory analysis: based on the goal and scope, this step aims at data collection and the description of the production system in a flow chart. Data refers to the potential environmental impacts of material/utility consumption, which can be sampled in a relevant database (*e.g.*, GaBi database) or estimated by some methods based on stoichiometric relationships and thermodynamic methods [76]. The flow chart is required to represent the system boundaries and the mass/energy balances.

Impact assessment: based on the inventory analysis, these steps quantify the potential environmental impacts of the mass/energy flows. A system can have numerous mass/energy flows, thus leading to difficulty in evaluating the systems. By contrast, the impact assessment delivers an overall result, which can be easily used to compare different pathways [76].

2.4.1.1 Process expansion for multi-functional comparison

The uncertainty of life cycle analysis is unavoidable due to the lack of data and improper assumptions. To minimize these uncertainties, the difference between the assessed system and a reference system is normally quantified during LCA. Most CCU systems can achieve multiple functions, *e.g.,* power generation and production of certain CO_2-based products. For a fair comparison, Wunderlich *et al.* reported several methods to solve multifunctionality: sub-division, system expansion substitution and allocation [76]. To the best of my knowledge, the system expansion is the most straightforward way to compare a CCU system with a reference system. The system expansion requires a reference system to include all the functions as the CCU system. As shown in Figure 2.11, a reference system refers to a conventional process, which contains the same CO_2 source (but no capture technology) and conventional technology to produce the CO_2-based product. Meanwhile, the main product and CO_2-based product are supposed to be equivalent in the two systems for a reasonable comparison.

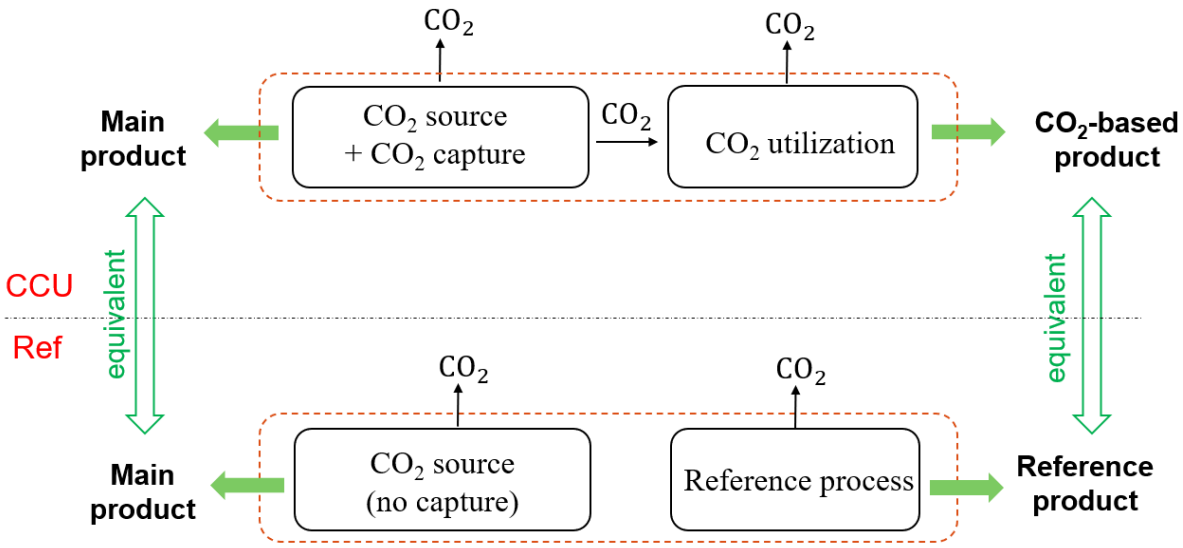


Figure 2.11. System expansion for the comparison between CCU and the reference process. Adapted from [76].

2.4.2 LCA-Economic assessment

The combination of LCA and economic assessment can support the decision-making on the technology selection and optimal operating conditions regarding environmental and economic aspects. These aspects can commonly conflict with each other. To solve the conflicting issue, multi-objective optimization can be applied to identify the trade-off between two criteria [62]. Such trade-offs may gradually change with time. Economic assessment is subject to market dynamics, while LCA is concerned with the long-run environmental effect.

2.5 Simulation tools

Numerous simulators have been developed in academia and industry as to accelerate the process design and development (several examples listed in Table 2.6). These simulators can deliver the mass and energy balances of process systems, which are directly concerned with process design, control and optimization.

Table 2.6. A list of common chemical process simulators.

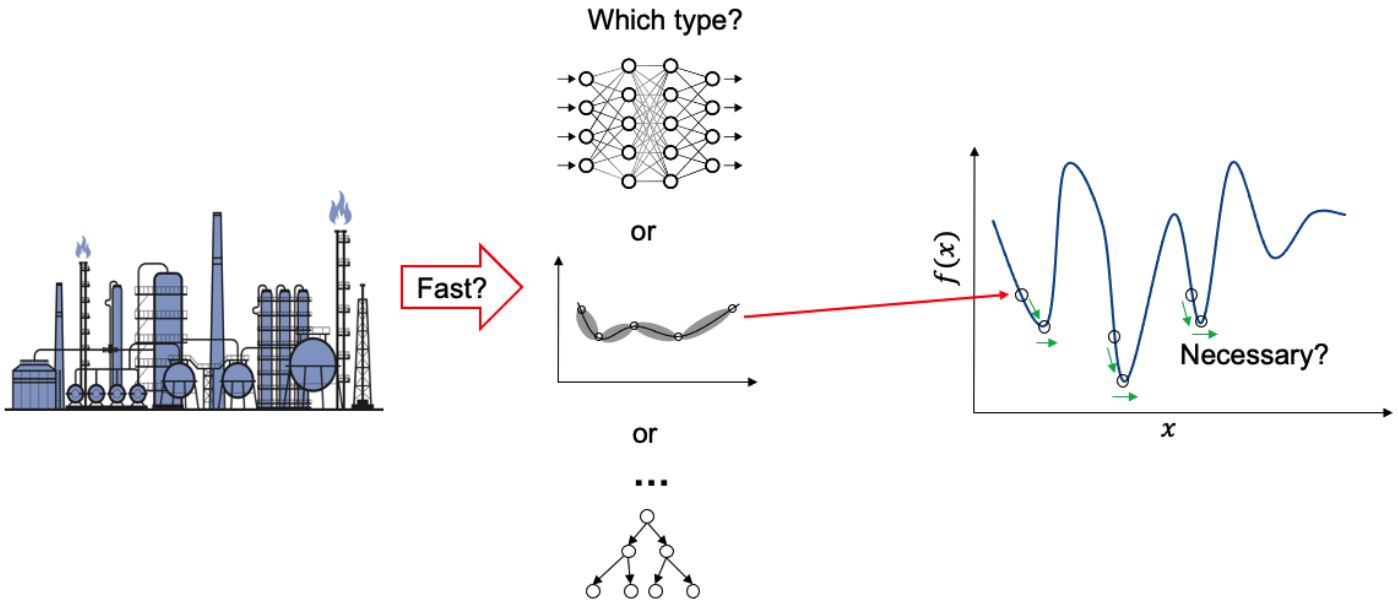
Software	Developer	Application	Type
Aspen Plus [116]	Aspen Technology	Steady-state simulation Optimization	Sequential-modular
Dymola [117]	Dassault Systèmes	Modelling Dynamic simulation	Equation-oriented
gPROMS [118]	Siemens	Modelling Dynamic simulation Optimization	Equation-oriented
OpenModelica [119]	Modelica Consortium	Modelling Dynamic simulation	Equation-oriented
PRO/II [120]	AVEVA	Steady-state simulation	Sequential-modular

The development of process simulators greatly facilitates engineers to establish a flowsheet for process evaluation. However, many of those software platforms have limitations in modelling and optimization. For modelling, the equations behind the unit operations are hidden in many commercial platforms, such as Aspen Plus, gPROMS ProcessBuilder. The good thing is that many software platforms offer users to edit their models for their novel units in a custom way, such as Aspen Custom Modeler, gPROMS ModelBuilder, and Dymola. Regarding optimization, most process simulators have limited choices in optimization algorithms.

A CCU system can contain several sub-systems (several capture processes and a couple of utilization pathways to respond to market demands), thus resulting in a large multi-process system. The process models of sub-systems may be established in different platforms, whereas for overall optimization, they ideally should be accessible from a single platform. Therefore, it is necessary to establish interfaces between process simulators and a high-level platform (*e.g.*, MATLAB, Python, C++). MATLAB can access various optimization algorithms and machine learning models. The interfaces of MATLAB to process simulators were previously reported in the optimization [24, 77, 121] and control [122] for the sub-systems of CCU. Hence, MATLAB is selected as the high-level interactive platform in this thesis.

Part I: Methodology development

Part I contains Chapters 3 ~ 5. Chapter 3 examines whether optimality check is necessary or not. To build the surrogate-based optimization framework, Chapter 4 explores which surrogate types are suitable for CCU process options, and Chapter 5 develops the methodology in improving the efficiency of surrogates' generation.



(Part of image elements come from VectorStock with the standard license EULA https://www.vectorstock.com/faq/member/standard-license)

Chapter 3 Optimality check: hybrid optimization of pressure swing adsorption

[Contribution declaration: the work in Chapter 3 is in collaboration with the PSE group in RWTH Aachen University. Dr. Schweidtmann proposed the hybrid optimization strategy. Dr. Caspari supported me in the implementation of dynamic optimization in DyOS. The DyOS package was supplied by the PSE group of RWTH Aachen University. I implemented the rest of work in Chapter 3.]

This chapter initially aims to establish an efficient optimization framework to tackle one of the most challenging carbon capture process options – pressure swing adsorption (PSA). Meanwhile, this chapter tests whether the stochastic surrogate-based optimization is far from the optimal solution. To check this, a hybrid two-step optimization framework (TSEMO + DyOS) is proposed. In the first step, a Bayesian stochastic multi-objective optimization algorithm (*e.g.*, TSEMO) searches the entire decision space and identifies an approximated Pareto front within a small number of simulations. Within TSEMO, Gaussian process (GP) surrogates are trained to approximate the original process models. In the second step, a gradient-based deterministic algorithm (*e.g.*, DyOS) is initialized at the approximated Pareto front to further refine the solutions until local optimality. Therein, the rigorous process model is used in the optimization. As a result, the improvement in the second step can be negligible compared to the first step, which means that the solution by stochastic surrogate-based optimization is good enough (very close to optimality). Additionally, the proposed hybrid framework is efficient, because it benefits from the coarse-to-fine function evaluations and stochastic-to-deterministic searching strategy. When the result is far from the optima, TSEMO can efficiently approximate a trade-off curve as good as a commonly used evolutional algorithm, *i.e.*, Nondominated Sorting Genetic Algorithm II (NSGA-II). This is because the GP-based surrogate model is utilized for function evaluations in the initial coarse search. When

the result is near the optima, the searching efficiency of TSEMO dramatically decreases, while DyOS can accelerate the searching efficiency by over 10 times. This is because, in the proximity of optima, the exploitation capacity of DyOS is significantly higher than that of TSEMO.

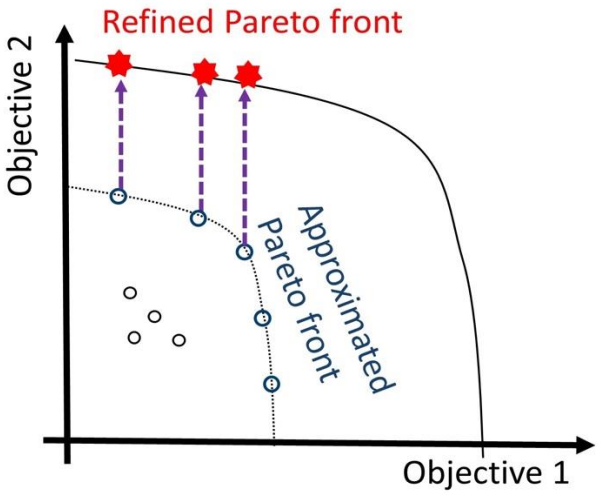


Figure 3.1. The concept of hybrid optimization framework: a stochastic multi-objective optimization algorithm is used to search the decision space globally and approximate Pareto front points, which are locally refined by a deterministic algorithm.

Nomenclature in Chapter 3

P	Pressure
P_H	Setpoint for high pressure
P_I	Setpoint for blowdown pressure
P_L	Setpoint for evacuation pressure
T	Temperature
t	Time
z	Bed coordinate
v	Interstitial velocity
R	Gas constant
ε	Bed voidage
q	Concentration in the solid phase

y	Composition in the gas phase
k	Mass transfer Coefficient
ρ_s	Density of adsorbent
$C_{P,a}$	Specific heat capacity of the adsorbed phase
$C_{P,s}$	Specific heat capacity of the adsorbent
$C_{P,g}$	Specific heat capacity of the gas phase
K_z	Effective gas thermal conductivity
H	Enthalpy
T_w	The temperature of the column wall
T_a	Ambient temperature
ρ_w	The density of column wall
$C_{p,w}$	Specific heat capacity of the column wall
K_w	Thermal conductivity of the column wall
r_{in}	Column inner radius
r_{out}	Column outer radius
h_{in}	Heat transfer coefficient inside the column
h_{out}	Heat transfer coefficient outside the column
μ	Fluid viscosity
c_i	Fluid phase concentration
U	Internal Energy
δ	a small value for checking cyclic steady state
Subscripts	
i	Index for component
ads	Adsorption stage
bd	Blowdown stage
$evac$	Evacuation stage
$press$	Feed pressurization stage

3.1 Introduction

The optimal operation of PSA processes is a challenging task due to the inherent cyclic and dynamic behaviour of the system and highly nonlinear process models [123]. Since the column pressure varies over time, the PSA process can never reach a steady-state operating point. Instead, it eventually comes to a cyclic steady state (CSS), where the trajectories of state variables (*e.g.*, temperature, gas velocity, molar fractions of components) are the same for consecutive cycles. From an industrial operation perspective, PSA is required to operate at CSS as to achieve a constant process performance. However, it is difficult to analytically calculate CSS, which generally requires a numerical simulation [124-126]. Additionally, multiple (conflicting) objectives co-exist, including product purity, recovery rate, energy consumption, and operating cost [18, 23, 127]. The process design and operation problems often involve nonconvex functions [4-6], where multiple local optimal solutions exist. Further, PSA may be operated in more complicated modes, *e.g.*, multiple columns integrated with recycles [18, 23, 82, 123]. Overall, the above-mentioned factors contribute to the difficulty in optimizing PSA processes.

In the previous literature, stochastic optimization algorithms have been used to optimize PSA processes [23, 127, 128]. Stochastic optimization algorithms consider the simulation as a black-box function. They vary the values of decision variables and run the PSA simulation until CSS. Following this procedure, the values of objectives and constraints are returned to the optimizer for evaluation. Haghpanah *et al.* used a genetic algorithm (GA) to optimize the PSA operation, while the time-consuming feature of PSA simulation leads to the slow performance of the overall optimization [23]. Capra *et al.* [127] reported a multi-level coordinate search (MCS) algorithm, where the decision space is divided for parallel computing on multiple workers to speed up the overall optimization. Stochastic algorithms can search the decision space globally. However, optimality cannot be guaranteed in finite time [3].

Deterministic algorithms belong to another type of method that can be used for PSA optimization, where gradient information is used to guide the search direction (thus, it is often referred to ‘gradient-based optimization’). There are two common approaches for the gradient-based optimization of dynamic systems, *i.e.*, the simultaneous and the sequential approaches [63]. The simultaneous approach discretizes the state and decision variables. Herein, both

temporal and spatial domains of partial differential equations (PDEs) are discretized, resulting in a large set of algebraic equations and eventually large-scale nonlinear programming (NLP) problems. Tsay *et al*. proposed a pseudo-transient optimization framework to identify the final cycle of PSA under CSS using a 'tear' method, which is similar to tearing loops and iteratively converging loops in process flowsheeting [129]. In Tsay's method, CSS can be approximated in a much faster way than the direct simulation from the initial condition, so the temporal domain is significantly reduced [129]. The sequential approach is well-suited to problems with a few decision variables and complex dynamic behaviour. The integrator solves the differential equations and provides the gradient to the NLP solver. However, in the case of PSA, a significant amount of computational time is required to calculate the sensitivity information and its integration over many PSA cycles for the gradient. Additionally, the sensitivity integration may fail due to the highly nonlinear PSA model [124]. Jiang *et al.* focused on one PSA cycle $[t_0\ t_{end}]$ and applied the sequential approach to converge the initial conditions (t_0) to the endpoint (t_{end}) of state variables [124]. This concept can dramatically accelerate the simulation to reach CSS. However, the spatial-discretized PSA model contains over numerous state variables (because each discretized cell has a set of pressure, temperature, compositions; 30 or more is recommended for the number of discretized cells [23]), and thus the convergency of them is still a large optimization problem.

Besides the extensive work on applying various optimization algorithms to PSA, researchers have exerted effort on developing surrogate models to represent the dynamic behaviour of PSA. Surrogate models are cheap-to-evaluate and can approximate the relationship between inputs and outputs of physical models. Jiang *et al.* employed a Lagrange interpolation polynomial to approximate the profiles of state variables, as to simplify the convergence problem. Nevertheless, such approximation was reported to introduce inaccuracy for the further optimal design of PSA process [124]. Agarwal *et al.* demonstrated that proper orthogonal decomposition (POD) could be employed to replace the stiff PDEs of PSA. A POD surrogate is reported to achieve a significant reduction of state variables and thus lead to low-order surrogate models [130]. Nevertheless, the error between the original PDEs and POD surrogate can be accumulated during the simulation of endless cycles of PSA, and the simulation result based on surrogate is not meaningful if the error is unacceptable.

With the recent increasing attention to machine learning, Artificial Neural Networks (ANNs) and Gaussian processes (GP) surrogate models have become prominent options for replacing computationally expensive models [53, 55, 131]. Subraveti *et al.* applied the ANN-based surrogate model to represent the original model, which was coupled with nondominated sorting genetic algorithm II (NSGA-II) for multi-objective optimization. The CPU time was reported to be 10 times shorter compared to NSGA-II coupled with the original PSA model [18]. Leperi *et al*. employed individual ANN-based surrogate models to represent typical PSA stages. Then, these surrogate-based PSA stages can synthesize different types of cycles (three-stage, four-stage or five-stage cycle) [128]. Boukouvala *et al*. applied a grey-box method to capture both the analytical information of the physical models and noise information by a GP-based surrogate model [19]. With this method, PSA processes with different materials were optimized successfully within acceptable computational time [19]. However, surrogate models are often criticized for their inaccuracy and lack of generalization [132].

In summary, prior studies on PSA optimization are based on (1) stochastic algorithms using expensive full-order models, in which optimality cannot be guaranteed, (2) deterministic algorithms which require the expensive-to-obtain gradient information, or (3) surrogate formulations in which accuracy might be compromised. A hybrid method may integrate the complementary advantages of the individual methods. The concept of hybrid optimization methods – a synthesis of a global solver with a local solver – was proposed initially by computer scientists to solve nonconvex problems many years ago [66-68]. Similarly, a concept of 'coarse-to-fine' search also proposes to transform the original problem into a coarse approximation for the initial search and then gradually approach the actual problem for refined search [133]. The efficiency of these concepts has been proven in the areas of computer vision [133], speech signal processing [134], and image processing [135].

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

The efficiency of the hybrid optimization framework benefits from:

- the stochastic-to-deterministic search strategy;

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

The remainder of Chapter 3 is structured as follows: Section 3.2 briefly describes the process model of PSA; Section 3.3 introduces the state-of-the-art algorithms used in the hybrid framework; Section 3.4 presents the optimization formulation of PSA using a hybrid optimization framework; Section 3.5 shows results, followed by the discussion on why the overall optimization efficiency of the hybrid framework is competitive in Section 3.6; the final section presents conclusions and outlook.

3.2 Model description of pressure swing adsorption

PSA is operated in a cyclic mode that alternates between adsorbing the desired gas species at a higher pressure and releasing them at a lower pressure. As shown in Figure 3.2, this thesis mainly focuses on a four-stage pressure swing adsorption (PSA) process model for CO_2 capture, which consists of four stages: (1) adsorption stage: the gas mixture flows into the column at the high pressure, P_H; (2) blowdown stage: the undesired gas species are extracted out due to their weaker interactions with adsorbents, while the column pressure decreases to an intermediate pressure - the blowdown pressure, P_I; (3) evacuation stage: the column continues to be evacuated to an even lower pressure – evacuation pressure, P_L and the desired product is expected to be extracted in the meanwhile; (4) feed pressurization stage: the column is fed with the gas mixture until the high pressure, P_H. In some cases, P_H is set at a near-ambient pressure, while P_L and P_I are set to vacuum levels, thus resulting in vacuum swing adsorption (VSA). These four stages make up one cycle of PSA and the repeating cycles purify CO_2 in a cyclic way.

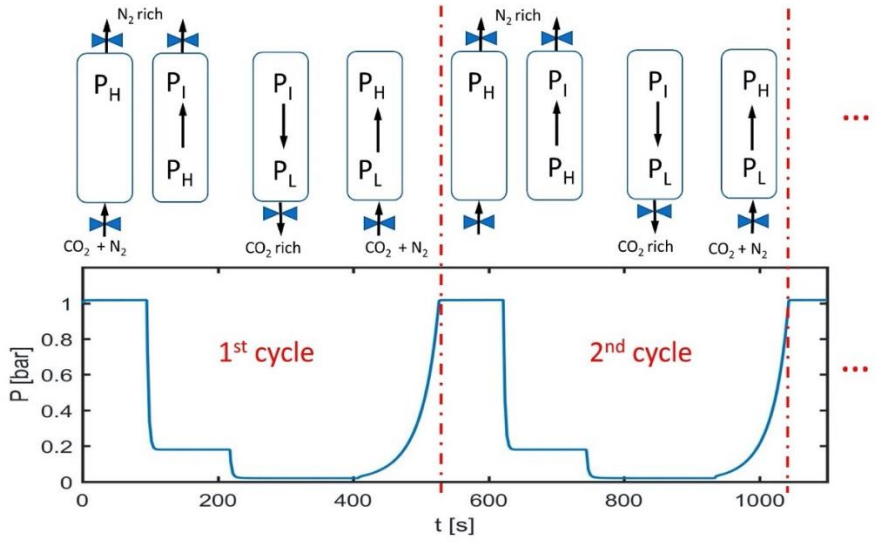


Figure 3.2. Simulation of four-stage PSA for CO_2 capture. P_H, P_I, P_L represent the pressure set point – the highest, intermediate and lowest one. The simulation is based on the operating condition $[t_{ads}, t_{bd}, t_{evac}, P_I, P_L, v_{feed}, y_{CO_2}]$ = [94.89 s, 122.84 s, 189.46 s, 0.18 bar, 0.02 bar, 0.58 m/s, 0.15].

3.2.1 Model equations

The model construction is based on the work of Haghpanah *et al.* [23]. Due to the variations in time and space, the PSA system can be mathematically described by PDEs, which are based on the mass, energy and momentum balances. A column packed with solid adsorbent is considered, and the following assumptions are used to derive the balance equations：

(1) A one-dimensional dispersed plug flow model is applied to simulate the bulk fluid flow in the axial direction.
(2) No mass, temperature, or pressure gradient exists in the radius direction.
(3) Ideal gas law is applied for the state of the gas phase.
(4) Darcy's law is used for the pressure drop in the axial direction.
(5) The thermal equilibrium between the gas and solid phase is established instantaneously.
(6) Diffusion through adsorbent pores is considered as molecular diffusion in the macropores.
(7) Multisite Langmuir model is applied to calculate the solid phase saturation loading.

Total mass balance in gas phase:

$$\frac{1}{P}\frac{\partial P}{\partial t}-\frac{1}{T}\frac{\partial T}{\partial t}=-\frac{T}{P}\frac{\partial}{\partial z}\left(\frac{P}{T}v\right)-\frac{RT}{P}\frac{1-\varepsilon}{\varepsilon}\sum_{i=1}^{n_{comp}}\frac{\partial q_i}{\partial t} \quad \text{Eq 3.1}$$

Component mass balances ($n_{comp}-1$) in gas phase:

$$\frac{\partial y_i}{\partial t}+\frac{y_i}{P}\frac{\partial P}{\partial t}-\frac{y_i}{T}\frac{\partial T}{\partial t}=\frac{T}{P}D_L\frac{\partial}{\partial z}\left(\frac{P}{T}\frac{\partial y_i}{\partial z}\right)-\frac{T}{P}\frac{\partial}{\partial z}\left(\frac{Py_i}{T}v\right)-\frac{RT}{P}\frac{1-\varepsilon}{\varepsilon}\frac{\partial q_i}{\partial t} \quad \text{Eq 3.2}$$

Component mass balance in solid phase:

$$\frac{\partial q_i}{\partial t}=k_i(q_i^*-q_i) \quad \text{Eq 3.3}$$

Energy balance inside column:

$$\left[\frac{1-\varepsilon}{\varepsilon}\left(\rho_s C_{p,s}+C_{p,a}\sum_{i=1}^{n_{comp}}q_i\right)\right]\frac{\partial T}{\partial t} = \frac{K_z}{\varepsilon}\frac{\partial^2 T}{\partial z^2}-\frac{C_{p,g}}{R}\frac{\partial(vP)}{\partial Z}-\frac{C_{p,g}}{R}\frac{\partial P}{\partial t}-\frac{1-\varepsilon}{\varepsilon}C_{p,a}T\sum_{i=1}^{n_{comp}}\frac{\partial q_i}{\partial t} + \frac{1-\varepsilon}{\varepsilon}\sum_{i=1}^{n_{comp}}\left(-\Delta H_i\frac{\partial q_i}{\partial t}\right)-\frac{2\mathrm{h}}{\varepsilon r_{in}}(T-T_w) \quad \text{Eq 3.4}$$

Energy balance in the column wall:

$$\rho_w C_{p,w}\frac{\partial T_w}{\partial t}=K_w\frac{\partial^2 T_w}{\partial z^2}+\frac{2\mathrm{r_{in}}h_{in}}{r_{out}^2-r_{in}^2}(T-T_w)-\frac{2\mathrm{r}_{out}h_{out}}{r_{out}^2-r_{in}^2}(T_w-T_a) \quad \text{Eq 3.5}$$

Pressure drop by Darcy's Equation:

$$\frac{\partial P}{\partial z}=-\frac{150}{4}\frac{1}{r_p^2}\left(\frac{1-\varepsilon}{\varepsilon}\right)^2\mu v \quad \text{Eq 3.6}$$

The mass transport coefficient given by

$$k_i = \frac{c_i}{q_i^*}\frac{15\varepsilon_p D_p}{r_p^2} \quad \text{Eq 3.7}$$

q_i^* is obtained from a dual-site Langmuir model:

$$q_i^* = \frac{q_{sb,i} b_i c_i}{1+\sum_{i=1}^{n_{comp}} b_i c_i} + \frac{q_{sd,i} d_i c_i}{1+\sum_{i=1}^{n_{comp}} d_i c_i} \quad \text{Eq 3.8}$$

where b_i, d_i are the solid phase saturation loadings of sites 1 and 2, respectively. They can be calculated based on Arrhenius-type temperature dependence:

$$b_i = b_{0,i}\exp\left(-\frac{\Delta U_{b,i}}{RT}\right) \quad \text{Eq 3.9}$$

$$d_i = d_{0,i}\exp\left(-\frac{\Delta U_{d,i}}{RT}\right) \quad \text{Eq 3.10}$$

3.2.2 Boundary conditions

The model equations for four stages of PSA are the same, while different stages are distinguished from each other by the boundary conditions of PDEs: Adsorption (Ads), Blowdown (Bd), Evacuation (Evac) and Feed pressurization (Press). t_{cycle} refers to the time point spent in a PSA cycle. t_{cycle} is always initialized as 0 when starting a new PSA cycle.

Ads $\quad t_{cycle} \in [0, t_{ads}]$

$$\frac{\partial P}{\partial z}_{|z=0} = -\frac{150}{4}\frac{1}{r_{p^2}}\left(\frac{1-\varepsilon}{\varepsilon}\right)^2 \mu\, v_{feed} \quad \text{Eq 3.11}$$

$$P_{|z=L} = P_{atm} \quad \text{Eq 3.12}$$

Bd $\quad t_{cycle} \in [t_{ads}, t_{ads} + t_{bd}]$

$$\frac{\partial P}{\partial z}_{|z=0} = 0 \qquad \text{Eq 3.13}$$

$$P_{|z=L} = \frac{1}{P_0}[P_I + (P_H - P_I)\exp^{-\lambda t_{cycle}}] \qquad \text{Eq 3.14}$$

Evac $t_{cycle} \in [t_{ads} + t_{bd}, t_{ads} + t_{bd} + t_{evac}]$

$$P_{|z=0} = \frac{1}{P_0}[P_L + (P_I - P_L)exp^{-\lambda t_{cycle}}] \qquad \text{Eq 3.15}$$

$$\frac{\partial P}{\partial z}_{|z=L} = 0 \qquad \text{Eq 3.16}$$

Press $t_{cycle} \in [t_{ads} + t_{bd} + t_{evac}, t_{ads} + t_{bd} + t_{evac} + t_{press}]$

$$\frac{\partial P}{\partial z}_{|z=0} = -\frac{150}{4}\frac{1}{r_{p^2}}\left(\frac{1-\varepsilon}{\varepsilon}\right)^2 \mu\, v_{feed} \qquad \text{Eq 3.17}$$

$$\frac{\partial P}{\partial z}_{|z=L} = 0 \qquad \text{Eq 3.18}$$

I implemented the process model in Modelica using Dymola. The weighted essentially nonoscillatory (WENO) method, a finite volume method, is applied to discretize the PDEs into DAEs. Through balancing the solving efficiency and accuracy, the number of discretized volumes is recommended as 30 by Haghpanah *et al.* [23]. The *combined discrete/continuous* feature of PSA can first be described by a superstructure formulation of all PSA stages (Eq. 3.19), and then external controls (binary variables, see Table 3.1) are imposed to determine which stage to execute. As such, the *combined discrete/continuous* PSA is transformed into a set of continuous subsystems. Each subsystem is mathematically described by DAEs.

$$PSA = Y_1 S_{ads} + Y_2 S_{bd} + Y_3 S_{evac} + Y_4 S_{press} \qquad \text{Eq 3.19}$$

Table 3.1. Binary variables for the four stages of PSA.

	S_{ads}	S_{bd}	S_{evac}	S_{press}
Y_1	1	0	0	0
Y_2	0	1	0	0
Y_3	0	0	1	0
Y_4	0	0	0	1

The simulation result is presented in Section Appx.A.1.1. Haghpanah's model has been validated experimentally [136, 137] and our simulation result is in good agreement (Appx. A.1.3) with those reported by Haghpanah *et al*. [23].

3.3 State-of-the-art of hybrid optimization framework

This work develops a hybrid optimization framework, which integrates TSEMO [54] with DyOS [138]. The characteristics of the methods are summarized in Table 3.2. TSEMO uses the input-output dataset of simulation results to train a GPs-based surrogate, which is refined iteratively by sampling new inputs for more simulation results. Within TSEMO, Thompson sampling is the acquisition function for updating the dataset. In each iteration, the surrogate model is used as the evaluation function for multi-objective optimization [54]. With these characteristics (model-based, refined iteratively with more data input, acquisition function for data exploration), TSEMO belongs to Bayesian optimization [139]. NSGA-II is the optimizer within TSEMO, so the searching strategy of TSEMO is stochastic and the optimality cannot be guaranteed. DyOS contains a local sequential dynamic optimization solver, so the searching strategy belongs to gradient-based (deterministic) optimization and the optimality can be secured. The original dynamic process model is required to calculate the gradient information, and thus the function evaluations of DyOS are based on the rigorous process model.

Table 3.2. Characteristics of TSEMO, DyOS and hybrid framework.

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

The proposed hybrid optimization framework consists of two steps. In Step 1, TSEMO searches the decision space globally to generate an approximated trade-off curve, which contains the best points obtained by TSEMO. In Step 2, DyOS is initialized at one of the best points obtained in Step 1 and improves the solution until local optimality is reached. DyOS can only improve one point per time, so the second step needs to be repeated to 'one-by-one' improve all the best points obtained in Step 1. Overall, the searching strategy is stochastic-to-deterministic, and the function evaluations are 'coarse-to-fine' type: initially the GP-based surrogate for rough evaluations, then the rigorous model for the refined evaluations. The overall optimization framework is implemented in MATLAB, as illustrated in Figure 3.3. The model in Dymola can be compiled into an executable file (Dymosim.exe) and Functional mock-up Unit (FMU), which can be seamlessly integrated into the MATLAB environment. In Step 1, the PSA model is coupled to TSEMO as an executable. In Step 2, the model is coupled to DyOS through the functional mock-up interface (FUM), and then MATLAB calls DyOS through a mex interface.

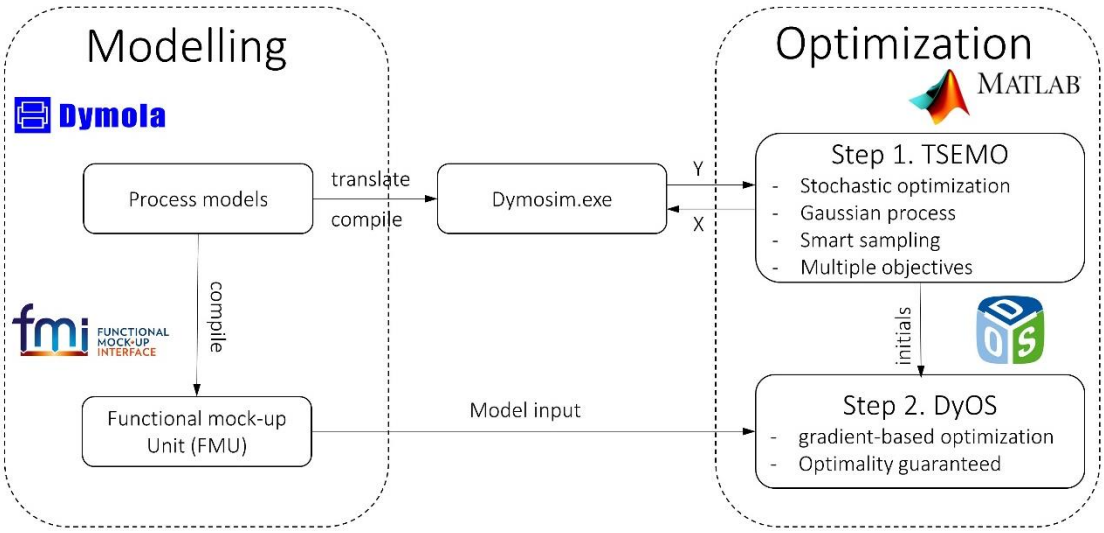


Figure 3.3. Illustration of the integrated platform for modelling and optimization of PSA. The process model of PSA is programmed using Modelica language in Dymola. The Modelica language can be translated and compiled into an executable Dymosim.exe and called directly from MATLAB. Alternatively, the Modelica model can be compiled as an FMU [140]. TSEMO runs Dymosim.exe for stochastic optimization, while DyOS takes an FMU as a model input for gradient-based optimization.

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

3.4 Optimization formulation of PSA using the hybrid framework

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

$$\text{Recovery} = \frac{\text{CO}_2 \text{ in product within a CSS cycle}}{\text{CO}_2 \text{ fed into column within a CSS cycle}} \times 100\% \qquad \text{Eq 3.20}$$

$$\text{Purity} = \frac{\text{CO}_2 \text{ in product within a CSS cycle}}{\text{total gas in product within a CSS cycle}} \times 100\% \qquad \text{Eq 3.21}$$

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

3.4.1 First step: optimization formulation using TSEMO

TSEMO can deal with multi-objective optimization problems directly, and two objectives can be inserted in the solver without any further reformulation. The formulation is constrained by the process equations (Eqs. 3.1 – 3.19). The evaluation and optimization of PSA are only meaningful after the process reaches CSS. As an evaluation method for CSS, a small tolerance value, δ, is used to check the difference between state variables x over one cycle. When $\left|x(t) - x(t + t_{cycle})\right| \leq \delta$, PSA is deemed to be under CSS. Overall, in the TSEMO optimization framework, the PSA optimization problem is formulated as follows, Eq 3.22 -Eq 3.24:

$$\max_{\boldsymbol{\theta}}(\text{Recovery},\ \text{Purity}) \qquad \text{Eq 3.22}$$

$$\text{s.t.} \quad \text{Dynamic process model (Eqs } 3.1 - 3.19) \qquad \text{Eq 3.23}$$

$$\text{CSS} = \left|x(t) - x(t + t_{cycle})\right| \leq \delta \qquad \text{Eq 3.24}$$

where θ is a vector of six decision variables of four-stage PSA system including the duration of the first stage - adsorption (t_{ads}), the duration of the second stage - blowdown (t_{bd}), the duration of the third stage - evacuation (t_{evac}), two pressure setpoints - intermediate pressure (P_I), low pressure (P_L), respectively as well as feed velocity (v_{feed}). The lower and upper bounds of the decision variables are given in Table 3.3. In this work, the highest pressure is fixed at 1 bar. The duration of the pressurization stage (the fourth stage) is reported to have a negligible effect on the operation of PSA; therefore, it is fixed to 20 s [23].

Table 3.3. The decision space in the PSA optimization *via* TSEMO. The decision space is based on Haghpanah's work [23].

$\boldsymbol{\theta}$	t_{ads} [s]	t_{bd} [s]	t_{evac} [s]	P_I [bar]	P_L [bar]	v_{feed} [m/s]
range	20-100	30-200	30-200	0.07-0.5	0.005-0.05	0.1-2

3.4.2 Second step: optimization formulation of PSA using DyOS

DyOS is designed to solve single-objective optimization problems. Herein, we reformulate the multi-objective optimization problem into a series of single-objective optimization problems

via the epsilon-constrained method [141]. In other words, the recovery remains to be the objective, while the other objective (purity of CO_2) is reformulated as an inequality constraint. Following the results from the first step, the constraint and the initial values of decision variables are based on the results obtained from TSEMO. In case that the constraint is too tight, a relaxation coefficient ($\eta = 0.99$) is given for the purity constraint (Eq 3.27). I tried to run DyOS with no relaxation ($\eta = 1$), DyOS crashed in some cases, while a relaxation can improve these situations. When optimizing PSA using DyOS, the system is assumed to reach CSS at the same number of cycles as the optimization using TSEMO (Eq 3.28). The set-up of DyOS for PSA optimization is illustrated in Appx. A.1.4. The formulation of PSA optimization in DyOS is as follows, Eq 3.25 -Eq 3.28:

$$\max_{\theta_i} \text{Recovery} \qquad \text{Eq 3.25}$$

s.t. $\text{dynamic process equations (Eqs } 3.1 - 3.19) \qquad \text{Eq 3.26}$

$$\text{purity} \geq \text{purity}_{\text{TSEMO}} \cdot \eta \qquad \text{Eq 3.27}$$

$$\text{N} = \text{N}_{TSEMO} \qquad \text{Eq 3.28}$$

The PSA optimization *via* DyOS is conducted with respect to three decision variables: intermediate pressure, low pressure and inlet flowrate, as shown in Table 3.4. In the initial trials with DyOS we included the duration variables (t_{ads}, t_{bd}, t_{evac}), which caused the method not to converge, likely because sensitivity integration over time is highly related to duration variables. Since the reason for unsuccessful termination is unclear so far, we did not include the duration variables into the optimization.

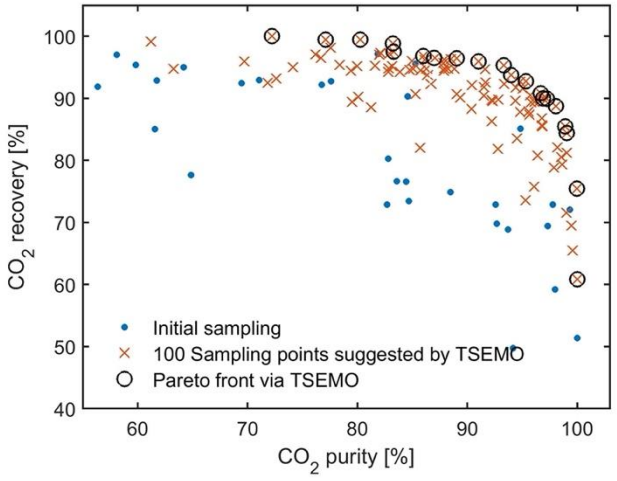
Table 3.4. The decision space in the PSA optimization *via* DyOS.

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

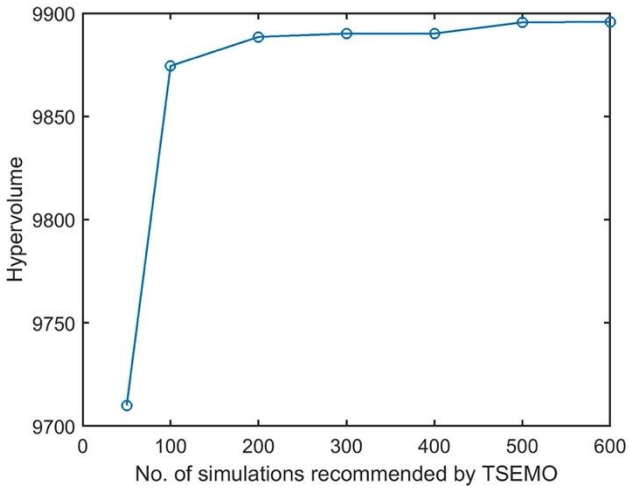
3.5 Results

3.5.1 First step: optimization using TSEMO

To initialize TSEMO, 30 random sets of inputs were sampled using a Latin Hypercube Sampling (LHS) method, and then the simulation inputs (*i.e.*, decision variables) and outputs (*i.e.*, recovery and purity) are used to train the initial GPs. Then, random samples were drawn from the GPs and multi-objective optimization is performed. Following this, new inputs for simulations were recommended by the algorithm to improve the objectives. Then, the new data points were added to the whole dataset to refine the GP surrogate in the next iteration. I present the optimization results after 7 iterations (50 simulation inputs are recommended by the TSEMO algorithm in the first two iterations, while 100 simulation inputs for the rest of iterations. As such, the 7 iterations correspond to 50, 100, 200, 300, 400, 500, and 600 PSA simulations). Figure 3.4a shows the obtained Pareto front, which represents the trade-off between recovery and purity through different numbers of simulations. The hypervolume can be used as an indicator to quantify the performance of Pareto front [142, 143]. Figure 3.4b shows that the hypervolume improves with the increase in the number of simulations. A significant improvement for the estimated Pareto front between 50 and 100 simulations is observed, while only moderate change is observed when further increasing the number of simulations. Once the number of iterations is above 200, the growth in the hypervolume is negligible (the improvement in the estimated Pareto front is negligible), which might be explained in two ways: one explanation is that the estimated Pareto front is almost close to the actual Pareto front and leaves little space for further improvement; an alternative explanation is that the searching efficiency of TSEMO considerably drops when the identified solutions are approaching optimality. This is a known issue of any stochastic search algorithm: the convergence is only guaranteed in the limit of an infinite number of function evaluations.



(a)



(b)

Figure 3.4. Multi-objective optimization of PSA *via* TSEMO. (a) Optimization results through 100 simulations recommended by TSEMO: to initialize TSEMO, LHS generated 30 simulations, shown as the blue points; the algorithm recommended additional 100 simulations, shown as the red crosses. The estimated Pareto front was evolved, shown as the black circles. (b) Hypervolume quantification (reference point is [0, 0]) varying from 50 to 600 simulations recommended by TSEMO.

3.5.2 Second step: optimization using DyOS

One issue with the stochastic global search is the lack of local refinement of the identified solutions. In particular, TSEMO does not use gradient information to improve approximate solutions further. Hence, it is desired to perform further gradient-based optimization that is

initialized from the approximated solution points obtained in the first step. Following 600 simulations *via* TSEMO, we selected 22 non-dominated points with purity over 80% and recovery over 75%, which are the starting points in the second step. For every individual point, DyOS is called to perform gradient-based optimization using the rigorous process model. As shown in Figure 3.5, DyOS slightly improves the estimated Pareto front until local optimality is satisfied. When referring to the hypervolume in Table 3.5, the improvement is not significant, which indicates that the estimated Pareto front based on the limited number of TSEMO simulations is very close to the local refined solution by gradient-based optimization.

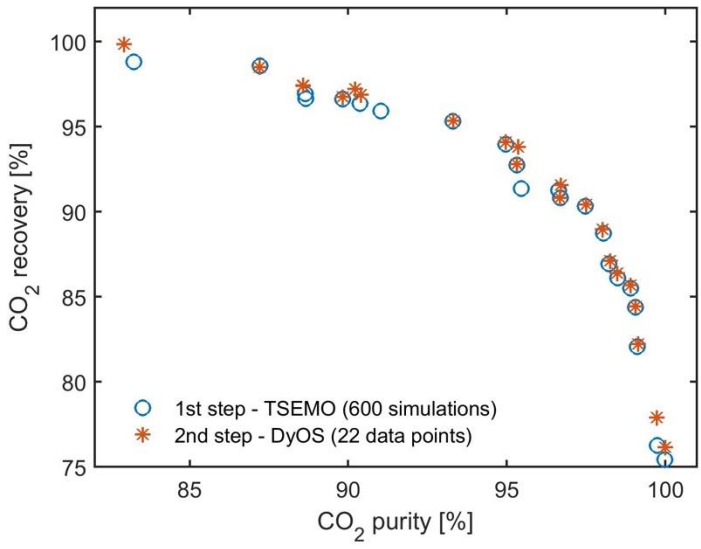


Figure 3.5. The result of the hybrid approach for the multi-objective optimization of PSA.

Table 3.5 presents the optimization performance. The hypervolume quantification indicates that DyOS does further improve the results from TSEMO. Nevertheless, the CPU time of DyOS is almost three times that of TSEMO. This is because TSEMO uses cheap-to-evaluate surrogate models and parallel computing is possible for surrogate models. By contrast, DyOS relies on gradients calculated from the sensitivity integration over all PSA cycles, and thus a large percentage of time is consumed to obtain the gradient information. Notably, the full-order physical model is evaluated to ensure accuracy, which further increases the CPU cost in the second step. Hence, the second step is time-consuming.

Table 3.5. Optimization performance *via* TSEMO and DyOS (reference point of hypervolume quantification is [0, 0]).

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

From Table 3.5, we noticed that the optimization result from TSEMO is close to that of DyOS, but DyOS costs significantly more CPU time. However, it is important to notice that the deterministic local search also offers distinct advantages for the considered case study. Firstly, DyOS verifies that the optimization result of TSEMO is 'good enough'. Without the verification, there are no criteria to check the optimality only by TSEMO. Secondly, DyOS indeed improves the optimization result. A slight improvement of operating condition may only introduce little difference in one hour for a laboratory set-up. However, such improvement can be significant for an annually operated industrial PSA plant.

3.6 Efficiency of hybrid framework

To demonstrate the efficiency of this hybrid framework, we firstly compare the performance of TSEMO with that of NSGA-II. As shown in Figure 3.6, the estimated Pareto front from TSEMO is comparable to NSGA-II, while NSGA-II requires a significantly larger number of simulations than TSEMO. As shown in Table 3.6, TSEMO with 100 simulations has a closed hypervolume value the same as the NSGA-II with 2,400 simulations, while TSEMO only uses around $1/16^{th}$ of the CPU time of NSGA-II. This is reasonable because TSEMO trains the GP-surrogate for the function evaluations during optimization, so it is not CPU-intensive as the rigorous model. NSGA-II is, actually, the optimizer within the TSEMO framework, so TSEMO has a similar exploration capacity as NSGA-II. TSEMO also employs Thompson sampling (acquisition function) to choose new sampling points, thus improving the exploitation capability. Therefore, the efficiency of TSEMO is higher than NSGA-II.

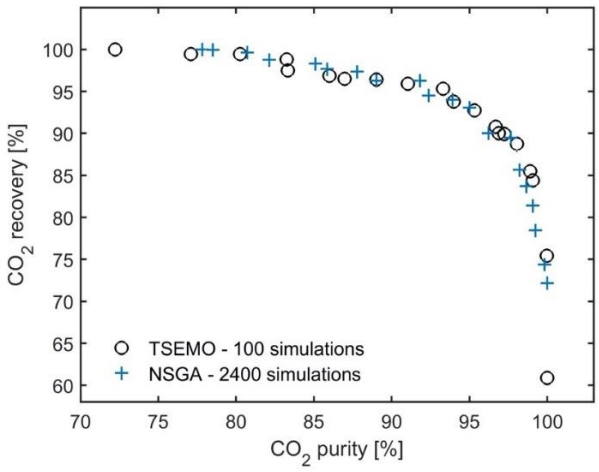


Figure 3.6. Comparison between Pareto set of solutions obtained by TSEMO – 100 simulations and NSGA-II – 2,400 simulations.

Table 3.6. Optimization performance between NSGA-II and TSEMO (reference point of hypervolume quantification is [0, 0]).

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

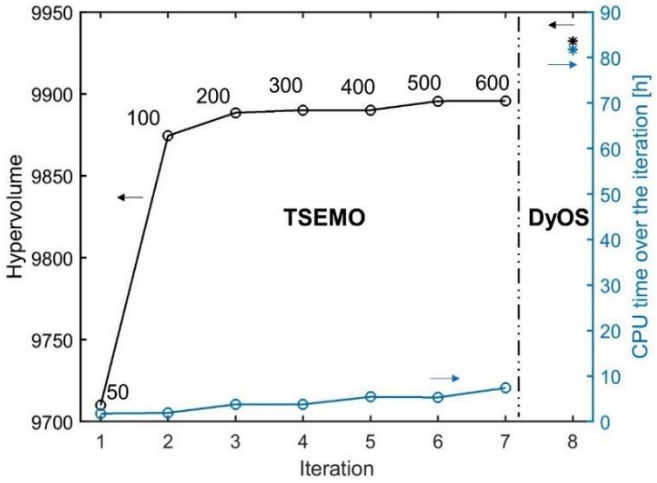
We introduce Eq 3.29 to quantify the searching efficiency:

$$searching\ efficiency = \frac{hypervolume\ improvement}{CPU\ time} \qquad \text{Eq 3.29}$$

As shown in Figure 3.7a, the growth of hypervolume slows down with the increase of iteration of TSEMO, while the CPU time starts to increase gradually. Thus, the search efficiency of TSEMO dramatically decreases after 3rd iteration. DyOS is initialized based on the result of the 7th iteration of TSEMO. The searching efficiency of DyOS is over 11 times that of TSEMO on its 7th iteration (Table 3.7). This means that TSEMO requires much more than 11 times CPU

time to achieve the same trade-off curve calculated from DyOS, given the searching efficiency of TSEMO keeps going down.

(a)



(b)

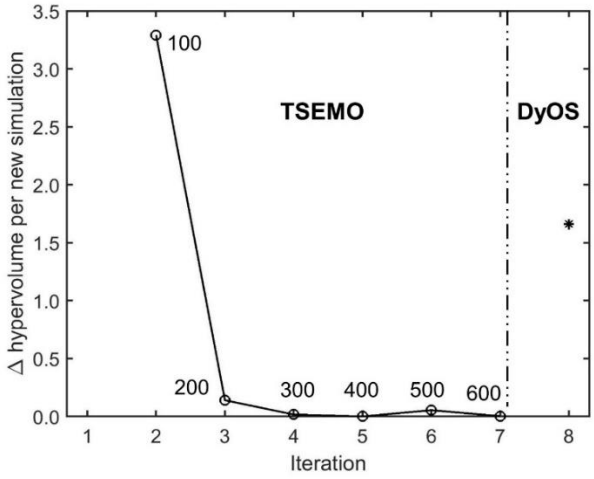


Figure 3.7. (a) Hypervolume and CPU time *via* TSEMO and DyOS (the reference point of hypervolume quantification is [0, 0]). (b) The average hypervolume improvement when a new simulation is added. Iterations 1 – 7 refer to the influence of TSEMO, which recommends 50, 100, 200, 300, 400, 500, and 600 simulations, respectively. Iteration 8 refers to the influence of DyOS based on 22 data points.

TSEMO is focused on space-filling rather than the improvement of individual points as DyOS. Both TSEMO and DyOS tend to find better results than the last iteration, but the improvement on individual points is quite different. As shown in Figure 3.7b, the average hypervolume improvement on an individual point drops significantly with the increase of TSEMO iteration,

while DyOS can still take advantage of the gradient to further optimize the individual point. As shown in Table 3.7, the difference can be 553 times when comparing between DyOS and the last iteration of TSEMO, regarding the hypervolume improvement of an individual point. In other words, in the proximity of an optimal solution, DyOS possesses a significantly higher exploitation capacity than TSEMO.

Table 3.7. Searching efficiency *via* TSEMO and DyOS (reference point of hypervolume quantification is [0, 0]).

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

3.7 Conclusions and reflections

This Chapter has developed a hybrid (TSEMO + DyOS) optimization framework to secure a high searching efficiency for a four-stage PSA process with an application in CO_2 capture. In the hybrid optimization framework, the first step employs a stochastic optimization algorithm –TSEMO- to search the entire decision space efficiently. This step identifies an approximate Pareto front of two objectives, CO_2 purity and recovery. In the second step, DyOS starts from the most promising objective points obtained in the first step and further improves the optimization result of PSA until optimality. The second step consumes nearly three times of CPU time compared to the first step, but the improvement in the 2^{nd} step is negligible, which indicates that TSEMO can achieve nearly optimal solutions within the limited number of simulations.

The hybrid optimization framework possesses an excellent optimization efficiency. Such efficiency benefits from the coarse-to-fine function evaluations and stochastic-to-deterministic

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

This hybrid multi-objective optimization framework can explore other competing criteria, such as energy consumption and productivity of PSA. Further, this approach can be extended to optimize any other complex expensive-to-evaluate dynamic processes. TSEMO seems to already deliver a 'good-enough' trade-off curve among multiple criteria in a relatively low time cost, while the hybrid framework can be used to accelerate the trade-off curve to converge to the real 'good-enough' solution.

From chapter 3, we can see that achieving optimality is computationally expensive even starting from a nearly optimal solution. After this project, I reflect that the necessity of pursuing optimality depends on the objective and model.

Objective: in the multi-objective optimization of PSA, one objective is purity of CO_2. A purity around 90% is commonly acceptable [144], while exploiting the improvement space (achieving optimality) is not that important. By contrast, a high purity for silicon in semiconducting industry can be essential, and optimality for purity is more deserved to exploit.

Model: the optimization is based on a model, but it is almost impossible to build a 100% accurate model as physical system. When the objective is related to economic factors, the model accuracy might be significantly influenced by external disturbances – market prices. For example, the methanol price in the EU considerably fluctuated, ranging from 125 to 525 euro/ton in the past 20 years [145]. Additionally, the measurement for the physical system has noises. Specifically, this PhD project is concerned with the conceptual process design of CCU, the knowledge for the system is far less than 100%. Hence, the model is not possible to be 100% accurate. An occasion probably occurs when the model error is larger than the difference

between nearly-optima and optima. As such, pursuing optimality may result in a problem - optimizer's curse, where the optimizer recommends a solution with the optimality, but this solution is not achievable in reality. Therefore, it might not be so meaningful to pursue optimality unless the model has extremely high accuracy.

Therefore, the remainder of this thesis will not guarantee optimality, and the optimization method is mainly stochastic surrogate-based optimization.

Chapter 4 Performance of different machine learning-based surrogates

TSEMO, mentioned in the prior chapter, uses GPs as surrogates and proves the efficiency of surrogate-based optimization. However, numerous surrogates exist for regressions, and the question remains which is the most suitable one for the process systems. This chapter used the direct CO_2 hydrogenation to methanol process as a toy problem and tested various surrogate types, which were then shortlisted to ANNs and GPs based on their fitting performance on a single output. Following this, ANNs and GPs were further tested to fit multiple outputs simultaneously and then the surrogate-based optimization. The performance of ANNs is slightly better than GP, while ANNs show better flexibility for fitting multiple outputs. Notably, the bottleneck for surrogate-based optimization of the investigated process is the time for data generation, regardless of surrogate types.

4.1 A toy problem – direct CO_2 hydrogenation to methanol

In the field of carbon utilization, the direct CO_2 hydrogenation to methanol (DCHM), see RX 4.1, has attracted increasing attention in academia, as indicated in excessive research works in catalyst development [92, 94] and process design [14, 20, 27, 113]. DCHM is reported to form fewer by-products and require lower heat [14]. As the product, methanol is an alternative fuel and an essential intermediate to valued chemicals. To accelerate the lab research to the plant scale, optimization can help achieve the optimal design and operation for DCHM process.

$$CO_2 + 3H_2 \leftrightarrow CH_3OH + H_2O \qquad \Delta H_{298K} = -49.5 kJ/mol \qquad \text{RX 4.1}$$

The DCHM process is implemented in Aspen Plus, which is mainly referred to the work of Kiss *et al*. [113]. As shown in Figure 4.1, DCHM starts with two feedstocks - CO_2 and H_2. H_2 flows into the process *via* a stripper, where CO/CO_2 components can be primarily recycled to

the gas stream. Following this, the gas mixture is compressed and then mixed with the CO_2 feed stream, which is preheated by a heat exchanger before the reactor. The reaction is exothermic, so the reactor outflow can supply the heat for the heat exchanger (HX). The reactor outflow is further cooled down and then flashed in a gas-liquid separator to split the gas components ($CO/CO_2/H_2$) from the liquid products. Sequentially, a stripper can enhance the recycle rate of gas components and sent them back to the recycle stream. For the purpose of easier simulation convergence, a splitter is used to purge partial recycle stream (0.1%~1.5%). Eventually, a distillation column is employed to purify methanol from the liquid products.

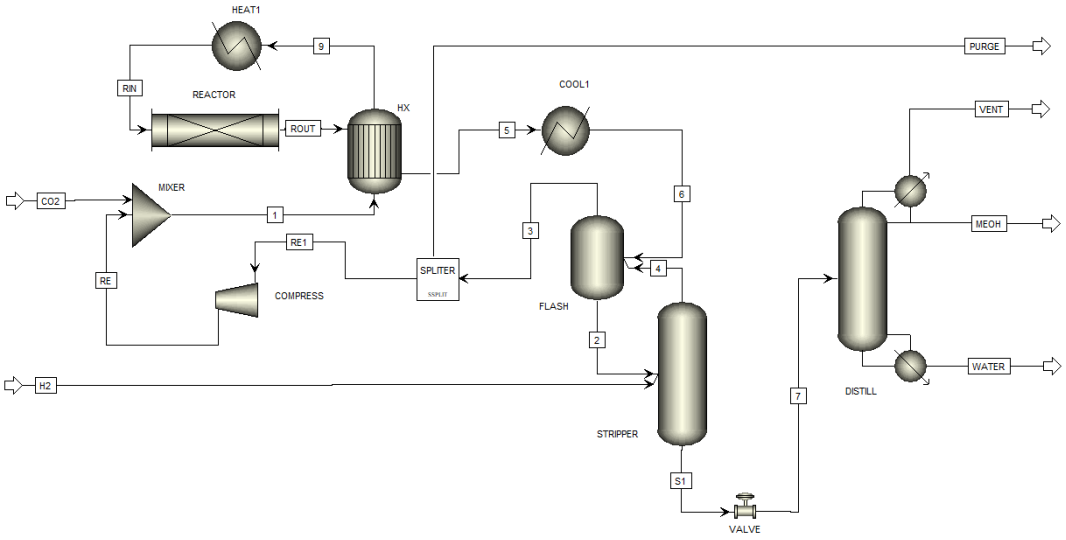


Figure 4.1. Flowsheet for the process of direct CO_2 hydrogenation to methanol (DCHM).

Redilich-Kwong-Soave (RKS) equation of state is selected as the thermodynamic method for the simulation for the most part of the DCHM process (P>10 bar) involved with polar components. For the distillation column (P<10 bar), non-random two-liquid (NRTL), an activity coefficient model, is selected as the thermodynamic method for the simulation of low-pressure operation [146]. Further, I list the specifications of unit operations in Table 4.1.

Table 4.1. Specifications for the DCHM in Aspen Plus.

Reactor	RPlug; Graaf's kinetic model; Isothermal; 200-300 ℃; 50-100 bar.
Stripper	RadFrac; Equilibrium; No condenser/reboiler; Convergence: Petroleum/wide-boiling; no absorber;
Distillation column	RadFrac; Equilibrium; partial-vapor-liquid for condenser (the existence of a little H_2 can affect the convergence of distillation column); Convergence: *strongly non-ideal liquid* algorithm (for the water-methanol system, the strong interaction exists due to hydrogen bond between water and methanol); Through design specifications varying the distillate rate and reflux ratio, both the recovery rate and purity of methanol can achieve 99%.

For an optimal design and operation of DCHM process, the energy is desired to be minimized as well as the yield of methanol is maximized, which yields a multi-objective optimization problem as follows,

$$\min_{\theta}(E_{total},\ -Y_{MeOH}) \qquad \text{Eq 4.1}$$

$$\text{s.t.} \quad LB \leq \theta \leq UB \qquad \text{Eq 4.2}$$

Where E_{total} is the total energy consumption of DCHM process; Y_{MeOH} is the yield of methanol; θ is the decision variables; LB is the lower bound; UB is the upper bound.

Gradient information is hard to be extracted from the simulator, so simulation-based optimization seems to be a suitable choice. If the Aspen plus simulation is called iteratively for the objective function, too much unnecessary information (not directly related to the decision variables in the optimization) is generated (as shown in Appx.A.2.2). By contrast, a more straightforward way is to develop a reduced-order surrogate, where decision variables can be directly related to the objective functions. Table 4.2 describes decision variables and objectives,

which are the input/output for DCHM surrogate. Once a surrogate is established, the prediction capacity of the surrogate is evaluated by a separate dataset.

Table 4.2. Decisions (input) and objectives (output) for CO_2 hydrogeneration to methanol.

Input $\boldsymbol{\theta}$	Design space	Unit	Notes
F_{H_2}	1000-1500	[kmol/h]	Inlet flowrate of H_2
T_R	200-300	[℃]	Temperature in DCHM reactor
P_R	30-50	[bar]	Pressure in DCHM reactor
T_F	20-40	[℃]	Temperature in flash
P_F	30-45	[bar]	Pressure in flash
N_{trays} (integer)	20-30	[-]	No. of trays in distillation column
$Split_{purge}$	0.001-0.015	[-]	Split fraction to purge (the other to recycle)
Output			
E_{total}		[Gcal/h]	Total energy consumption
$-Y_{MeOH}$		[%]	Negative value of MeOH yield

4.2 Data generation

I briefly demonstrate the procedure for data generation. As shown in Figure 4.2, the inputs are sampled by LHS in MATLAB and passed to simulators, *i.e.*, Aspen Plus for steady-state process simulations of DCHM (in Chapter 5, Dymola for dynamic simulations of PSA). The obtained outputs are sent back to MATLAB for data collection. In the case of DCHM, 5557 input-output data points were generated.

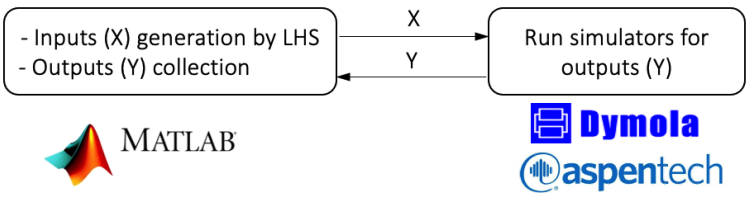


Figure 4.2. Data generation by interfacing MATLAB with process simulators.

4.3 Fitting a single output

With the obtained 5557 data points, I would like to explore the regression capacity of various surrogate types on individual output (either E_{total} or Y_{MeOH}). This stage is mainly to roughly screen the potential surrogate type, so I did not optimize the hyperparameters in details for each surrogate type. To generate a variety of surrogate types, I used MATLAB -> *Statistics and Machine Learning Toolbox -> regression learner APP*, which can easily access linear regression, SVM, decision tree, decision trees ensemble and GP. To generate ANN surrogate type, I used *fitnet* function to generate one-layer network. Herein, the whole dataset was divided into two parts: training and test with a ratio at 80% / 20% (these data are sampled in a random way by LHS, so there is no need to introduce randomness in splitting the whole dataset). These surrogate types were regressed by fitting the relationship between input and an individual output based on the training dataset, followed by the assessment based on the test dataset. Table 4.3 shows the performance of surrogates by the root mean square error (RMSE) based on the test dataset.

Table 4.3. Regression performances for individual outputs by various surrogates.

Surrogate types	RMSE	
	E_{total}	Y_{MeOH}
Linear regression (Linear)	45.994	3.454
Linear regression (Interactions)	42.429	3.383
Linear regression (Robust)	49.426	3.454
Linear regression (Stepwise)	42.365	3.385
SVM (Linear)	51.340	3.475
SVM (Quadratic)	32.899	1.495
SVM (Cubic)	23.910	0.702
SVM (Fine Gaussian)	55.863	4.640
SVM (Medium Gaussian)	24.538	0.636
SVM (Coarse Gaussian)	39.468	1.847
Decision tree (Fine)	14.907	1.579
Decision tree (Medium)	16.401	1.717
Decision tree (Coarse)	22.281	1.926

Ensemble (Boosted Trees)	16.085	4.141
Ensemble (Bagged Gaussian)	15.796	1.168
GP (Squared Exponential GPR)	14.910	0.319
GP (Matern 5/2 GPR)	14.596	0.290
GP (Exponential GPR)	17.940	0.489
GP (Rotational Quadratic GPR)	14.588	0.302
ANN ([5])	12.061	0.543
ANN ([10])	9.640	0.304
ANN ([15])	9.806	0.258
ANN ([20])	11.489	0.170

Not surprisingly, linear regressions fail to deliver an excellent fitting performance. That is because input/output is highly nonlinear due to the complex thermodynamics and kinetics as well as the recycle within the DCHM flowsheet. SVM cannot regress the relationship well probably since SVM is primarily designed for classification rather than regression [147]. Further, decision trees and the ensemble of decision trees can give a fair fitting result, but the surrogate predictions for test data are not well distributed in the parity plot. Inspiringly, Gaussian Process (GP) outperforms the above-mentioned surrogates in the parity plot distribution. Among the tested kernels, Rotational Quadratic delivers moderately better performance than other kernel types. Furthermore, I considered single-layer ANNs with neurons of 5~20. All ANNs can regress the input/output quite well while the number of neurons slightly affects the fitting performance. This indicates that network parameters (*i.e.*, neurons, layers, and activation functions *etc.*) can be optimized for a slightly better regression performance.

Overall, through exploring various surrogate types, I narrowed down the surrogate options to GP and ANN for the application in the regression of process systems. Rotational Quadratic is identified as the suitable kernel for GP. ANN can fit the output better than GP, but ANN requires extra effort in structure optimization before its application.

4.4 Fitting multiple outputs

For the purpose of optimization, the generated surrogate has to fit two objectives of DCHM (multiple outputs). Similar to the workflow in Section 4.3, Figure 4.3 presents a workflow to

fit the multiple outputs simultaneously by ANN or GP. With the input/output training dataset, we can train a GP- or ANN-based surrogate. These two types of formulation are detailed in Sections 4.4.1 and 4.4.2.

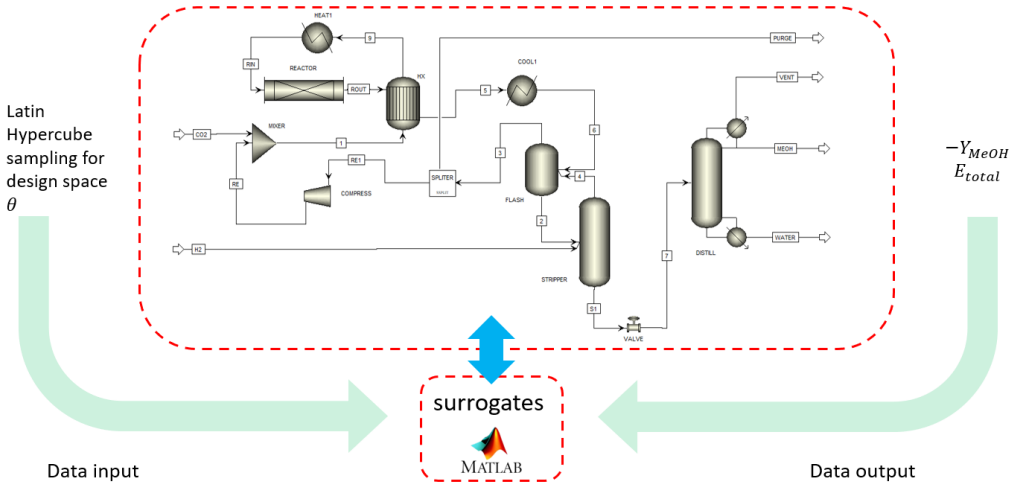


Figure 4.3. Surrogate by GP or ANN for direct hydrogenation to methanol (DCHM).

The sampled data points (5557 in total) are split into training dataset and test dataset with a ratio at 90% / 10% (5000 for surrogate training / 557 for test). The test dataset is the assessment basis to compare ANN- and GP-based surrogates.

4.4.1 ANN

One ANN can regress multiple outputs simultaneously. Since the relationship of inputs/outputs is highly nonlinear, I selected a nonlinear activation function - hyperbolic tangent (tanh) - for regression. The network structure needs to be optimized: random search was used to optimize ANN parameters - the structure of networks, *e.g.*, the number of layers and the number of neurons. I gradually increased the total layer number. Each layer was allocated to a random number of neurons and this step was repeated to create sufficient network candidates. The hyperparameters of these random structures were regressed to fit the 80% of training dataset. The best network candidate was identified by the minimal MSE value on the other 20% of the training dataset. Because the two outputs have different ranges of values, normalization is required to make them in the same scale (as to fit two outputs equally) during the surrogate training: following the MATLAB documentation, I used the suggested setting – set the

normalization performance parameter to 'standard', where all outputs are normalized between -1 and 1. Following this, the obtained networks are evaluated by the output prediction for the test dataset: I used the MSE function (normalization with 'standard') in MATLAB to compare the test dataset with the surrogate prediction; meanwhile, I used the normalized root mean square error (NRMSE as Eq 4.4) to assist the evaluation.

$$RMSE_i = \sqrt{\frac{\sum_{j=1}^{j=N}\left(x_{i,j} - \hat{x}_{i,j}\right)^2}{N}} \quad \text{Eq 4.3}$$

$$NRMSE_i = \frac{RMSE_i}{\bar{x}_i} \quad \text{Eq 4.4}$$

where, i refers to the index for output ($i = 1$ for energy; $i = 2$ for yield).

j refers to the index for data points.

x refers to the value of test data.

$\hat{x}$ refers to the prediction of surrogate.

$\bar{x}$ refers to the average value of test data.

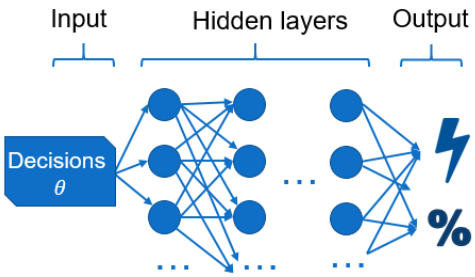


Figure 4.4. Regression of DCHM process by ANN.

I explored the total (hidden) layer number ranging from 1 to 5. Table 4.4 shows that no significant difference was observed for the *MSE* values based on the different-layer networks. Among all the structures, the network with two layers delivered a slightly better regression performance for the test dataset. Herein, deep learning with too many layers is unnecessary in this case. Further, for the two-layer structure, I increased the searching times from 60 times to 1000 times and allowed more neurons in the ANN structure. However, the improvement seemed not significant, either. The default *MSE* function in MATLAB indeed decreased when

increasing the searching times, but the NRMSE does not show any advantages. In other words, there are few points in spending excessive time searching for a 'perfect' network structure. Additionally, a more extensive network structure (containing many layers and neurons) means a higher number of parameters, which can introduce a higher risk of overfitting issues. Therefore, a small network structure (2 layers) is preferred, and 60 searching times are used to optimize network structure in the surrogate generation for other CCU process options.

Table 4.4. Optimization of network structure.

Total number of layers*	1	2		3	4	5
Search times	60	60	1000	60	60	60
Max neurons	60	60	100	60	60	60
Time per search [s]	3.48	2.37	2.37	3.17	4.22	4.79
Best network structure [Neurons in each layer]	[19]	[10,25]	[17, 13]	[6,12,28]	[6,24,3,26]	[11,8,17,19,3]
MSE_{MATLAB} (normalized)	5e-4	3e-4	2e-4	4e-4	5e-4	5e-4
$NRMSE_{energy}$	6.79%	4.75%	5.04%	5.74%	6.88%	7.01%
$NRMSE_{yield}$	0.25%	0.22%	0.28%	0.25%	0.24%	0.26%

*layer refers to the hidden layers within the network.

4.4.2 GP

The formulation for GP-based surrogates can be more straightforward than ANNs. Since GP is a non-parametric model, no extra workloads are needed to adjust the model parameters as ANNs. The only prior work is the selection of kernel. As concluded in Section 4.3, Rotational Quadratic is selected as the suitable kernel for the DCHM process. However, one GP is generally designed to regress one single output, whereas the formulation for fitting multiple outputs can be complex [58, 59]. An alternative solution is to implement two GPs for the two output variables - yield and energy (Figure 4.5). Two GPs are trained, separately. Herein, the two GPs make up the GP-based surrogate for the DCHM process.

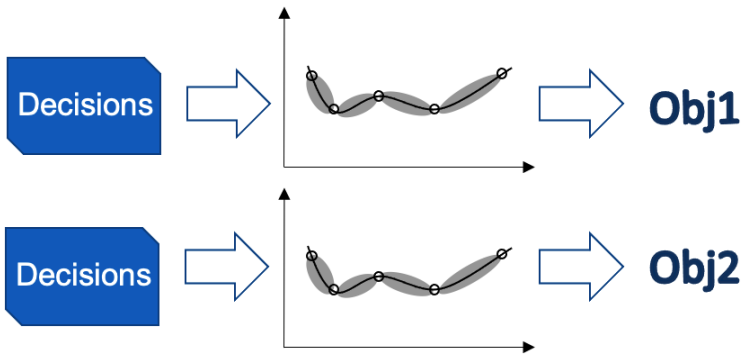


Figure 4.5. Regression of DCHM by GPs.

4.4.3 Performance check

I used the boxplot to check the fitting performance by the relative errors of two outputs between the surrogate predictions and the test data. To train the surrogates, I gradually increased the training data points from 500 to 5000. Herein, the training dataset was used to adapt the surrogate model to data. For ANN, the training dataset was split into 80% for hyperparameter tuning and 20% for parameter tuning. For GP, 80% of the training dataset was used for hyperparameter tuning, and 20% of training dataset for validation. As shown in Figure 4.6, the overall fitting performance can significantly improve when increasing the number of training data points from 500 to 2500. Yet, no significant improvement is observed when further doubling the number of training data points to 5000. With the same amount of dataset (500, 2500, 5000), ANN performs slightly better than GP. Notably, the relative error of energy is much larger than the yield of methanol (<10% for energy, <0.5% for yield can be obtained based on 5000 training data points). This means that the mass flow can be well fitted, whereas the fitting of the total energy consumption is not perfect. Two outputs are fitted based on the same amount of data points, so the reason should not be the insufficient number of data points. One possible reason is the lack of relevant features (input variables). Yet, I summed all the utility consumptions together, thus resulting in the difficulty in checking the hotspots of error sources.

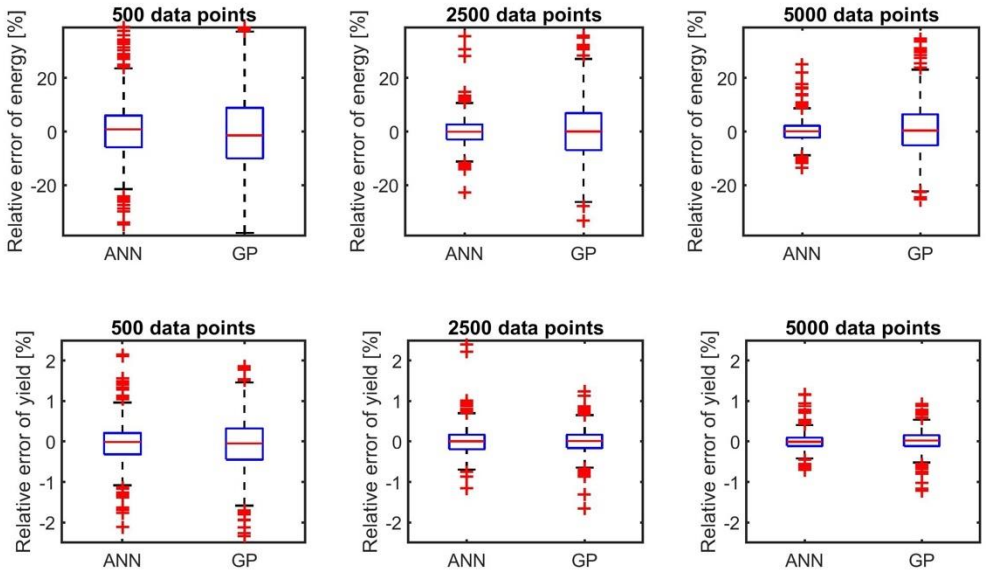


Figure 4.6. Regression performance for two outputs by ANN or GP, based on the increasing number of training data points.

Although the regression result shows that ANN is slightly better than GP, further evaluation might be required to confirm this comparison result by the two surrogate types. The training of GP is computationally intensive for large datasets, with a time complexity of $O(n^3)$, where n is the number of data points [148]. GP scales poorly with the growth of data points, meaning the significant increase of training time, which leads to a disadvantage in the sequential sampling. This high computational complexity even prohibits the application of GP for the regression of moderate-sized datasets [149]. In this case study, the initial dataset size is 500, which might excess the regression capacity of GP. GP only has a small number of hyperparameters to optimize, so the required dataset is theoretically smaller than ANN. Furthermore, ANN can only produce one output given one input, while GP inherently can access uncertainty estimation [150], which make GP more advantageous, because the real-world process systems are subject to uncertainty.

4.5 Surrogate-based optimization

Eventually, I used the obtained surrogate for the application of optimization. Optimization with direct Aspen Plus model of DCHM is used as a reference: MATLAB is used to run the Aspen

Plus simulations of DCHM in an iterative way; NSGA-II varies the values of decision variables in each iteration; NSGA-II generates Pareto front (the trade-off curve between two objectives). As shown in Figure 4.7, Pareto front by the surrogates has a significant deviation from the result by the rigorous model (direct Aspen). With the obtained values of decision variable in surrogate-based optimization, I ran rigorous simulations *via* Aspen Plus, which delivered the Pareto front denoted as ANN-Aspen or GP-Aspen. As shown in Figure 4.7, ANN-Aspen or GP-Aspen gives a similar Pareto front as the rigorous optimization. This is because the surrogate can identify the improvement direction for decision variables, although the model accuracy is decreased. When the surrogates can provide the right improvement direction, the final result can still improve. Therefore, the surrogate-based optimization is capable of guiding the DCHM system to approach the optimal operating conditions.

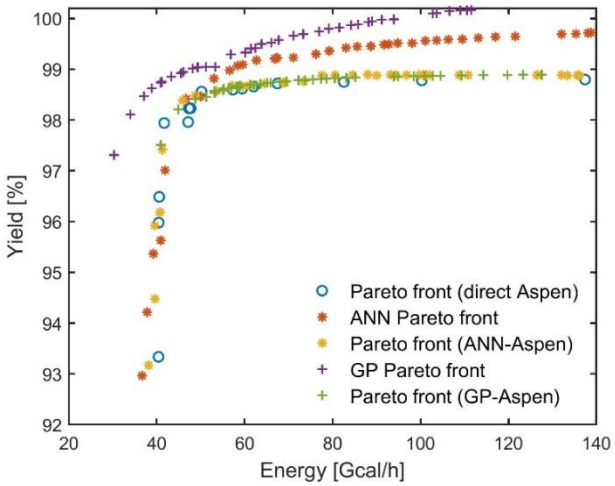


Figure 4.7. Optimization *via* ANN and GP surrogates.

I analysed the time breakdowns spent on optimization. Surrogates offer a cheap-to-evaluate objective function, so the optimization time can be significantly reduced compared to the optimization with direct Aspen simulations (direct Aspen for 935 min, with the same setting for the parameters in the optimizer - GA). ANN requires time to tune network parameters, so ANN requires more time than GP in surrogate training. Also, the GP model costs less time in optimization (GA in Table 4.5) than ANN, which means that the evaluation of GP model is faster than that of ANN model. Further, the most notable finding here is that data generation can take up over 95% of time in the whole surrogate-based optimization.

Table 4.5. Time spent on each segment of surrogate-based optimization.

CPU time [min]	5557 data generation	Training	GA	Validation (Aspen simulations)	Total
ANN	483	18.03	1.05	6.49	508.57
GP	483	5.25	0.19	3.75	492.19

4.6 Conclusions

In this chapter, I explored various surrogate types, in which GP and ANN were identified as two suitable ones to fit the selected process systems. Although the optimization performance of ANN is similar to the GP, ANN shows a slightly better regression capacity and more straightforward implementation for multiple outputs. Therefore, ANN will be employed as the primary tool for surrogate generation in the following study.

Additionally, the hotspot of surrogate-based optimization is the data generation for process systems. One-shot sampling can easily result in oversampling, so sequential sampling can be an option. This motivates me to develop an efficient workflow to generate surrogates by reducing time on data generation iteratively (see Chapter 5).

For ANN-based optimization, the second largest time is normally spent on tuning parameters and hyperparameters. In this work, I find that excessive searching times are unnecessary for the network tuning. Also, I explored not only the shallow networks but also deep networks. As a result, I can find that it seems worthless to explore deep networks, which should also apply to similar process systems. Based on this experience, the training times for ANN surrogates can be largely decreased in future research works.

In this case of surrogate for DCHM, the energy term has a much larger relative error than the mass term. We may consider two strategies to identify the error sources for similar systems in future works. The first strategy is heuristic given the sense of process engineering, where the total energy (one output) can be split into different heating/cooling/electricity utility types (multiple outputs). The corresponding errors due to different energy sources can be explored,

followed by adding the relevant features for the improvement. The second strategy can be machine learning-based feature selection/reduction techniques [151, 152], where numerous inputs are considered initially and adjusted in an iterative way. The second strategy is out of the scope of this thesis, and I will apply the first strategy in the following works.

Chapter 5 An efficient workflow to generate surrogates

Chapter 5 builds an efficient workflow for generating surrogates *via* ANNs. The key of this workflow is to reduce the time for data generation. The time for data generation can be reduced by: (*i*) applying an SVM classifier to improve data quality and avoid evaluation of infeasible inputs, and (*ii*) employing a slowdown sampling strategy to reduce data quantity [153]. The slowdown sampling strategy links a dynamic sampling rate to the quality of regression: the initial sampling rate is large to generate enough data for surrogate regression in a few iterations; the sampling rate gradually slows down (fewer and fewer data points are sampled, with the increasing number of iterations) with the regression improvement of the iteratively refined surrogate. A dynamic process and a steady-state process from the field of carbon capture and utilization are used as two case studies: pressure swing adsorption (PSA) and Gas-to-Liquids (GTL). A GTL complex contains the reforming of natural gas and Fischer-Tropsch synthesis for the production of liquid fuels. With the proposed methodology, the computational costs for surrogate generation are reduced by 86% for PSA and 51% for GTL, compared to the computational costs when using a static sampling rate.

5.1 Introduction

To build surrogates, one of the prerequisites is data, the generation of which can be prohibitively expensive for real-world engineering systems. Conventional sampling methods can lead to under/oversampling issues [154]. Our strategy is to develop a workflow to reduce the total time spent on data generation by: (*i*) lowering the total number of the required data points, and (*ii*) shortening the time per data generation.

To successfully set up such a methodology, it is beneficial to review prior works on sampling methods. Data generation can be extremely expensive for real-world engineering systems [52]. To demonstrate it, we consider an example of a chemical process – pressure swing adsorption (PSA) [80, 81, 85]. Data is sampled iteratively: in each iteration, 750 data points are sampled and a surrogate is trained. As shown in Figure 5.1, the computational cost for data generation has a significantly higher order of magnitude than that for surrogate training, which is one of the common problems in process systems. Insufficient data quantity cannot guarantee good quality for constructing a surrogate, while Garud *et al.* review that simply increasing the data quantity cannot lead to better performance of a surrogate [154]. Thus, the quality of surrogates should rely on both - data quantity *and* quality.

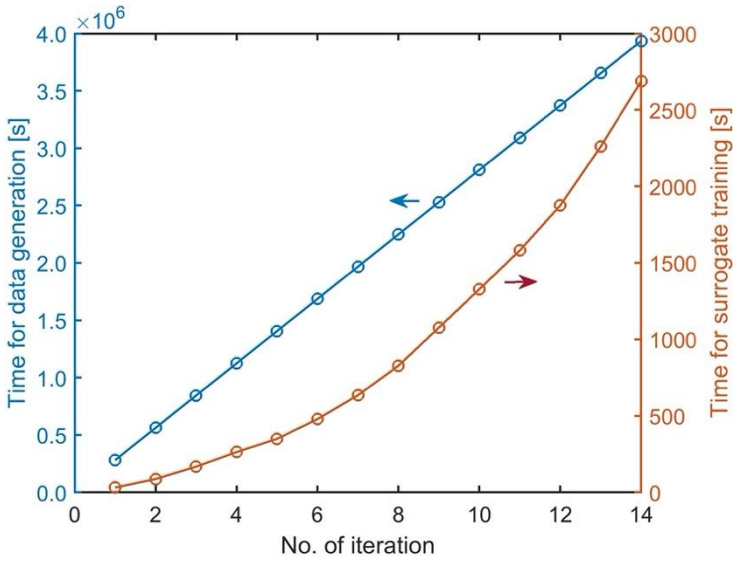


Figure 5.1. Computational costs on data generation *vs.* surrogate training for PSA process.

To reduce the time for data generation, the first objective is to obtain good-enough surrogates with the minimum amount of data. There are two types of sampling methods: one-shot and sequential (adaptive) methods [42]. The former method samples the design space uniformly in one go and then builds a surrogate, while the latter samples data in batch and refine the surrogate iteratively. In recent years, sequential methods tend to be popular because they are reported to better balance the regression performance and efficiency [42].

However, oversampling is still hard to avoid by a typical sequential method. To demonstrate this, we still use the example of the PSA process. Mean squared errors (MSE) is employed to

evaluate the regression performance. I plotted the fitting performance against time to observe the termination condition, as shown in Figure 5.2. I divided the sequential sampling as two equal parts based on the number of data points (approximately equivalent to time, because the time for surrogate training can be neglected compared to data generation as shown in Figure 5.1). The plot indicates that regression improvement in the first half is significantly greater than that in the second half. This suggests that too much data is not worth collecting. In other words, further sampling should be stopped after achieving a certain fitting performance. Also, it is noticed that the MSE values fluctuate all the time. Hence, it is rather challenging to determine an optimal termination criterion.

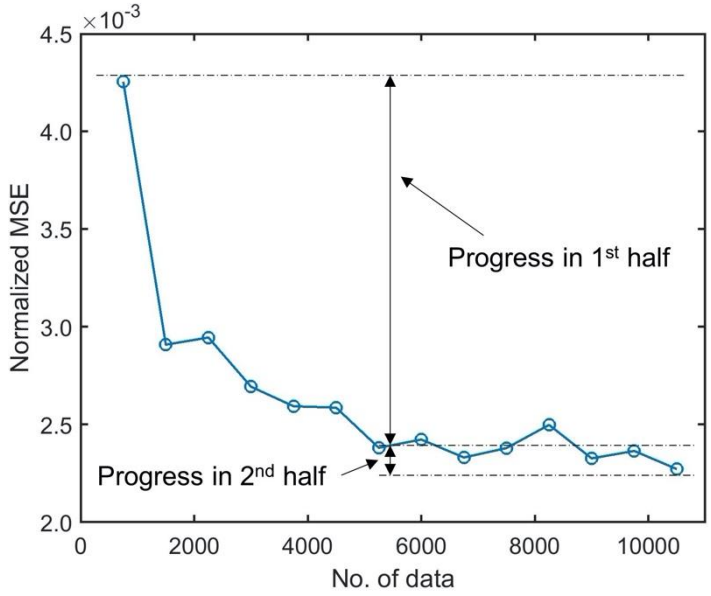


Figure 5.2. Illustration of why too many data points might not be worth sampling.

Meanwhile, with the linear increase in the number of data points generated, time seems to exponentially increase for surrogate training (Figure 5.1). Consequently, the surrogate training might be extremely time-consuming if the number of sampled data points is high. Therefore, oversampling brings unnecessary computational costs for data generation and extra effort for surrogate training. To avoid oversampling, it might be beneficial to spot the non-improvement trend as early as possible, which is a problem to be solved in this chapter.

The second objective in sampling is concerned with the improvement of data quality. The design space for sampling is initially based on limited prior experience or even random guesses,

and the infeasible design space is commonly unavoidable. Consequently, some inputs, which happen to be sampled from the infeasible design space, can lead to unexpected outputs, such as non-converged simulation outputs or experimental failures. Such outputs will introduce significant errors to the surrogate construction. To increase data effectiveness, a classifier can be constructed to distinguish between infeasible and feasible design spaces. Such application of a classifier has been successfully demonstrated in prior research works. Ibrahim *et al*. reported that a support vector machine (SVM) can be used to set a feasibility constraint to filter infeasible design space for non-converged simulations [155]. Cao *et al.* adopted a Bayes classifier to improve the design space for the experimental conditions of formulations [37]. Kim *et al.* applied a combined classification system to increase the quality of design space for computation-based material discovery, which can significantly reduce the number of further samplings [156]. Houben *et al*. included a classifier into a Bayesian optimization algorithm to avoid infeasible experiments in emulsion polymerisation [157].

To further enhance data quality, exploitation-based methods can be considered to identify the promising sample placement. The exploitation-based methods tend to place more samples in the highly nonlinear/complex regions [42]. Cozad *et al*. developed a workflow called ALAMO for algebraic model building in a sequential sampling way [132, 158]. For new data to sample, they apply a derivative-free optimization technique to identify the sample placement, which holds the largest error between the surrogate and the original model. To identify one optimal sample placement, many new data points are required to be generated for evaluation during the optimization. Consequently, this method generates far more data points than the reported number of optimal data points. An alternative approach is to employ GP-based surrogates, which can predict the model uncertainty. The region with the largest uncetainty is selected for new points [159, 160]. However, this approach is limited to GP-based surrogate type, since the error prediction is not a generic characteristic for other surrogate types [154]. Garud *et al.* review that the surrogate-independent strategies can be more advantageous, because they can be more generic and can guarantee sampling randomness [154]. Most of these strategies are based on specific score criteria to identify complex regions, which then require exploitation-based methods for local improvement. Since it is out of the scope of this work, more detailed information can be referred to in Garud's review paper [154]. Although the exploitation-based

approaches are powerful in improving data quality, the complex mathematical formulations make them difficult in implementation.

Herein, I aim to develop a generic and easy-to-implement sampling method for surrogate generation. The sampling efficiency benefits from:

- reduction in the total number of sampling points;
- reduction in the time per data generation.

The remainder of Chapter 5 is structured as follows: Section 5.2 proposes the overall workflow for the surrogate construction; Section 5.3 demonstrates the state-of-the-art of two principles for efficient data generation; Section 5.4 presents two case studies on chemical processes, followed by conclusions in the final section.

5.2 Workflow for surrogate construction

This section presents the workflow for surrogates generation. Latin Hypercube Sampling (LHS) is selected as the sampling technique, because it does not lose generality with the increase of dimensionality and can deliver a well-distributed sampling result [154]. As shown in Figure 5.3**,** the algorithm samples initial data by LHS. Then, simulations or experiments generate the corresponding outputs (similar procedure as Figure 4.2). With the initial data points (or together with a few iterations), an SVM classifier is trained to separate the feasible design space (with desired features) from the infeasible one. The data inputs from the infeasible region are deleted, while inputs in the feasible region are passed to the simulator for outputs. To fit multiple outputs simultaneously, ANN is selected as the surrogate type. In the successive iterations, data is sampled in batch by LHS for surrogate refinement, with which the sampling gradually slows down (the number of sampled data points gradually decreases with iterations).

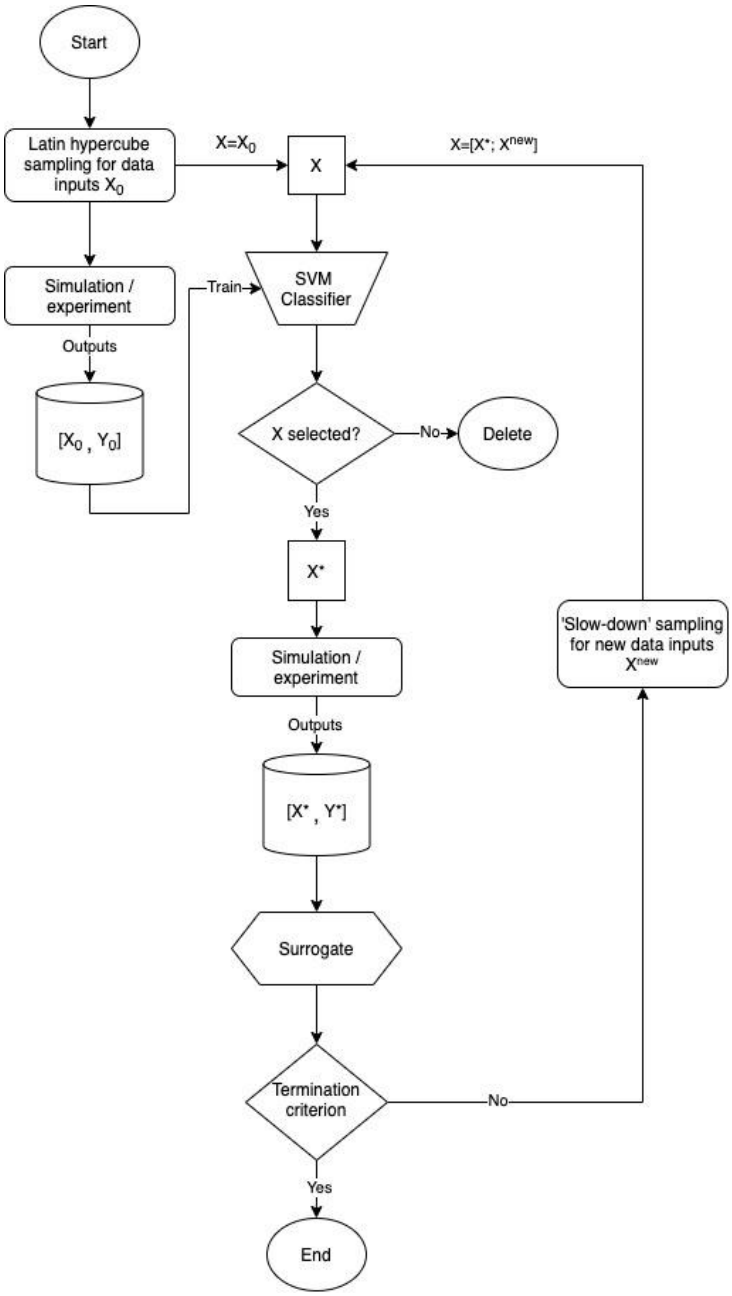


Figure 5.3. Proposed workflow for surrogate generation. [X_0, Y_0] are initial inputs/outputs to train an SVM classifier; [X^*, Y^*] are the inputs/outputs (selected by SVM classifier) for surrogate training in the latest iteration; X^{new} are the inputs for the next iteration; X are the updated inputs in the latest iteration. The added number of samples in iteration i refers to the sample rate (N_{added_i}) in iteration i.

At each sequential sampling iteration (i^{th} iteration), the workflow can generate a surrogate ($Surrogate_i$). The regression performance of $Surrogate_i$ is computed by a training-validation-test method. Specifically, the obtained dataset is divided into three subsets: training, validation, and test at a ratio 70% / 20% / 10%. Given the nonlinearity of process systems, a nonlinear activation function - hyperbolic tangent (tanh) – is used. I optimized the structure of networks by a random search strategy. In the random search strategy, a set of network candidates are established with random structures (*e.g.,* the number of layers and the number of neurons is different within the network candidates), and they are regressed by the training dataset. Following this, these trained network candidates are evaluated using validation dataset, and the network candidate with the minimal MSE value is selected as e_i. The regression performance of $Surrogate_i$ is determined by its MSE based on the test dataset.

5.3 State-of-the-art for efficient data generation

The two principles, the classifier and slowdown sampling, are detailed in this section.

5.3.1 Classifier SVM

The sampled points might fall in the infeasible design space due to extreme operating conditions for experiments (*e.g.,* unexpected reactions occur at high temperature) or non-converged recycle streams, or integration failure on stiff models during computational simulations. A classifier can be trained to pre-treat the data inputs. Only the selected data inputs can be passed into the simulation or experiment stage, thus saving the average time spent on a single data point.

Support vector machine (SVM) is a machine learning technique primarily for classification. SVM was initially proposed as a linear classifier, while Vapnik *et al.* expanded its application as a nonlinear classifier in 1995 [147]. SVM has been successfully applied in pattern recognition and computer vision problems [58]. Ibrahim has demonstrated its successful application in chemical process engineering [155]. A toolbox of SVM can be accessed in MATLAB, so SVM is selected as the classifier in this work. The training process for SVM is similar to the steps for surrogate training. Two differences are specified here. Firstly, only the dataset in the several initial iterations is used to train the classifier. This is because the classifier

in this work is expected to give a rough classification between infeasible and feasible design spaces, so the iterative refinement for the classifier is not necessary. Secondly, the output for the classifier is binary, 0 and 1: set 0 if the simulation outputs fall on the infeasible space, while set 1 if the simulation outputs fall on the feasible area. Following this, the data inputs together with the classifier outputs are used to train the SVM.

5.3.2 Slowdown sampling

To clearly explain the slowdown sampling strategy, I start with the definition of two variables as follows:

- Sampling rate (N_{added_i}): the number of new samples in i^{th} iteration.
- Surrogate improvement rate ($\left|slope_{\overline{MSE}_i}\right|$): the surrogate improvement per sample added.

5.3.2.1 Logic behind slowdown sampling

When employing sequential sampling based on a static sampling rate, a practical question falls on how to determine a proper value for the sampling rate. A large rate can result in the oversampling in the final iterations, while a low rate will lead to too many iterations, but the training in the early iterations is not meaningful based on a small dataset. Herein, this chapter sets up a dynamic sampling rate: initially, the sampling rate is relatively large as to achieve a reasonable data quantity for surrogate regression in just a few iterations; the sampling rate gradually slows down with the regression improvement of the iteratively refined surrogate. This refers to the slowdown sampling principle. To achieve this, I need to build the relationship between sampling rate and surrogate improvement rate.

First, I explain how to quantify the surrogate improvement rate ($\left|slope_{\overline{MSE}_i}\right|$) in i^{th} iteration. The first iteration obtains the result directly from the classifier section. For a successive iteration ($i \geq 2$), MSE_i is used to quantify the regression performance. I use the moving mean ($\overline{MSE}_i$) to smooth the fluctuation of the MSE curve. The MSE decrease per data added, or I call it the $slope_{MSE_i}$, is defined as Eq 5.1. Its absolute value can reflect on how the surrogate

can be refined based on one more data point, so $\left|slope_{MSE_i}\right|$ is suitable to express the surrogate improvement rate.

$$slope_{\overline{MSE}_i} = \frac{\overline{MSE}_i - \overline{MSE}_{i-1}}{N_{added_i}} \qquad \text{Eq 5.1}$$

Second, I propose how the sampling rate is expected to respond to the surrogate improvement rate. A large value of $\left|slope_{\overline{MSE}_i}\right|$ indicates that the addition of new samples can significantly improve the quality of the surrogate; hence, the sampling rate of the next iteration ($N_{added_{i+1}}$) is expected to be large; while a very small value of $\left|slope_{\overline{MSE}_i}\right|$ indicates that oversampling tends to occur, so $N_{added_{i+1}}$ should approach 0. In brief, the smaller $\left|slope_{\overline{MSE}_i}\right|$ is, the smaller $N_{added_{i+1}}$ is.

Third, I display the steps of relating the surrogate improvement rate ($\left|slope_{\overline{MSE}_i}\right|$) to the sampling rate ($N_{added_{i+1}}$).

- Step 1: $slope_{MSE_i}$ can be scaled to $slope_{relative_i}$ based on the initial slope value ($slope_{MSE_2}$), as Eq 5.2. This scaled slope value falls between -1 and 1 (due to the fluctuation, the value of the slope can be positive). The $slope_{MSE_2}$ normally has the largest absolute value among all the $slope_{\overline{MSE}_i}$, meaning the surrogate improvement rate is largest in the beginning. Hence, the absolute value of $slope_{relative_i}$ reflects how $\left|slope_{\overline{MSE}_i}\right|$ drops, when comparing to the largest value in the beginning.

$$slope_{relative_i} = \frac{slope_{\overline{MSE}_i}}{slope_{MSE_2}} \qquad \text{Eq 5.2}$$

- Step 2: this step aims to achieve 'the smaller $\left|slope_{MSE_i}\right|$ is, the smaller $N_{added_{i+1}}$ is.'. As shown in Eq 5.3, a ratio function can convert the scaled slope to a positive value as the added ratio ($added_{ratio_{i+1}}$, typically between 0 and 1). $added_{ratio_{i+1}}$ refers to the ratio of the sampling rate over the maximum sampling rate. The ratio function, $ratio_{function}$, can be formulated to make $added_{ratio_i}$ directly be proportional to $\left|slope_{relative_i}\right|$; to further decrease the sampling rate near the optimal surrogate, I used

a trigonometric-type function for the ratio function as shown in Eq 5.4. With the assistance of trigonometric-type ratio function, the sampling rate can drop extensively when the relative slope is approaching 0, compared to direct proportionality (Figure 5.4).

$$added_{ratio_{i+1}} = ratio_{function}(|slope_{relative_i}|) \quad \text{Eq 5.3}$$

$$ratio_{function}(|slope_{relative_i}|) = \begin{cases} -\mathrm{S} \cdot \cos\left(\frac{\pi}{2} \cdot |slope_{relative_i}|\right) + S & , |slope_{relative_i}| < 1 \\ S & , |slope_{relative_i}| \geq 1 \end{cases} \quad \text{Eq 5.4}$$

where S is a scaling factor, normally set as S=1.

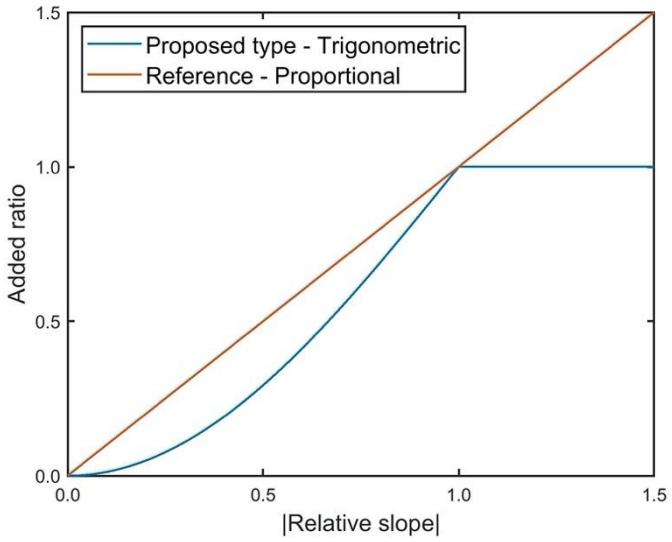


Figure 5.4. Added ratio function: trigonometric (S =1) vs. proportional type.

- Step 3: the sampling rate ($N_{added_{i+1}}$) is calculated through multiplying $added_{ratio_i}$ by the maximum sampling rate (the maximum number of new samples per iteration, N_{upper}), see Eq 5.5.

$$N_{added_{i+1}} = N_{upper} \cdot added_{ratio_{i+1}} \quad \text{Eq 5.5}$$

5.3.2.2 Demonstration of slowdown sampling by fitting kinetics for $A \overset{k_1}{\rightarrow} B \overset{k_2}{\rightarrow} C$

To better demonstrate the slowdown sampling, I use a simple example of fitting two reactions in series $A \overset{k_1}{\rightarrow} B \overset{k_2}{\rightarrow} C$. The two reactions are assumed to obey first-order kinetics, as written in Eq 5.6 - Eq 5.8. The true values of kinetic parameters are assumed at $k_1 = 0.42, k_2 = 0.97$, and the initial concentration is set as $[A]_0 = 1$.

$$\frac{dA}{dt} = -k_1 A \qquad \text{Eq 5.6}$$

$$\frac{dB}{dt} = k_1 A - k_2 B \qquad \text{Eq 5.7}$$

$$\frac{dC}{dt} = k_2 B \qquad \text{Eq 5.8}$$

Based on this physical model, the concentration profiles of the three species are simulated. An ANNs-based surrogate is iteratively refined by sequential sampling: the input is time; output is the concentration of species. As Figure 5.5a indicates, with more data added, the fitting performance improves (MSE decreases). Meanwhile, the decreasing rate of MSE becomes slower (Figure 5.5a) and the absolute value of relative slope tends to be smaller (Figure 5.5b). Following this, the added ratio decreases (Figure 5.5c) as well as the same trend is indicated for the sampling rate (Figure 5.5d). Once $\left|slope_{relative_i}\right| < 0.02$, the algorithm is terminated and collects 25 data points in total.

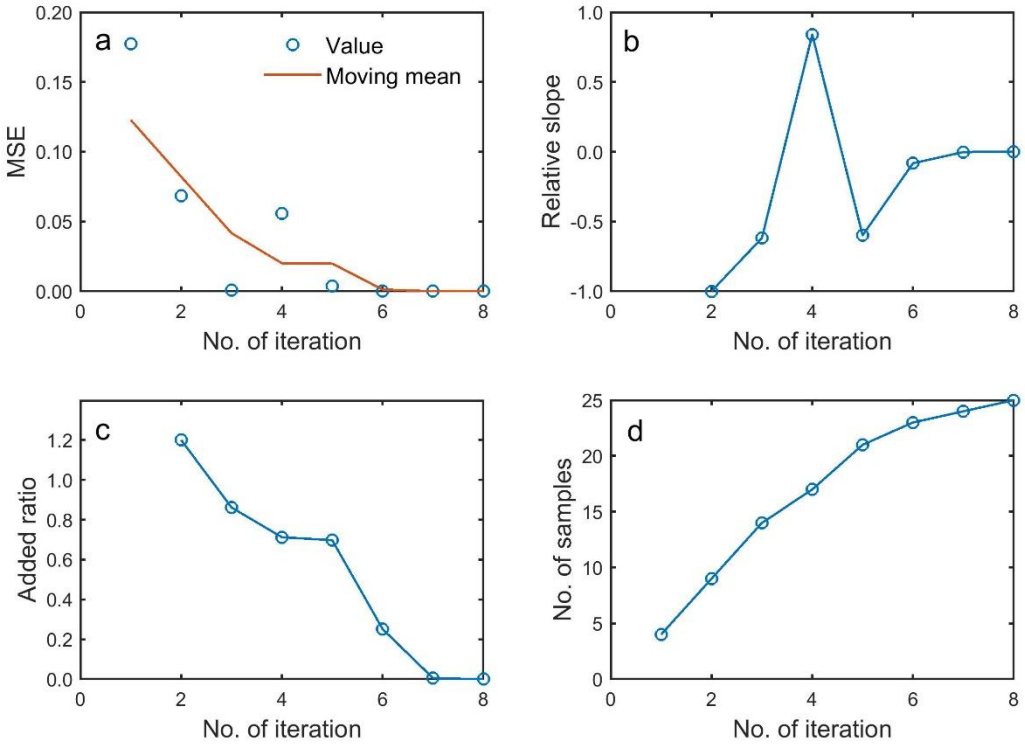


Figure 5.5. Slowdown strategy for the surrogate construction of series reaction kinetics.

To further evaluate the performance of the obtained surrogate, I simulate the concentration profiles of three species using physical model and surrogate, respectively. Figure 5.6 shows that the regression performance of the iteratively refined surrogate gradually improves with iteration. The surrogate obtained in the final iteration (iteration 8) can perfectly model the original concentration profiles of the three species.

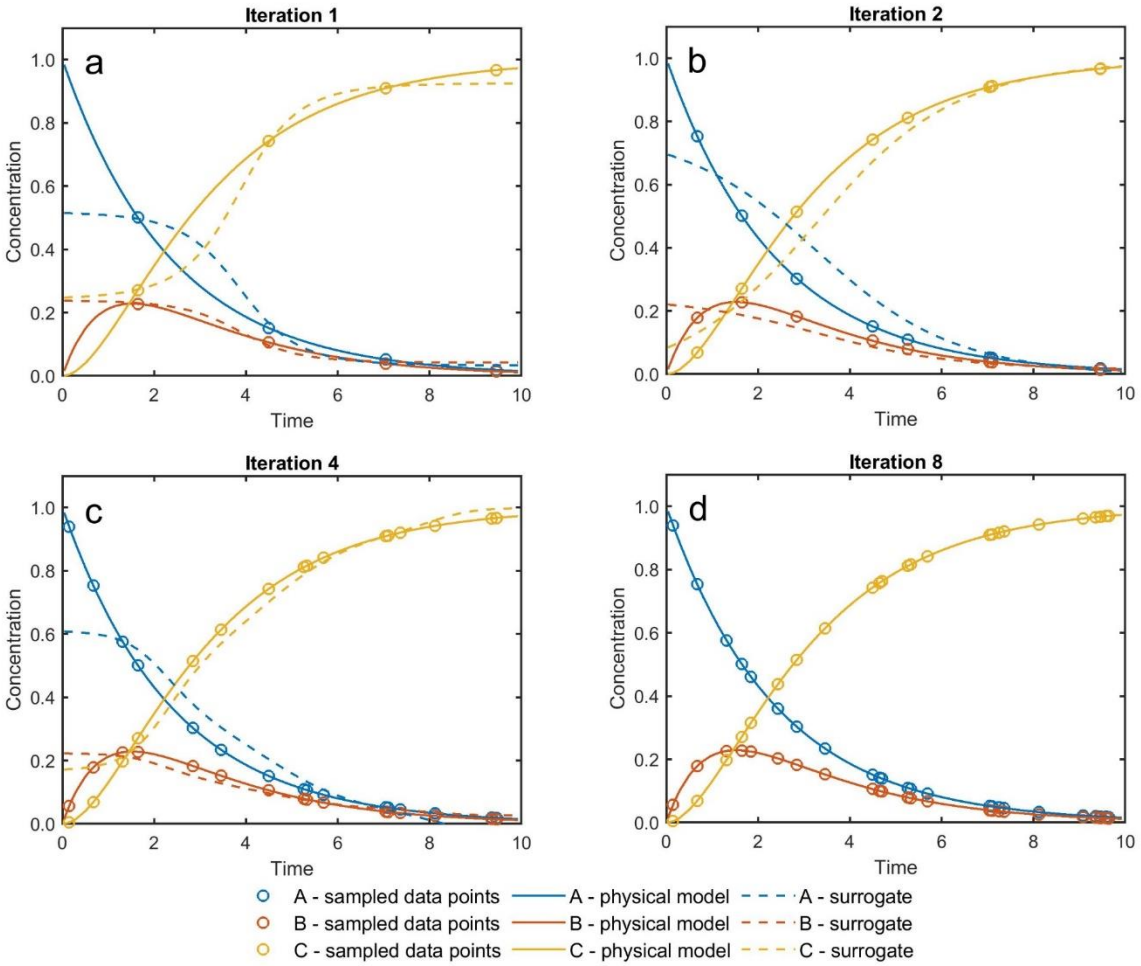


Figure 5.6. The regression performance of ANN surrogate for the concentration profiles of three species regarding the series reaction. Each iteration adds new data points to refine the surrogate: the input is time; output is the concentration of species. The performance of the surrogate gradually improves from iteration 1, 2, 4 to iteration 8. Solid lines for the simulation by the physical model, while dashed lines for the simulation by the surrogate model. Surrogate is built based on the sampled data points.

5.3.2.3 Discussion on slowdown sampling

The slowdown sampling maintains a good balance between training and sampling. In each iteration, a small number of networks are recommended to test. I consider two extremes:

(1) When data is sufficient, slowdown sampling tends to give a low rate. As a result, the number of total samples does not significantly change, while training is still performed in every iteration. This can be equivalent to an extreme situation, where sampling stops but excessive trainings are executed for sufficient data.

(2) In contrast, when the data is insufficient in the very initial iterations, slowdown sampling tends to deliver a large sampling rate, so fewer trainings but more samplings are executed in the initial iterations.

Such a balance between training and sampling is automatically built by relating the improvement rate of surrogate ($slope_{MSE_i}$) to the sampling rate (N_{added_i}). However, this balance advantage is not obvious in this work because I focus on the case studies, where the computational cost on data generation is much more expensive than surrogate training.

The number of total sampled data points required by the slowdown sampling can be well reproduced, which can be referred to an example of peaks function in Appx.C.3. Slowdown sampling is performed four times for the fitting the peaks function: the sampling trends are similar for the four times, and the number of total sampled data points are close to each other (between 190 and 220).

5.4 Case studies

Two case studies come from two processes in carbon capture and utilization (CCU): pressure swing adsorption (PSA), and Gas-to-Liquids (GTL), which starts from combined reforming (steam + CO_2) of natural gas.

5.4.1 Case study 1: surrogate generation for PSA

The physical model, dynamic simulation, and a typical input-output data point of PSA are presented in Appx.A.1. The complexity of PSA has been emphasized in Chapter 3. A surrogate can reduce the complexity, but I still need to minimize the computational cost of data generation to build the surrogate. As shown in Figure 5.7, I program the physical model of PSA in Dymola and use MATLAB to run Dymola to collect inputs/outputs dataset automatically [24]. Table 5.1 describes the inputs and outputs for the PSA system.

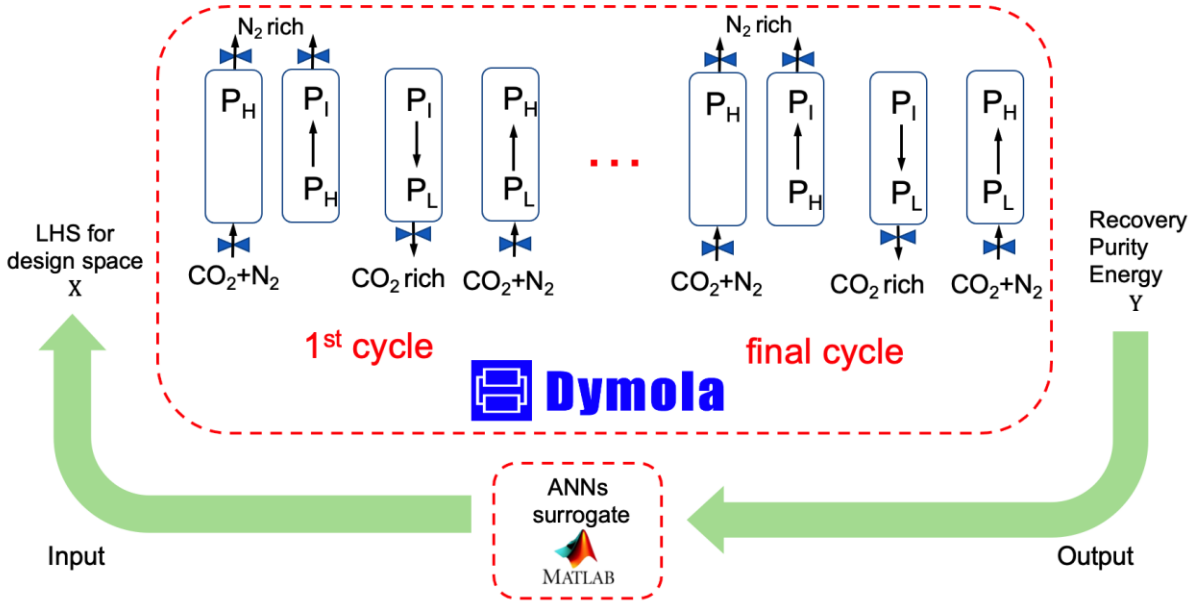


Figure 5.7. Surrogate construction of the four-stage PSA for CO_2 capture.

Table 5.1. Description of input and output variables for PSA surrogate.

Input variables	Range	Unit	Notes
t_{ads}	20-100	[s]	Duration of adsorption stage
t_{bd}	30-200	[s]	Duration of blowdown stage
t_{evac}	30-200	[s]	Duration of evacuation stage
P_I	0.07-0.5	[bar]	Setpoint of intermediate pressure
P_L	0.005-0.05	[bar]	Setpoint of low pressure
v_{feed}	0.1-2	[m/s]	Inlet flowrate
y_{CO_2}	0.02-0.06	[-]	Inlet molar fraction of CO_2 (an uncertainty)
Output variables			
Recovery		[-]	Recovery rate of CO_2
Purity		[-]	Purity of CO_2 in the product flow
Energy		[kWh/ton-CO_2]	Energy usage per ton CO_2 captured

In this case, PSA is applied to capture CO_2 from the flue gas of a natural gas power plant. Due to the low CO_2 concentration in the flue gas (~ 4%), one PSA unit cannot guarantee the required purity (I use GA to optimize one PSA unit, but it cannot achieve the purity of CO_2 in the product

flow over 90%). PSA in series can be an option (see Appx.6.3.1.2). In this chapter, I mainly focus on the performance of the first PSA unit, where the recovery of CO_2 is supposed to be high enough. The purity of CO_2 should improve as well. A trade-off relationship is reported between recovery and purity [23, 24], so the CO_2 purity cannot be too high given the priority on recovery. Therefore, I trained an SVM classifier to select the sample inputs, which are predicted to achieve a high recovery (higher part, >50%) and a moderate purity (middle part, 25%-75%). The classifier's performance can be referred to in Figure 5.8, and eventually, only 24% of the initial-sampled data is selected to fall in the desired space.

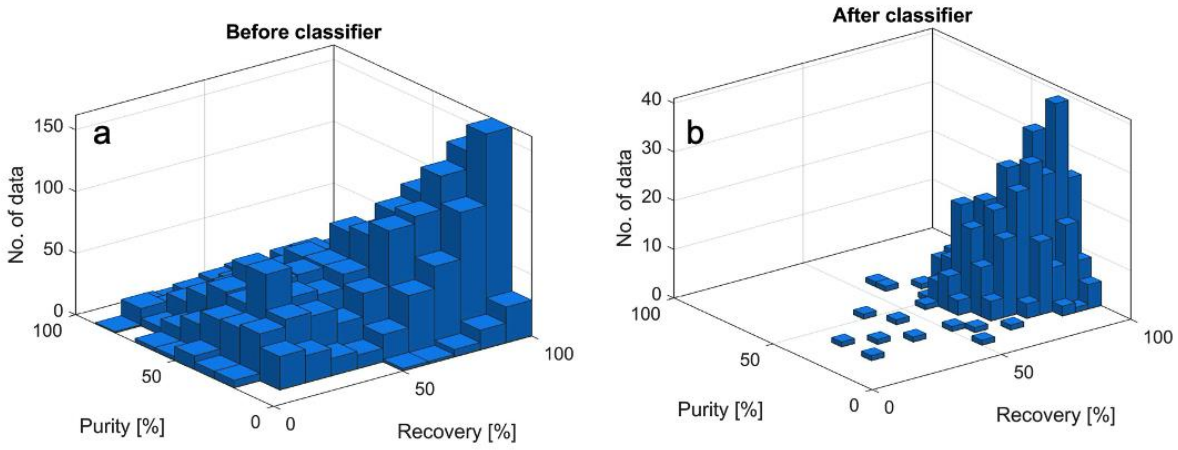


Figure 5.8. Classification performance for PSA.

The performance of the SVM classifier looks imperfect, since some selected sample inputs still lead to undesired outputs. For example, a few filtered samples have a recovery smaller than 50% (Figure 5.8b). Yet, this result is good-enough when referring to the prediction accuracy by the classifier (Figure 5.9). Raw data, containing effective data (desired) and ineffective data (undesired), is used to train SVM classifier. The training-test split is used to calculate the accuracy of prediction: 90% raw data is used to train the SVM classifier, while the other 10% raw data is used to examine its prediction performance. As shown in Figure 5.9, increasing the number of raw data can improve prediction accuracy to 90%. With more than 2,000 data points, I see negligible improvement until 4,000 data points. Since 90% accuracy is already good enough for a classifier, I stop sampling ineffective inputs/outputs for SVM training. That is to say, after 4,000 raw data points, the SVM classifier commences its filtering function for the newly sampled inputs. To clarify the relationship between the SVM classifier and the

slowdown sampling, 4,000 raw data points (24% desired) only contain 960 effective data points, which initializes the first iteration of the slowdown sampling.

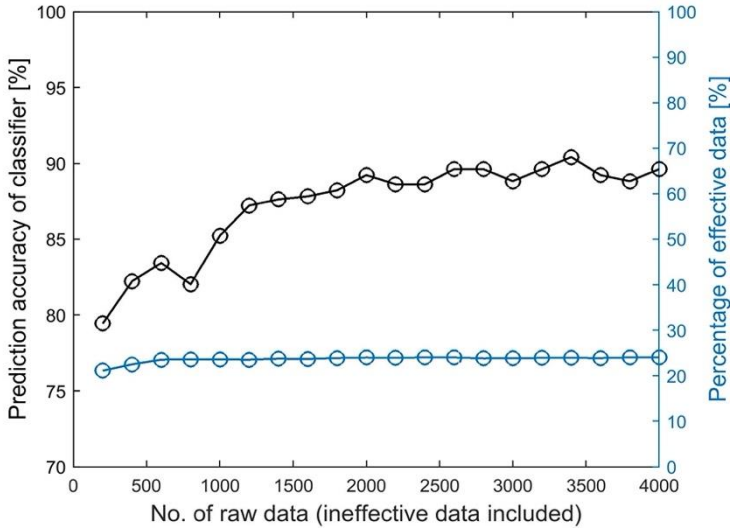


Figure 5.9. Effect of the number of raw data points on the classification performance for PSA. [Note: raw data includes both effective data and ineffective data. Given the 24% effective data, 4,000 raw data points result in 960 effective data points, which corresponds to the first iteration in the slowdown sampling].

The slowdown sampling is applied to collect effective data iteratively. Figure 5.10 indicates that the regression improvement is not significant after 10 iterations (Figure 5.10a), and the corresponding sampling rate gradually decreases in the meanwhile (Figure 5.10d). The relative slope fluctuates significantly along with the iteration (Figure 5.10b), while the added ratio function helps smooth the fluctuation (Figure 5.10c). Eventually, I terminated the algorithm after 50 iterations, since the MSE value hardly decreases after the 40^{th} iteration.

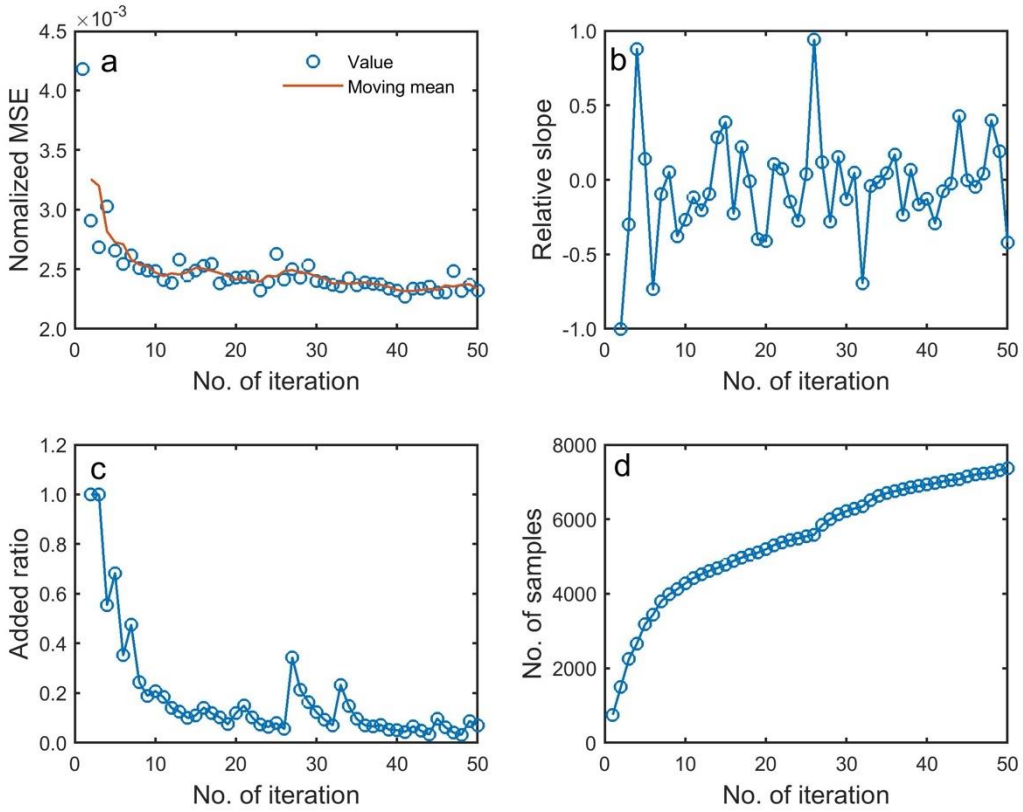


Figure 5.10. Slowdown sampling principle for the sequential sampling of PSA.

The efficiency of the proposed workflow can be demonstrated by comparing it with a reference method with no classifier and with a static sampling rate (a slow static sampling rate is applied at 200 data points per iteration). Although the initial rate of slowdown sampling is over three times that of equal sampling, the sampling rate keeps dropping and, eventually, falls below 50 data/iteration after the 40th iteration. Within 50 iterations, slowdown sampling generates 7,372 samples, while equal sampling generates 10,000 data points. Notably, Figure 5.11a indicates that slowdown sampling has a much higher possibility for earlier termination. When a similar fitting performance is reached (*e.g.,* MSE = 2.2E-3), much fewer data points are collected by the slowdown sampling (6,967 data points) than by equally sampling (8,800 data points). Figure 5.11b illustrates the effect of the SVM classifier. The classier is trained by the dataset in the first iteration. The average time per data generated is assumed to be kept the same as the 1st iteration if no classifier applies (as the dashed line in Figure 5.11b). Herein, the data effectiveness without a classifier is around 24%. By contrast, the classifier can significantly

improve the data quality by avoiding undesired inputs for the data generation, thus reducing the time per data point generation by 83%, from 375 s (1st iteration) to 65 s (50th iteration).

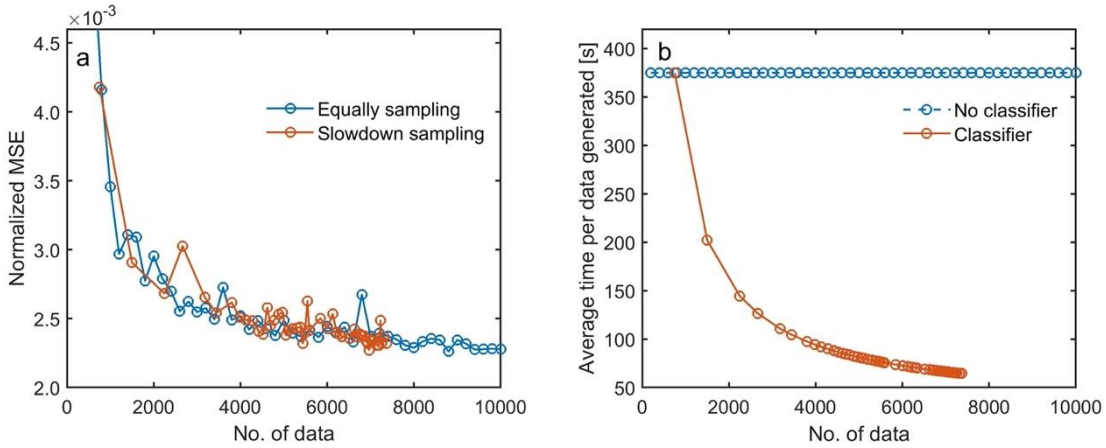


Figure 5.11. The contribution of slowdown sampling and classifier for the efficiency improvement of PSA surrogate construction: (a) slowdown sampling has a higher chance to spot non-improvement trend than equal sampling; (b) a classifier reduces the time per data point generation.

The effect of two principles (slowdown sampling and the classifier) can be merged to improve the efficiency of surrogate generation for PSA. Since fluctuations exist through sequential iterations, a termination is hard to be determined. Herein, I terminated the algorithms after 50 iterations. As shown in Figure 5.12, if equally sampling without classifier is applied, the time spent on surrogate generation for PSA is 3.8E+6 s (50th iteration), which can be reduced by 87% if the two principles apply (4.8E+5 s, 50th iteration). It might be unfair to compare slowdown sampling with equal sampling based on the number of iterations, because the performance of equal sampling can be different when the sampling rate for equal sampling changes. A reasonable comparison criterion can be based on a key iteration, which identifies the surrogate with the best regression performance. Based on the found minimal MSE = 2.2E-3, the [slowdown sampling + classifier] requires 4.7E+5 s (6,967 data points, 41st iteration), while [equally sampling + no classifier] requires 3.3E+6 s (8,800 data points, 44th iteration). Hence, the proposed workflow can reduce the total time by 86%.

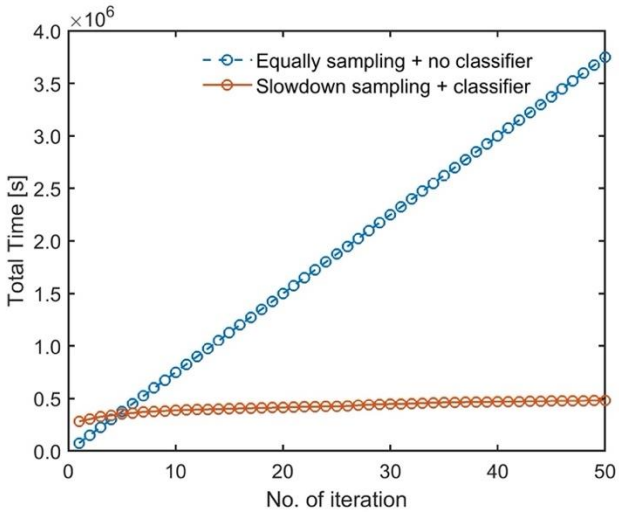


Figure 5.12. Comparison of total time spent on surrogate generation for PSA between [equally sampling + no classifier] and [slowdown sampling + classifier]. Total time is the sum of time spent on data generation and surrogate training.

A separate dataset was used to test the performance of the iteratively refined surrogate for PSA. I employed the boxplot for the relative errors between the surrogate predictions and the rigorous simulations for the three outputs – recovery of CO_2, purity of CO_2 in the product flow, and energy consumption of the system. As shown in Figure 5.13, most outputs can be predicted with relative errors smaller than 5%.

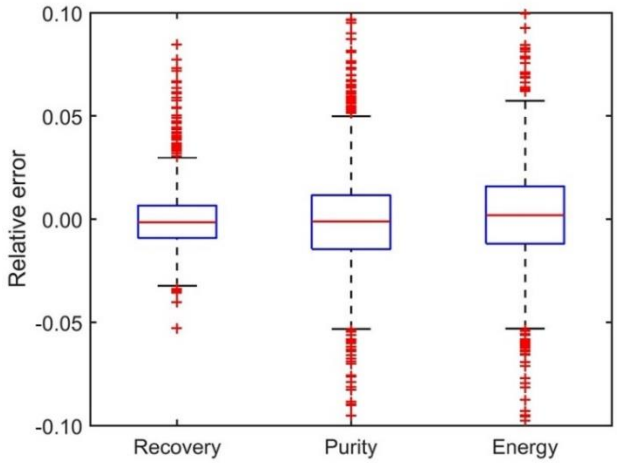


Figure 5.13. Prediction performance of the final surrogate for PSA.

5.4.2 Case study 2: surrogate generation for GTL

Gas-to-Liquids (GLT) is a classical chemical process for fuels production [102, 103]. I built a flowsheet in Aspen Plus (detailed information in Appx. A.3). As shown in Figure 5.14, GTL starts with the combined reforming (steam + CO_2) of NG to syngas, followed by Fischer-Tropsch (FT) for fuels. A recycle stream is split: one for reforming, the other for FT rector.

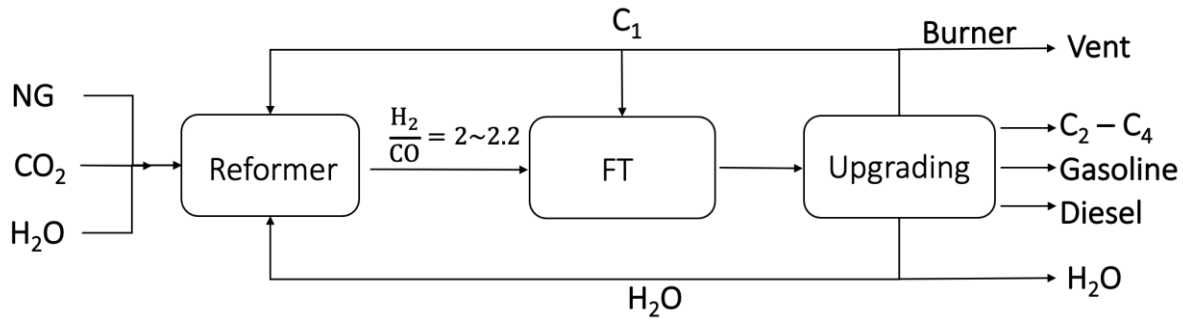


Figure 5.14. A simplified flow diagram for GTL. The corresponding Aspen plus model can be referred to Figure S4.

The combined reforming, FT (kinetics and chain growth probability for products distribution), simulation (convergence) and a typical input-output data point of GTL system are presented in Appx.A.3. To seek the optimal operating condition, we may optimize some decision variables under uncertainty (input for surrogate) to evaluate the corresponding process performance (output for surrogate), as shown in Table 5.2.

Table 5.2. Description of input and output variables for the surrogate building of GTL.

Input variables	Range	Unit	Notes
F_{CO_2}	72-8200	[kmol/h]	Inlet flowrate of CO_2
F_{NG}	*design spec.	[kmol/h]	Inlet flowrate of natural gas (NG)
x_{CH_4}	0.94-0.96	[-]	Inlet molar fraction of CH_4 (uncertainty)
T_{FT}	215-265	[℃]	Temperature in FT reactor
P_{FT}	15-50	[bar]	Pressure in FT reactor
N_{trays} (integer)	45-65	[-]	No. of trays in distillation column
$T_{reformer}$	750-1000	[℃]	Temperature in reformer reactor
$P_{reformer}$	3-7	[bar]	Pressure in reformer reactor
$Split_{vent}$	0.001-0.2	[-]	Split fraction to vent stream (the other to recycle)
$Split_{FT}$	0.01-0.99	[-]	Split fraction to FT (the other to reformer)
Output variables			
$F_{gasoline}$		[kmol/h]	Product flowrate of gasoline
F_{diesel}		[kmol/h]	Product flowrate of diesel
F_{gas}		[kmol/h]	Product flowrate of light HCs [C_1-C_4]
$F_{H_2O_{net}}$		[kmol/h]	Net flowrate of process water
$vent_{CO_2}$		[kmol/h]	Flowrate of CO_2 in the vent
Electricity		[GJ/h]	Electricity usage for pumps and compressors
U_{air}		[GJ/h]	Cooling utility by air
U_{1000}		[GJ/h]	Heating utility by 1000 ℃ fuel gas
U_{Steam}		[GJ/h]	Heating utility by high-pressure steam
U_{water}		[GJ/h]	Cooling utility by cooling water

* In Aspen Plus, F_{NG} is determined by a flowsheet option (design specification): the value of F_{NG} and steam flowrate are varied to achieve the desired syngas ratio for FT (H_2:CO = 2.0-2.2). Two surrogates are developed here: the first surrogate establishes the relationship between $[\ F_{CO_2}, x_{CH_4}, T_{FT}, P_{FT}, N_{trays}, T_{reformer}, P_{reformer}, Split_{vent}, Split_{FT}\ ]$ and F_{NG} ; the second surrogate establishes the relationship between input variables and output variables in Table 5.2. Section 5.4.2 is focused on the generation of the second surrogate.

Aspen simulations suffer from non-convergence issues when improper operating conditions are given, or the recycle stream is set too tight [155, 161]. Such problems also occur in our

case. An SVM classifier is employed to avoid non-convergence issues: through varying the values of input variables, check the simulation status (1 for convergence; 0 for non-convergence); then an SVM classifier is trained. As shown in Figure 5.15, 200 raw data can deliver a good classifier with an accuracy at 87%, which can further increase to 91% with 2,000 raw data points. After the 2,000 raw data points, the SVM classifier commences its filtering function for the newly sampled inputs. To clarify the relationship between SVM classifier and the slowdown sampling, 1,000 raw data (82% desired) and 2,000 raw data (81% desired) correspond to the first two iterations respectively in slowdown sampling.

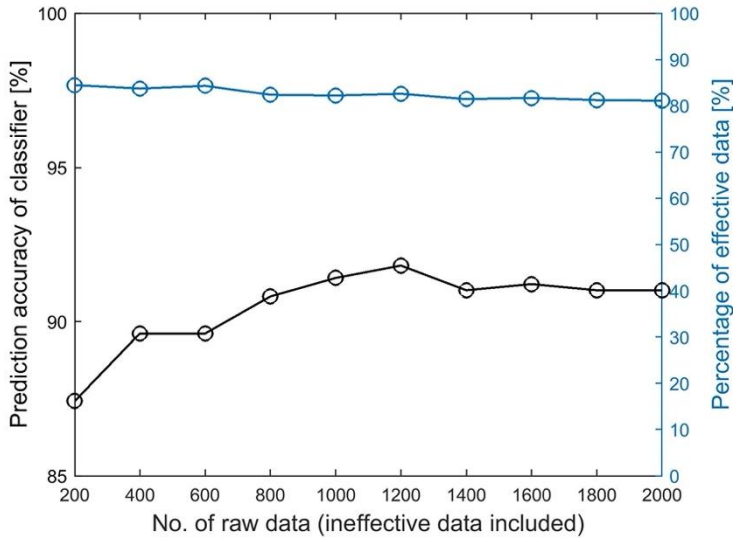


Figure 5.15. Effect of raw data number on the classification performance for GTL. [Note: 1,000 raw data points (82% desired) and 2,000 raw data points (81% desired) correspond to the first two iterations respectively in slowdown sampling.]

The obtained classifier can predict whether an input would deliver a converged simulation or not. As such, the classifier is capable to screen out some potential non-converged inputs in the successive iterations, thus improving the percentage of effective data from 81% to 91% (Figure 5.16). Notably, the non-converged simulations in Aspen Plus usually take a long time to stop but deliver invalid outputs. A 10% improvement for effective data tremendously cut the time per data generation by 46%, from 61 s (1~2 iterations) to 33 s (40^{th} iteration).

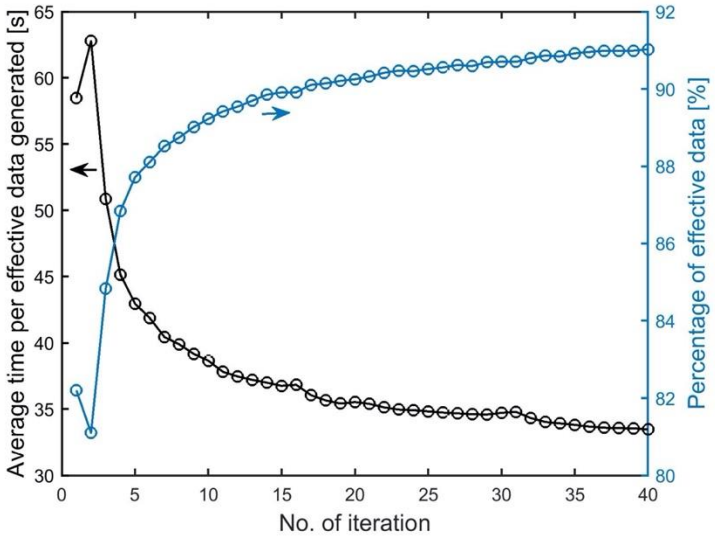


Figure 5.16. Improvement of data effectiveness by the classifier.

The slowdown sampling is applied to collect data iteratively. The relative slope in Figure 5.17b fluctuates more significantly than the case study of PSA, see Figure 5.10. That is probably because the GTL has more outputs to fit, and the regression is more complex than the case of PSA. The observed trend in Figure 5.17a indicates that the regression improvement is not significant after the 25th iteration. Eventually, I terminated the workflow after 40 iterations to avoid unnecessary computational costs.

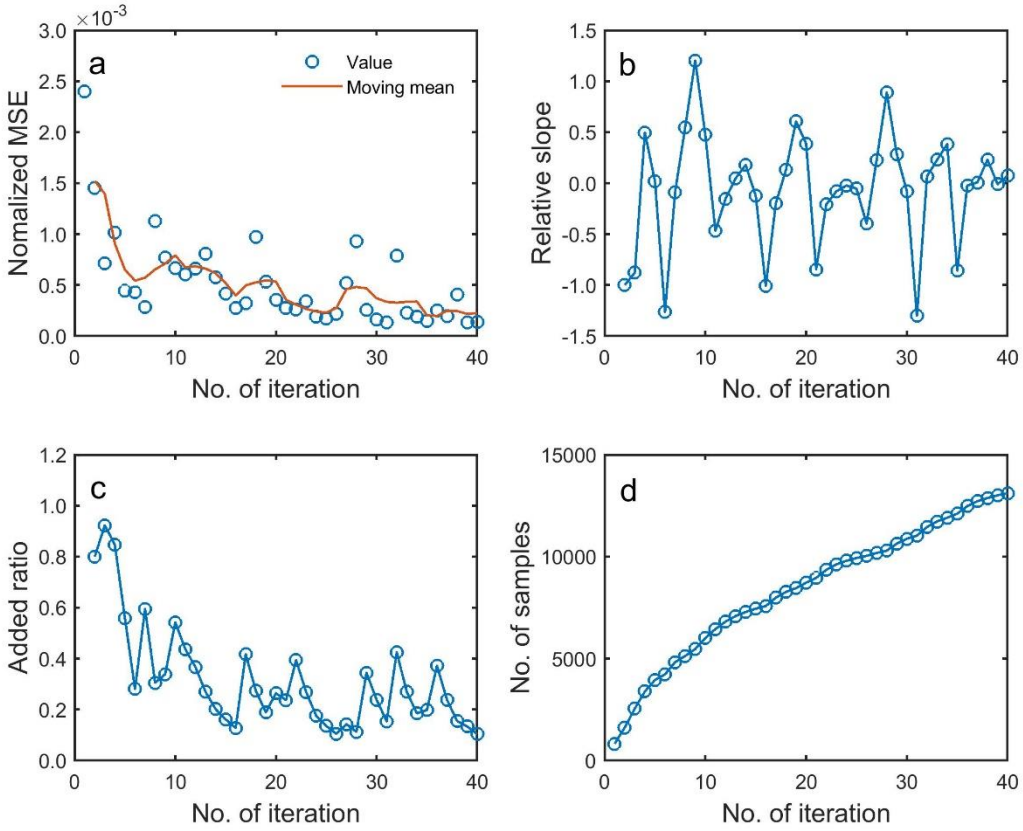


Figure 5.17. Slowdown principle for the sequential sampling of GTL.

The efficiency of the proposed workflow can be demonstrated by comparison to a reference method with no classifier and equal sampling (a slow static sampling). The two principles can separately improve the sampling efficiency for building surrogates for GTL. As shown in Figure 5.18a, the slowdown sampling has a higher chance for an earlier termination than the equally-sampling, to achieve a similar fitting performance (MSE=1E-4) with fewer data points (slowdown for 11000 data points vs. equally sampling for 13200 data points). The trend in Figure 5.18b shows that the SVM classifier is able to reduce the average time spent on individual points by 46%.

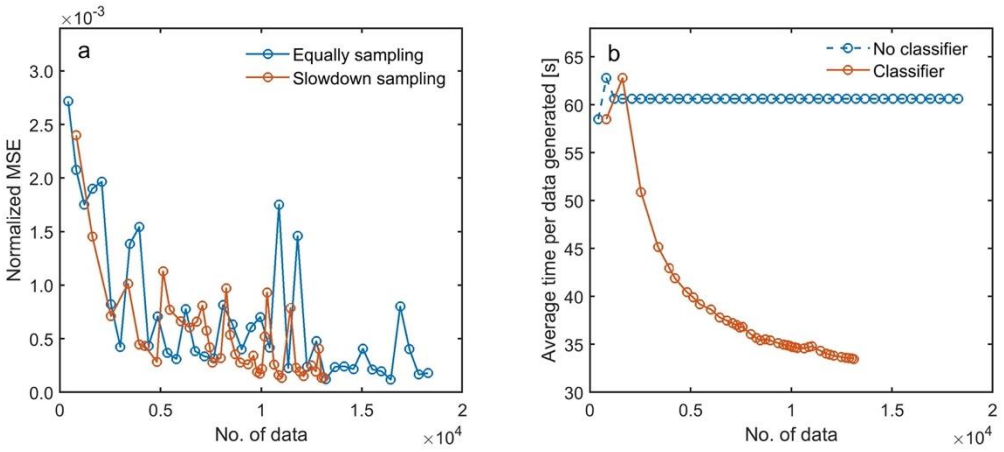


Figure 5.18. The contribution of slowdown sampling and classifier for the efficiency improvement of GTL surrogate construction: (a) slowdown sampling has much a higher possibility of collecting fewer data points than equal sampling; (b) a classifier can reduce the average time per data generated. (Clarification for the dashed line in b: since no classifier is used in the first two iterations, I assume that their average value for a single data generation will be the time in the successive iterations.)

Overall, the effect of slowdown sampling and the classifier can be merged to improve the efficiency of surrogate generation for GTL. As shown in Figure 5.19, based on the found minimal MSE = 1E-4, the [slowdown sampling + classifier] requires 3.9E+6 s (11,000 data points, 31st iteration), while [equally sampling + no classifier] requires 8.0E+6 s (13,200 data points, 29th iteration). Hence, the proposed workflow can reduce the total time by 51%.

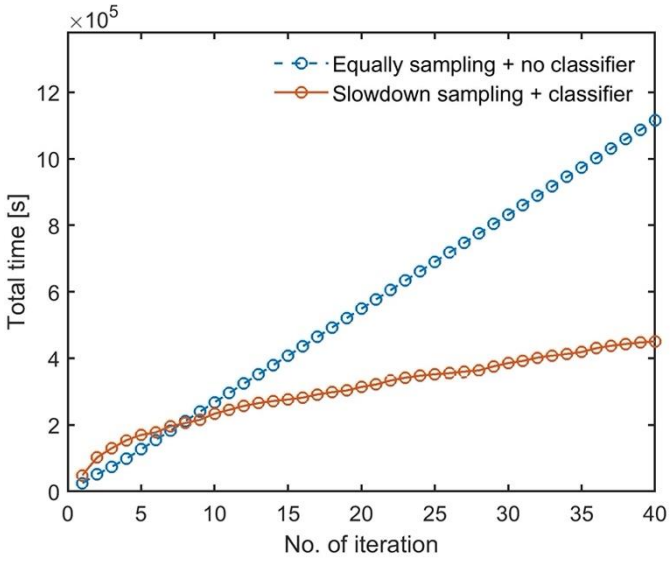


Figure 5.19. Comparison of total time spent on surrogate generation for GTL between [equally sampling + no classifier] and [slowdown sampling + classifier]. Total time is the sum of time spent on data generation and surrogate training.

A separate test dataset is used to evaluate the performance of the surrogate obtained in the final iteration. I employ the boxplot for the relative errors between the surrogate predictions and the rigorous simulations for the 10 outputs. As shown in Figure 5.20, most outputs can be well predicted with relative errors smaller than 5%, and some are even smaller than 1%, *e.g.*, the mass flowrate for the fuel products. The fitting for the utility is not ideal, and the relative error of the electricity consumption can go up to 15%. This is probably due to the insufficient feature selection for utility fitting. For example, the electricity consumption is related to the units of pumps and compressors, while no relevant features are taken into account as the input variables for the surrogate training. Meanwhile, no features related to heat exchangers are chosen, so the fitting performance of the utility is not as good as the mass flowrates. However, the motivation behind surrogate is to build a reduced-order model to replace the original full-order physical model, and thus sacrificing partial accuracy is unavoidable but acceptable.

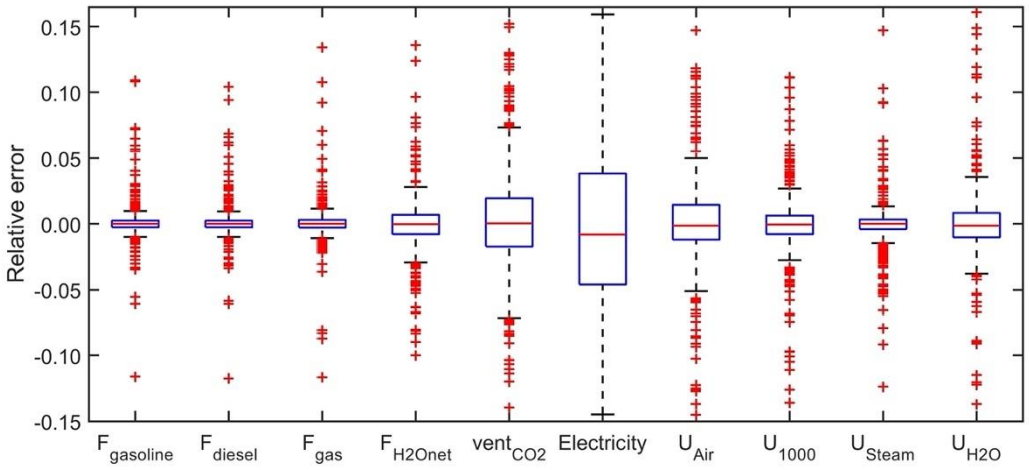


Figure 5.20. Prediction performance of the final surrogate for GTL.

5.5 Conclusions

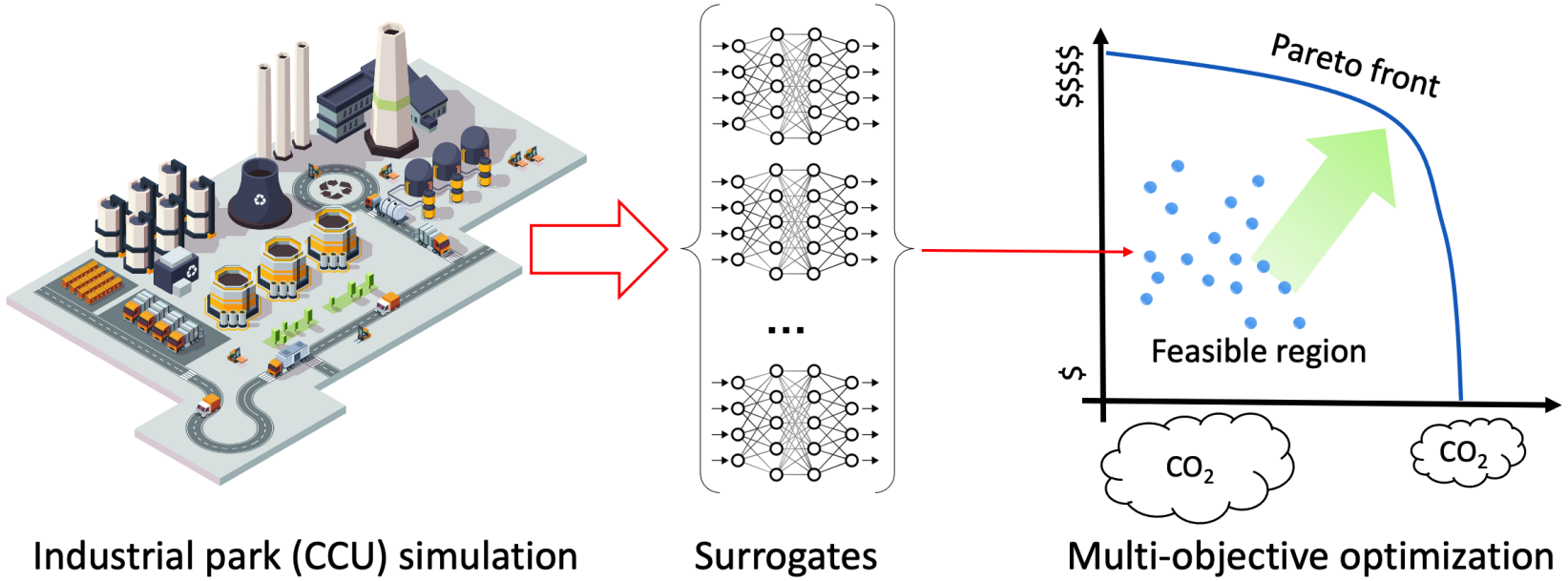
This chapter has developed an efficient workflow for the surrogate generation for engineering systems (typically $t_{data} \gg t_{training}$). The efficiency benefits from improved data quality and the reduction in data quantity. (1) A classifier is trained to avoid the undesired design space for data generation and improve the data quality. To train a good-enough classifier (over 90% accuracy) requires a relatively small amount of dataset, which can work as the data source for the initial iteration of slowdown sampling. The obtained SVM classifiers can dramatically cut the computational cost per data generation by 83% for PSA and 46% for GTL. (2) A slowdown sampling employs a dynamic sampling rate: initially, sampling is fast to collect a nearly sufficient amount of data in just a few iterations, and gradually slows down with the improvement of surrogate. The slowdown sampling can spot the non-improvement trend for the surrogate quality at a relatively early stage, thus lowering the possibility of oversampling (data quantity). With the proposed workflow, the computational costs of surrogate generation are shown to be reduced by 86% for PSA and 51% for GTL case studies, compared to that by employing a static sampling rate to achieve a similar standard of surrogate. Technically, our methodology is straightforward to implement because no intensive mathematical formulations are involved.

Surrogate training (hyperparameter tunning) is also important. In fact, slowdown sampling enables the exhaustive search for the hyperparameter tunning. In the future, the research can be focused on exploring the other types of activation functions and optimize the hyperparameters as to refine the ANN surrogates.

Notably, the proposed workflow can be generalized to other surrogate types, and it should be compatible with the other existing sampling methods. The exploitation-based methods can be introduced to integrate with our workflow, as to properly increase sampling probability in the nonlinear/complex design space. The primary goal of this work was to investigate the influence of the sampling rate for the surrogate generation. Thus, the sampling was desired to be a homogenous type, which might be disturbed by exploitation-based methods. As a result, I only considered exploration-based methods in our current workflow. Another work that can be done is to determine proper termination criteria: I tried to stop the algorithm when the MSE difference between two consecutive iterations approached 0, or the slope approached 0, but the fluctuation of MSE values always existed for the case study of GTL or PSA, which made the tolerance value for termination hard to set. One possible solution is to apply feature selection techniques [151, 152] (*i.e.*, automatically adjust input variables) to improve fitting performance and reduce the fluctuation during sequential sampling, but this is beyond the scope of this thesis.

Part II: Problem Solving

Part I has laid the foundation for the digitalization of CCU process options by surrogates. With these surrogates assembled in an interactive platform, **Part II** aims to optimize large CCU systems.



(Part of image elements come from VectorStock with the standard license EULA https://www.vectorstock.com/faq/member/standard-license)

Chapter 6 Optimization of a large CCU system

Net zero requires an accelerated transition from fossil fuels to renewables. Carbon capture and utilization (CCU) can be an effective intermediate solution for decarbonizing fossil fuels. However, many research works contain renewables in the design of CCU systems, which may mislead stakeholders regarding the hotspots of CCU systems. This chapter focuses on a 'worst condition' – designing a hypothesized industrial park containing power plants integrated with CCU, but no renewables are involved. I modelled this industrial park in process simulators. A three-level solution is applied to digitalize (the developed methodology from Chapter 4 and Chapter 5) and optimize the whole system simultaneously. I compare this industrial park with a conventional process (no carbon capture). The sources of GHG emissions are also examined. Single-objective optimization is performed to reduce GHG emissions, but such extreme conditions can significantly sacrifice the economic benefit. By contrast, multi-objective optimization can show how the decisions can affect the balance between GHG emissions and profit. Further, this work discusses the dual effect of carbon pricing on the CCU system – raising the cost of raw materials and utilities and gaining credits when emissions are reduced in producing valued products [162].

6.1 Introduction

Carbon capture is reported as both an effective and scalable technology to decarbonize the fossil fuels-based energy sectors [10]. Further conversion of captured CO_2 to high-value products (or 'utilization') requires an excessive amount of energy to break its chemical bonds, because CO_2 is thermodynamically highly stable. If the energy source is purely fossil fuels, carbon capture and utilization (CCU) is reported to cause more emissions than unabated fossil fuels [31-33]. To address this challenge, it has been proposed to apply renewable energy to power the carbon utilization, thus forming 'power-to-X' systems (power refers to solar or wind

renewable energy source; X refer to fuels or chemicals, such as methanol, H_2, gasoline and polymers) [12-14, 28, 31, 32, 163-166]. However, it is complex to immediately scale up these systems due to two facts: (1) a prerequisite is the access to cheap renewable energy, which requires a considerably higher renewable power capacity than todays' installations for solar photovoltaics [167] and wind turbines; (2) the intermittent renewable electricity requires either cheap battery systems or the feasibility for dynamic operation of the utilization processes [14, 28]. Further, several studies on hybrid systems, *i.e.* [CCU + renewable H_2/electricity], lead to a conclusion that the inclusion of renewable energy sources is indispensable to achieve emissions reduction [31, 32], and also the cost of renewables is considered to be the limiting factor for the economic feasibility of hybrid systems [27, 31, 33]. I anticipate that the involvement of renewables might underestimate the potential of CCU and mislead the hotspot identification for CCU itself.

I therefore sought to investigate whether CCU can be viable without the input of renewables. To answer this question, I created a hypothetical industrial park, where power plants are integrated with CCU, but no renewables are involved in the initial design. Following this, optimization is applied to explore the maximum potential of CCU regarding the environmental and economic aspects. The proposed strategy is inspired by the net-zero trends and prior works on CCU studies, which will be expanded in this section.

6.1.1 'Big picture' - energy transition trends urged by net zero

As shown in Section 2.2, there is a need for innovation that supports a stepwise transition from the current fossil-fuel-based energy production to the renewable-based future. [fossil fuels + carbon capture] may be an excellent intermediate solution to renewables. Additionally, I considered the electrification to enhance CCU as well as the influence of carbon price.

6.1.2 Prior works on CCU

Various pathways exist for either capture or utilization, thus making CCU a large and complex system. Optimization can accelerate the decision-making for the selection of pathways and the adjustment of operating conditions. Extensive studies had been done to optimize individual sub-systems of CCU, *e.g.*, pressure swing adsorption [18, 22-24] and chemisorption process

for CO_2 capture [25, 26], methanol synthesis [20, 27, 28] or Fischer-Tropsch[29, 30] for subsequent utilization. However, the performance of these sub-systems depends on each other, and thus individual optimal solutions cannot simultaneously co-exist. When a sub-system is optimized before extending to the whole CCU system, the decision space for other sub-systems is narrowed down, which may lead to a sub-optimal solution for the whole CCU system. In a recent review paper, Dieterich *et al.* also pointed out that the studies on the interaction between CCU sub-systems are still scarce [14]. Inspiringly, Roh *et al.* optimized a whole CCU system, where MEA is taken as the only CO_2 capture technology, and the 15 utilization pathways co-exist to satisfy market demands [32]. In Roh's work, the competitive interactions among different sub-systems are considered, but the complexity/non-linearity for individual sub-systems is neglected [32]. To manipulate both high-level system variables and sub-system variables, a more robust method is superstructure optimization [15], but this method leads to complex formulations and difficult-to-solve MINLP problems [40]. An alternative solution is surrogate-based optimization, where sub-systems can be represented by cheap-to-evaluate surrogates [40]. Still, most prior works limit the surrogate-based optimization to a CCU sub-system (either capture [18, 24, 128, 130, 168] or utilization [20, 27]).

Further, the development of CCU might be influenced by energy policy, *e.g.*, carbon pricing. I found very limited studies of carbon pricing specifically for the CCU, and the 'might' is used here because the latest studies indicate that the effectiveness of carbon pricing is uncertain in promoting low-carbon technologies. Green reviews that most studies of carbon pricing are focused on the EU, but criticized that carbon pricing has little impact on reducing GHG emissions [169]. Lilliestam *et al.* report that carbon pricing had an effect in accelerating the transition from coal to gas in the EU, but it has not promoted any other low-carbon technologies so far [170]. Daggash *et al.* present that penalty on emissions is not sufficient, but credits are required to support carbon-negative systems [171]. Barecka *et al.* show that a higher carbon price benefits the development of CO_2 utilization on the condition of the availability of low-cost electricity [163]. Additionally, Nicholson *et al.* predict that the rising carbon prices can raise energy cost [172].

Overall, previous works neither deliver a convincing evaluation of an impact of a CCU system in energy transition, nor address the complexity of optimizing a CCU system composed of different carbon capture and utilization technologies. To address both challenges, this work

focuses on an overseen scenario: CCU plants without renewable energy input considered in the initial design, and I develop a surrogate-based optimization methodology to assess its maximum potentials regarding emissions reduction and economic gain. Carbon pricing is included in the economic calculation to predict the future potential of this CCU system.

The remaining sections are structured as follows. Section 6.2 describes an industrial park where natural gas power plants are integrated with CCU. Section 6.3 illustrates the digitalization and optimization framework for the whole CCU system. Section 6.4 presents the single-objective optimization of maximizing the GHG reduction; this is set up to evaluate whether CCU can reduce CO_2 effectively, as well as to validate the overall optimization framework. Following this, multi-objective optimization is applied to the whole system concerning GHG reduction and economic gain in Section 6.5. Section 6.6 introduces carbon pricing within the economic evaluation. The final section presents conclusions.

6.2 Problem statement: an industrial park of power plants integrated with CCU

To explore the potential for decarbonization of energy and chemicals manufacturing by means of CCU, I sought to investigate all feasible process configurations that include well-understood, scalable process options for capture and utilization sections. I illustrate this approach with a case study of a hypothetical industrial park, which is powered by natural gas, and delivers electricity and liquid fuels as main products (Figure 6.1). In the reference case, where no carbon capture is deployed, all CO_2 emissions arising from electricity production are vented to the atmosphere. In the case of CCU, these CO_2 emissions will be captured and converted to fuels, thus reduce the input of petrochemical resources to the chemical synthesis and consequently decrease the carbon footprint of the industrial park. The industrial park is presumed to contain two 500 MW natural gas combined cycle (NGCC) power plants: one NGCC is equipped with Monoethanolamine (MEA) absorption process, while the other is coupled with PSA, as to capture CO_2. The CO_2 fraction of flue gas is concentrated from ~4 to ~90% by MEA and PSA, respectively. Following this, with the co-feed of NG and steam, the concentrated CO_2 is reformed to syngas, which is further converted to fuels, being reviewed as one of the most

promising product types for carbon utilization [73, 95]. Among different liquid fuels used on large scales, methanol, gasoline, and diesel are crucial for the mobility sector, because of their high energy density [14], and convenient handling. Hence, I focused on Fischer-Tropsch (FT) and methanol synthesis (MS) to manufacture fuel products (gasoline, diesel, and methanol). Overall, the proposed industrial park can be compatible with the existing industry in: (1) the upstream – by decarbonizing the energy sector, (2) the downstream – by supplying fuels to the mobility sector.

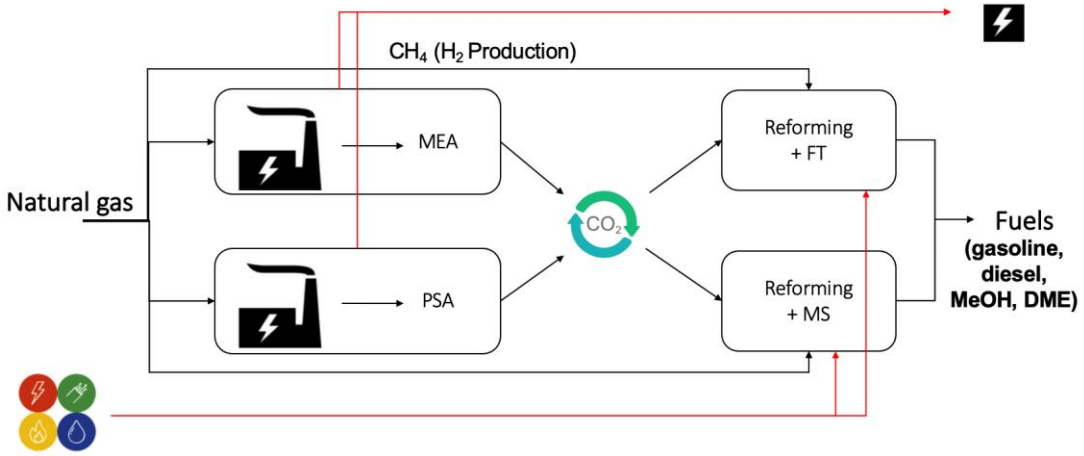


Figure 6.1. The hypothesized industrial park, where two 500 MW NG power plants are integrated with CCU. I call this industrial park as the whole CCU system, which contains four sub-systems: [NGCC+MEA], [NGCC+PSA], [Reforming + FT] and [Reforming + MS].

The scale of sequential CCU process is required to deal with the CO_2 in the flue gas from the two 500 MW power plants. Current work is an initial conceptual process design, which should allow more flexibility for the design space. The external constraints, e.g. market supply/demand, are not considered in this stage. With more information, the next stage can be the more robust design for the equipment size together with the consideration of external marketing influences.

The model of the industrial park is based on the following assumptions:

(1) The CO_2 captured by PSA is assumed to be temporarily stored in a collection hub, where CO_2 is well mixed before utilization. As such, the PSA performance under cyclic steady state [130] can be equivalent to steady-state. Additionally, the time scale of a PSA cycle (~10 min) [173] is much shorter than the start-up of chemical plants (~days)

[174]. Therefore, the overall system can be considered to operate under a steady-state condition.

(2) This PSA system contains two 4-step PSA columns in series to gradually improve the purity of CO_2 to 90% (Appx.6.3.2.2.2).

(3) Each NGCC power plant is equipped with a carbon capture technology, forming a sub-system.

(4) All of the captured CO_2 is mixed and then re-distributed to the downstream utilization pathways. The optimal ratio of CO: H_2 is slightly different between FT ($\frac{2CO}{H_2} = 1$) and MS ($\frac{2CO+3CO_2}{H_2} = 1$). This is because CO_2 can be a carbon source in MS (CO_2 is active on MS catalysts [175, 176]), while CO_2 is inactive on FT catalysts [108, 110]. Hence, CO_2 is distributed before reforming, which adjusts the desired ratio of CO/H_2 for FT and MS, respectively.

(5) Combined reforming is considered: dry reforming is considered to convert CO_2 to syngas, while steam reforming is also involved in adjusting the ratio between CO and H_2 in the final syngas. Both reforming processes can be assumed to achieve equilibrium [98, 99].

(6) A reforming process is closely connected to FT or MS, thus resulting in a single sub-system.

(7) The heating can be partially supplied with low-carbon electricity.

6.3 Three-level optimization framework

The scope of the optimization framework is designed around the entire industrial park, containing of four sub-systems, *i.e.*, [NGCC + MEA], [NGCC + PSA], [Reforming + FT] and [Reforming + MS]. To determine an optimal configuration, models of sub-systems are necessary. The current industrial practice involves the application of tailored simulators for specific systems (*e.g.*, Dymola for dynamic process modelling, Aspen for reactors and separation units). I anticipated that it might be insightful to search a global decision space by simultaneously optimizing all sub-systems, ideally from a level of a higher interactive platform. To achieve this goal, as well as improve the computational efficiency of the complex

optimization task, I digitalized the sub-systems using surrogates and proposed a three-level optimization framework shown in Figure 6.2.

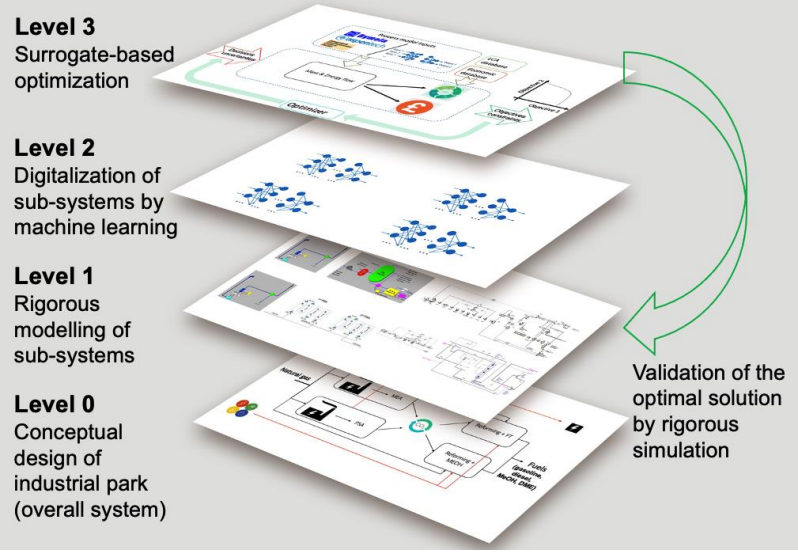


Figure 6.2. Three-level optimization framework for multi-process systems, illustrated by the case study of decarbonization of an integrated industrial park – two 500 MW power plants integrated with CCU.

6.3.1 Level 1: rigorous modelling of sub-systems

In Level 1, the sub-systems are modelled in different dedicated simulators. The two NGCC power plants and MEA absorption are represented in Integrated Environmental Control Model (IECM) [177]. The PSA is modelled in Dymola. Reforming integrated with FT/MS is modelled in Aspen Plus. The detailed information of simulators for the rigorous modelling of the individual sub-systems are given in this section.

6.3.1.1 [NGCC + MEA]

The natural gas combined cycle (NGCC) power plant is simulated in the IECM platform [177]. Wet cooling water is selected as the cooling system. Based on the simulation, I can obtain the consumptions of raw materials (NG, water), emissions per unit of electricity generated and CO_2 concentration in the flue gas.

An MEA absorption process is set up in IECM platform [177]. In IECM, a power plant equipped with carbon capture (NGCC-MEA) can be simulated. Since MEA absorption process

is a mature technology and its operating condition has been well optimized, I used the default values for operating condition in the IECM simulator. Based on the simulation, I can obtain the consumptions of raw materials (NG, water, MEA) and emissions per net electricity generated.

6.3.1.2 [NGCC + PSA]

NGCC power plant is simulated in the IECM platform [177]. PSA is equipped to reduce the GHG emissions. A more detailed description on PSA process (balance equations, boundary conditions, cyclic steady state) can be referred to Section 3.2. After my initial trial, one PSA unit cannot guarantee the required purity (90% for carbon capture), due to the low CO_2 concentration in the flue gas (~ 4%). Here, I propose to use two PSA in series to gradually improve the purity of CO_2, see Figure 6.3. As to the simulation of two PSA in series, MATLAB is used to run the Dymola to generate the simulation output of 1st PSA; the purity of CO_2 of 1st PSA is the CO_2% of 2nd PSA, then MATLAB runs Dymola to generate the simulation output of 2nd PSA. A trade-off relationship is reported between recovery and purity [23, 24]. To maintain a relatively high recovery, the first PSA aims to increase the CO_2 purity to 20%~50% (ranging from 25% to 75% among the CO_2 purity distribution), while the second one PSA further improve the CO_2 purity over 90%.

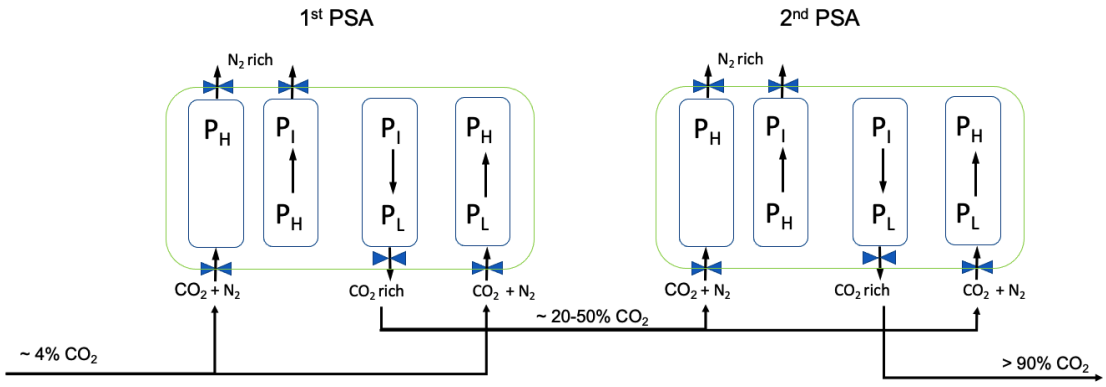


Figure 6.3. Two PSA in series.

6.3.1.3 [Reforming + FT]

This section is exactly the same as the GTL in Appx. A.3.

6.3.1.4 [Reforming + MS]

The [Reforming + MS] sub-system contains two sections: combined reforming and methanol synthesis. The combined reforming is quite similar as the reforming section in GTL, and the MS process model is taken out from the Aspen Plus model library [178]. More details can be referred to Appx. A.4.

6.3.2 Level 2: digitalization of sub-systems by surrogates

In Level 2, surrogates are established to replace the rigorous simulations for sub-systems for the overall optimization goal. This work mainly considers ANNs as surrogates, with the methodology developed in Chapter 4 and Chapter 5. Each sub-system can have one or two surrogates. For example, [Reforming + FT] sub-system contains only one surrogate, while the [NGCC + PSA] sub-system contains two surrogates for the two PSA in series. The detailed methodology for surrogate construction can be referred to Section 5.4, where I present how to build surrogates for the PSA and [reforming + FT]. The paramount step to generate surrogates is identifying the essential input/output variables, which is closely related to the optimization of the whole CCU system. Table 6.1 summarizes the decision variables. Design space of the decision variables is randomly sampled to generate sufficient input values, which are sent to the simulators in Level 1 for the corresponding outputs *via* rigorous simulations.

Table 6.1. Design variables (θ) for the model of the industrial park and their lower (LB) and upper bounds (UB) considered during optimization.

	Design variables (θ)	Unit	[LB,UB]	Ref	Definition
θ_{MEA}	r_{CO_2}	-	[0.60, 0.95]	[144]	Recovery rate of CO_2
θ_{PSA1}	P_{L1}	bar	[0.005, 0.05]	[23]	Low-pressure setpoint
	P_{I1}	bar	[0.07, 0.5]	[23]	Intermediate-pressure setpoint
	v_{feed1}	m s^{-1}	[0.1, 2]	[23]	Velocity of inlet flow
	t_{ads1}	s	[20, 100]	[23]	Duration of adsorption
	t_{bd1}	s	[30, 200]	[23]	Duration of blowdown
	t_{evac1}	s	[30, 200]	[23]	Duration of evacuation
θ_{PSA2}	P_{L2}	bar	[0.005, 0.05]	[23]	Low-pressure setpoint
	P_{I2}	bar	[0.07, 0.5]	[23]	Intermediate-pressure setpoint
	v_{feed2}	m s^{-1}	[0.1, 2]	[23]	Velocity of inlet flow
	t_{ads2}	s	[20, 100]	[23]	Duration of adsorption
	t_{bd2}	s	[30, 200]	[23]	Duration of blowdown
	t_{evac2}	s	[30, 200]	[23]	Duration of evacuation
CO_2 to FT	z_{FT}	-	[0.025, 0.975]		Splitting between FT and MS
θ_{FT}	T_{FT}	℃	[215, 265]	[29]	Reaction temperature for FT
	P_{FT}	bar	[15,50]	[112, 179]	Reaction pressure for FT
	$tray_{FT}$	-	[45, 65]		Tray no. of distillation column
	T_{ref1}	℃	[750, 1000]	[109]	Reformer temperature
	P_{ref1}	bar	[3, 7]	[109]	Reformer pressure
	S_{purge}	-	[0.001, 0.2]		Fraction for purge (recycle)
	Re_{FT}	-	[0.01,0.99]		Fraction for FT (reformer)
θ_{MS}	F_{NG}/F_{CO}	-	[2, 3.7]	[113]	Ratio of NG over CO_2
	T_{MS}	℃	[180, 220]	[113]	Reaction temperature for MS
	P_{MS}	bar	[50, 80]	[113]	Reaction pressure for MS
	$Tray_{MS}$	-	[45, 65]	[178]	Tray No. of distillation column
	T_{ref2}	℃	[800, 1000]	[109]	Reformer temperature
	P_{ref2}	bar	[3, 7]	[109]	Reformer pressure
Heating utility	$Frac_{fuel_{ele-CCS}}$	-	[0, 1]		Fraction of fuel heating substituted by CCS electricity
	$Frac_{steam_{ele-CCS}}$	-	[0, 1]		Fraction of steam heating substituted by CCS electricity

To identify input/output variables for individual surrogates, I use a top-down system thinking approach: (1) the decision variables and optimization objectives are the key input/output variables of the whole CCU system; the input should also include uncertainties, *e.g.*, concertation of methane in NG or carbon price; (2) the input/output variables of whole CCU system determines those for sub-systems, which can be referred to Section 6.3.2.1; (3) the input/outputs variables of a sub-system determines those for surrogates (Section 6.3.2.2). Eventually, the obtained input/output data points can train ANN-based surrogates.

6.3.2.1 Overview of essential input/output for sub-systems

Table 6.2 lists the input and output for sub-systems. The input of a sub-system contains the decision variables and relevant variables from other sub-system. The output of a sub-system contains its mass and energy balances. After the decision variables $\theta = [\theta_{MEA}, \theta_{PSA1}, \theta_{PSA2}, z_{FT}, \theta_{FT}, \theta_{MS}, Frac_{fuel_{ele-CCS}}, Frac_{steam_{ele-CCS}}]$ are given a set of values, the mass and energy balances of the whole CCU system can be assembled from the sub-systems.

Table 6.2. Inputs and outputs for sub-systems.

Sub-systems (i)	Input	Output
NGCC + MEA	θ_{MEA}	$F_{MEA,CO_2cap}, F_{MEA,CO_2e}, E_{NGCC-MEA}$
NGCC + PSA	$[\theta_{PSA1}, \theta_{PSA2}]$	$F_{PSA,CO_2cap}, F_{PSA,CO_2e}, E_{NGCC-PSA}$
Reforming + FT	$[\theta_{FT}, F_{MEA,CO_2cap}, F_{PSA,CO_2cap}, z_{FT}]$	$F_{FT,r}, F_{FT,p}, F_{FT,CO_2e}, U_{FT,u}$
Reforming+ MS	$[\theta_{MS}, F_{MEA,CO_2cap}, F_{PSA,CO_2cap}, 1 - z_{FT}]$	$F_{MS,r}, F_{MS,p}, F_{MS,CO_2e}, U_{MS,u}$

where,

θ_{MEA}: decision variables for MEA absorption process.

θ_{PSA1}: decision variables for 1st PSA process.

θ_{PSA2}: decision variables for 2nd PSA process.

θ_{FT}: decision variables for [reforming + FT].

θ_{MS}: decision variables for [reforming + MS] sub-system.

z_{FT}: distribution of captured CO_2 to [reforming + FT].

F_{MEA,CO_2cap}: mass flow of captured CO_2 by MEA absorption process, ton_{CO_2}/h.

F_{MEA,CO_2e}: mass flow of uncaptured CO_2 by MEA absorption process, ton_{CO_2}/h.

F_{PSA,CO_2cap}: mass flow of captured CO_2 by the two PSA in series, ton_{CO_2}/h.

F_{PSA,CO_2e}: mass flow of uncaptured CO_2 by the two PSA in series, ton_{CO_2}/h.

$E_{NGCC-MEA}$: net electricity output for [NGCC + MEA] sub-system, netMW.

$E_{NGCC-PSA}$: net electricity output for [NGCC + PSA] sub-system, netMW.

$F_{ii,r}$: mass flows for required raw materials, $ii = [FT, MS], r = [NG, H_2O]$, ton/h.

$F_{ii,p}$: mass flows for products, $ii = [FT, MS], p = [gasoline, diesel, MEOH]$, ton/h.

F_{ii,CO_2e}: mass flows for CO_2 emissions *via* the vent gas, $ii = [FT, MS]$, ton/h.

$U_{ii,u}$: utility consumption, $ii = [FT, MS], u = [fuel\ gas, steam, electricity, cooling]$, GJ/h.

6.3.2.2 Surrogates for sub-systems

6.3.2.2.1 [NGCC + MEA]

The [NGCC + MEA] sub-system aims to build the relationship between θ_{MEA} and $[F_{MEA,CO_2cap}, F_{MEA,CO_2e}, E_{NGCC-MEA}]$.

θ_{MEA} only contains one decision variable as the recovery rate of CO_2 in MEA process ($Re_{CO_2,MEA}$). When assigning a set of values to Re_{MEA}, the IECM software can yield the corresponding simulation outputs. With the inputs/outputs, I trained a linear regression model as follows:

$$[Re_{MEA}, \eta_{MEA}, H_2O_{netMWh}, MEA_{netMWh}] = surrogate_{NGCC-MEA}(\theta_{MEA}) \quad \text{Eq 6.1}$$

where, Re_{MEA}: recovery rate of CO_2 in MEA process.

η_{MEA}: net power out per NGCC power generation (partial loss in MEA), $netMW/MW$

With the default values for the parameters of NGCC, the IECM software can yield

GHG_{MWh}: emissions per unit of electricity generation, ton/MWh;

NG_{MWh}: NG consumption per unit of electricity generation, ton/MWh;

H_2O_{MWh}: water consumption per unit of electricity generation, ton/MWh.

After coupled with an MEA absorption process, a 500 MW NGCC power station generate net power ($E_{NGCC-MEA}$, netMW) and captured CO_2 (F_{MEA,CO_2cap}, ton_{CO_2}/h) as follows,

$$E_{NGCC-MEA} = \eta_{MEA} \cdot 500 \quad \text{Eq 6.2}$$

$$F_{MEA,CO_2cap} = 500 \cdot GHG_{MWh} \cdot Re_{MEA} \quad \text{Eq 6.3}$$

$$F_{MEA,CO_2e} = 500 \cdot GHG_{MWh} \cdot (1 - Re_{MEA}) \quad \text{Eq 6.4}$$

6.3.2.2.2 [NGCC + PSA]

The [NGCC + MEA] sub-system aims to build the relationship between $[\theta_{PSA1}, \theta_{PSA2}]$ and $[F_{PSA,CO_2cap}, F_{PSA,CO_2e}, E_{NGCC-PSA}]$.

Two surrogates are built for two PSA processes, respectively. The obtained CO_2 purity from 1st PSA is the inlet CO_2 concentration for the 2nd PSA. Their input variables are as follows,

$$\theta_{PSA1} = [P_{L1}, P_{I1}, v_{feed1}, t_{ads1}, t_{bd1}, t_{evac1}] \quad \text{Eq 6.5}$$

$$[Pu_{PSA1}, \theta_{PSA2}] = [Pu_{PSA1}, P_{L2}, P_{I2}, v_{feed2}, t_{ads2}, t_{bd2}, t_{evac2}] \quad \text{Eq 6.6}$$

Randomly-distributed values for input can be generated by employing LHS for the design space of input variables. Based on these input values, the rigorous PSA simulation on Dymola can yield outputs (purity, recovery and energy consumption). With the inputs/outputs, I trained two ANN models as follows,

$$[Pu_{PSA1}, Re_{PSA1}, Energy_{PSA1}] = Surrogate_{PSA1}(\theta_{PSA1}) \quad \text{Eq 6.7}$$

$$[Pu_{PSA2}, Re_{PSA2}, Energy_{PSA2}] = Surrogate_{PSA2}([Pu_{PSA1}, \theta_{PSA1}]) \quad \text{Eq 6.8}$$

where, Re_{PSA1}: recovery rate of CO_2 in 1st PSA process

Re_{PSA1}: recovery rate of CO_2 in 2nd PSA process

$\mathrm{Energy}_{\mathrm{PSA1}}$: energy consumption per CO_2 captured in 1st PSA, $\mathrm{MWh/ton}_{\mathrm{CO_2}}$

$\mathrm{Energy}_{\mathrm{PSA2}}$: energy consumption per CO_2 captured in 2nd PSA, $\mathrm{MWh/ton}_{\mathrm{CO_2}}$.

The overall performance for [NGCC + PSA] sub-system can be obtained in the following relationship:

$$\mathrm{Pu}_{\mathrm{PSA}} = \mathrm{Pu}_{\mathrm{PSA2}} \quad \text{Eq 6.9}$$

$$\mathrm{Re}_{\mathrm{PSA}} = \mathrm{Re}_{\mathrm{PSA1}} \cdot \mathrm{Re}_{\mathrm{PSA2}} \quad \text{Eq 6.10}$$

$$\eta_{\mathrm{PSA}} = 1 - \mathrm{GHG}_{\mathrm{MWh}} \cdot (\mathrm{Energy}_{\mathrm{PSA1}} \cdot \mathrm{Re}_{\mathrm{PSA1}} + \mathrm{Energy}_{\mathrm{PSA2}} \cdot \mathrm{Re}_{\mathrm{PSA}}) \quad \text{Eq 6.11}$$

where $\mathrm{Pu}_{\mathrm{PSA}}$: purity of CO_2 from the two PSA

$\mathrm{Re}_{\mathrm{PSA}}$: recovery rate of CO_2 from the two PSA

η_{PSA}: net power out per NGCC power generation (partial loss in PSA), $\mathrm{netMW/MW}$

After coupled with two PSA in series, a 500 MW NGCC power station generate net power ($\mathrm{E}_{\mathrm{NGCC-PSA}}$, netMW) and captured CO_2 ($\mathrm{F}_{\mathrm{PSA,CO_2cap}}$, $\mathrm{ton}_{\mathrm{CO_2}}/\mathrm{h}$) as follows,

$$\mathrm{E}_{\mathrm{NGCC-MEA}} = \eta_{\mathrm{PSA}} \cdot 500 \quad \text{Eq 6.12}$$

$$\mathrm{F}_{\mathrm{PSA,CO_2cap}} = 500 \cdot \mathrm{GHG}_{\mathrm{MWh}} \cdot \mathrm{Re}_{\mathrm{PSA}} \quad \text{Eq 6.13}$$

$$\mathrm{F}_{\mathrm{PSA,CO_2e}} = 500 \cdot \mathrm{GHG}_{\mathrm{MWh}} \cdot (1 - \mathrm{Re}_{\mathrm{PSA}}) \quad \text{Eq 6.14}$$

6.3.2.2.3 [Reforming + FT]

The [Reforming + FT] sub-system aims to build the relationship between $[\theta_{\mathrm{FT}}, \mathrm{F}_{\mathrm{MEA,CO_2cap}}, \mathrm{F}_{\mathrm{PSA,CO_2cap}}, \mathrm{z}_{\mathrm{FT}}]$ and $[\mathrm{F}_{\mathrm{FT,r}}, \mathrm{F}_{\mathrm{FT,p}}, \mathrm{F}_{\mathrm{FT,CO_2e}}, \mathrm{U}_{\mathrm{FT,u}}]$.

The amount of CO_2 flowing to [Reforming + FT] can be calculated as follows,

$$F_{FT,CO_2} = (F_{MEA,CO_2cap} + F_{PSA,CO_2cap}) \cdot z_{FT} \quad \text{Eq 6.15}$$

The input variables for [Reforming + FT] are as follows,

$$[F_{FT,CO_2}, \theta_{FT}] = [F_{FT,CO_2}, T_{FT}, P_{FT}, Tray_{ref1}, T_{ref1}, P_{ref1}, S_{purge}, Re_{FT}] \quad \text{Eq 6.16}$$

Based on randomly-sampled inputs, the rigorous simulation on Aspen Plus can yield outputs (mass flows and utilities). With the inputs/outputs, I trained an ANN model for [Reforming + FT] sub-system as follows,

$$[F_{FT,r}, F_{FT,p}, F_{FT,CO_{2e}}, U_{FT,u}] = Surrogate_{reforming+FT}([F_{FT,CO_2}, \theta_{FT}]) \quad \text{Eq 6.17}$$

6.3.2.2.4 [Reforming + MS]

The [Reforming + MS] sub-system aims to build the relationship between $[\theta_{MS}, F_{MEA,CO_2cap}, F_{PSA,CO_2cap}, 1 - z_{FT}]$ and $[F_{MS,r}, F_{MS,p}, F_{MS,CO_{2e}}, U_{MS,u}]$.

The amount of CO_2 flowing to [Reforming + MS] can be calculated as follows,

$$F_{MS,CO_2} = (F_{MEA,CO_2cap} + F_{PSA,CO_2cap}) \cdot (1 - z_{FT}) \quad \text{Eq 6.18}$$

The input variables for [Reforming + MS] are as follows,

$$[F_{MS,CO_2}, \theta_{MS}] = [F_{MS,CO_2}, T_{FT}, P_{FT}, Tray_{ref1}, T_{ref1}, P_{ref1}, S_{purge}, Re_{FT}] \quad \text{Eq 6.19}$$

Based on randomly-sampled inputs, the rigorous simulation on Aspen Plus can yield outputs (mass flows and utilities). With the inputs/outputs, I trained an ANN model for [Reforming + MS] sub-system as follows,

$$[F_{MS,r}, F_{MS,p}, F_{MS,CO_{2e}}, U_{MS,u}] = Surrogate_{reforming+MS}([F_{MS,CO_2}, \theta_{MS}]) \quad \text{Eq 6.20}$$

6.3.3 Level 3: surrogate-based optimization

In Level 3, surrogate-based optimization is performed, as illustrated in Figure 6.4. I deploy a simulation-based optimization approach, where simulation is executed within the optimizer. Level 1 and Level 2 offer process model inputs to one simulation platform, where decision variables and process uncertainties are used to run the overall flowsheet simulation. Subsequently, lifecycle GHG emission factors (Appx.B.1.6) and economic factors (Appx.B.2.5) are considered within the mass and energy balances calculated in the overall flowsheet simulation, thus resulting in the objective values. The optimizer varies the values of decision variables and improves the objectives iteratively. After the surrogate-based optimization is completed, I use the obtained values for the decision variables to perform rigorous simulations for individual sub-systems, as to validate the optimal solution.

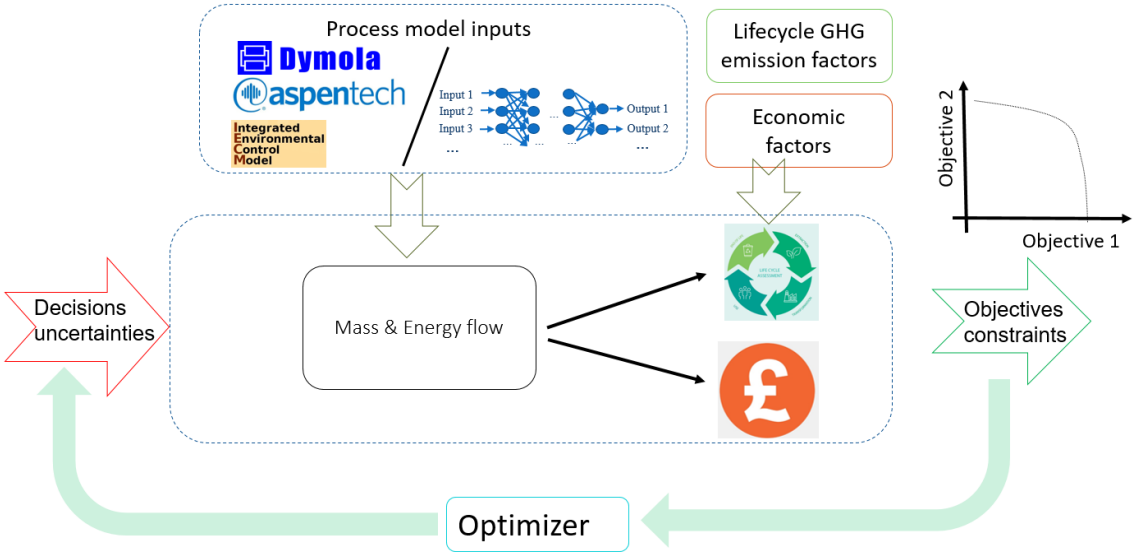


Figure 6.4. Detailed steps of the optimization deployed on Level 3: mass and energy balances, in conjunction with the input of environmental metrics (lifecycle GHG emissions) and economic factors are being used to evaluate the objectives and constraints. GA is the optimizer.

6.4 Single-objective optimization with respect to lifecycle GHG emissions reduction

The optimization framework described above was applied to assess the potential of CCU to solely reduce GHG emissions (*i.e.*, in the absence of renewable sources of energy). Here I only consider the GHG emission reduction as the objective of the optimization.

The GHG emissions are evaluated based on the life cycle assessment (LCA) with a cradle-to-gate boundary. I seek to compare the emissions from the reference process (described in Section 6.2: Problem statement) with emissions of the system with CCU. For a meaningful comparison (Figure 6.5), I evaluate multiple process configurations where both the reference process and the CCU system yield the same amount of electricity and fuels (defined as the system expansion strategy [76]). More detailed information for the system boundary and the system expansion strategy can be referred to Appx.B.1.1 - B.1.2.

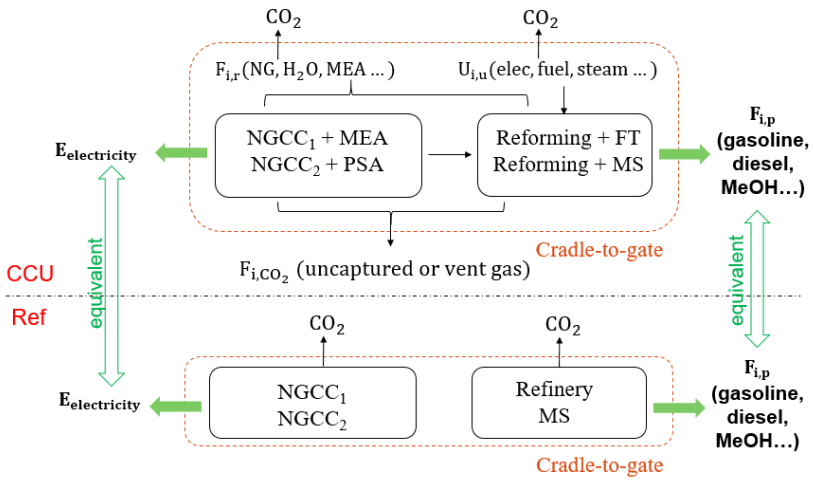


Figure 6.5. Comparison between CCU vs. Reference (Ref) system by the system expansion strategy: different process configurations considered within the optimization are designed to yield equivalent amounts of electricity and fuels for both CCU and Reference systems.

Based on the mass and energy flow from process models and lifecycle GHG emission factors (Appx.B.1.6), the GHG reduction is calculated in Eq 6.21 - Eq 6.23. More detailed information for the variables in equations can be found in Appx.B.1.

$$GHG_{CCU} = \sum_i \sum_r \alpha_r \cdot F_{i,r} + \sum_i \sum_u \alpha_u \cdot U_{i,u} + \sum_i F_{i,CO2} \quad \text{Eq 6.21}$$

$$GHG_{ref} = \alpha_{NGCC} \cdot E_{electricity} + \sum_i \sum_p \alpha_p \cdot F_{i,p} \quad \text{Eq 6.22}$$

$$GHG_{reduction} = 1 - \frac{GHG_{CCU}}{GHG_{ref}} \quad \text{Eq 6.23}$$

where

GHG_{CCU}	GHG emissions of the whole CCU system (the industrial park)
GHG_{ref}	GHG emissions of the reference system (no capture, refinery, MS)
F	Mass flow, ton/h
α_r	Lifecycle GHG emission factor per raw material r generation: ton_{CO_2eq}/ton_r
U	Consumption of utility, GJ/h
α_u	Lifecycle GHG emission factor per utility u generation: ton_{CO_2eq}/GJ
F_{i,CO_2}	Uncaptured CO_2 or CO_2 in the vent gas in sub-system i, ton_{CO_2eq}/h
α_{NGCC}	Lifecycle GHG emission factor per NGCC power generation: ton_{CO_2eq}/GJ
$E_{electricity}$	Net output of electricity from [NGCC + MEA/PSA], GJ/h
α_p	Lifecycle GHG emission factor per product p generation: ton_{CO_2eq}/ton_p
Subscript	
i	Notation for sub-systems
r	Notation for raw materials (natural gas, process water, MEA, etc.)
u	Notation for utilities (steam, fuel gas, electricity, cooling, etc.)
p	Notation for products (gasoline, diesel, methanol, etc.)

The optimization is formulated as follows,

$$\max_{\boldsymbol{\theta}}\left(1 - \frac{GHG_{CCU}}{GHG_{ref}}\right) \quad \text{Eq 6.24}$$

s.t. $LB \leq \theta \leq UB$ Eq 6.25

Genetic algorithm (GA) is used as the optimizer, and the optimization progress can be tracked with the generation (Figure 6.6). The mean objective value is the average objective value of populations at every iteration. In the initial generations, the mean objective value is negative, which indicates CCU can even cause more GHG emissions than the reference system. I terminate the optimizer after 50 iterations, where the mean objective value is closed to the best

objective. Here, I approximate the found values for decision variables as the optimal operating condition, as shown in Table 6.4. Under this condition, rigorous simulation is performed and yields a similar objective value as the simulation by surrogates.

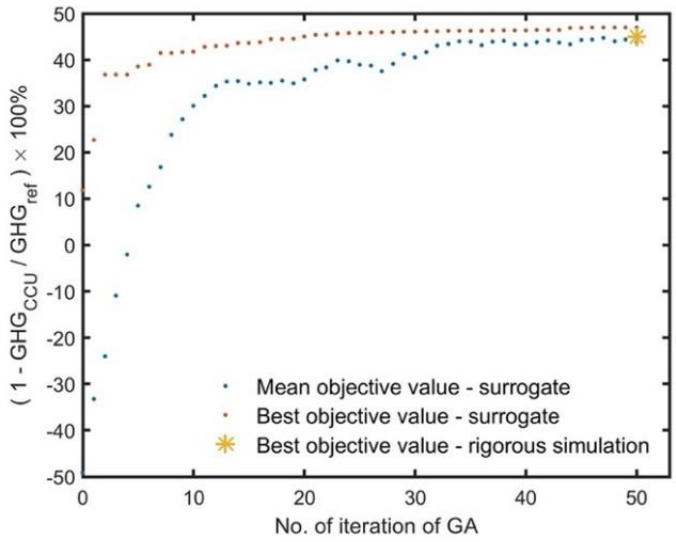


Figure 6.6. The optimization progress for GHG reduction in the industrial park.

Further, I find that surrogate simulation for GHG emissions of sub-systems is very close to rigorous simulation results under both initial (random guess) and optimal operating conditions (Figure S10). In fact, the surrogate is not necessary to be highly accurate. The crucial point is to find the improvement direction for decision variables, which can guide the improvement direction at a reduced computational cost during the optimization iterations.

The GHG emissions of sub-systems are presented in Figure 6.7. Under a random (initial) system configuration, CCU deployment results in more life cycle GHG emissions than the reference system, used to generate the same amount of electricity and products. This is because, within the initially evaluated process configuration, CO_2-based reforming requires extensive energy input, which can lead to more emissions if no proper operating conditions are set. For example, emissions from [Reforming + FT] are almost triple that from the refinery in the reference system (Figure 6.7.a). Under the optimal operating condition, the optimizer (by GA) recommends to produce methanol rather than gasoline (thus, emissions from [Reforming + FT] become negligible). This is probably because CO_2 cannot be converted in the FT path [108, 110], while CO_2 can be well utilized in MS [14, 175, 176].

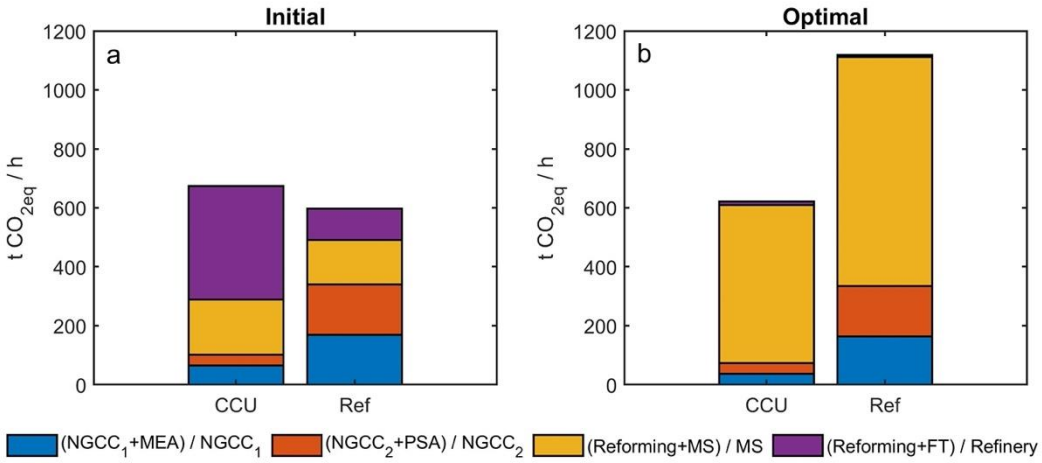


Figure 6.7. GHG emissions of sub-systems of the industrial park for the system with CCU deployment and the reference system (no CCU). Both systems are designed to deliver the equivalent output of products (electricity, fuels – during the optimization iterations, these values can change). a) Emissions for the initial configuration. b) Emissions for the configuration determined as optimal, where methanol production is favoured. Clarification for the legends: left of '/' for the CCU system, right of '/' for the reference system. The optimization is to maximize the difference of GHG emissions between reference system and CCU system, so the GHG emissions of reference system in the optimal condition are not necessarily smaller than that in the initial condition.

Furthermore, the optimization can distinguish between the choice of MEA from PSA unit operations, even though this is not evident from the system-level data. The use of carbon capture leads to two effects on the 500 MW NGCC plants: lowering the emissions but shrinking the net electricity output. As shown in Table 6.3, PSA has fewer emissions than MEA; 20% electricity loss is seen for the deployment of MEA, while 16% electricity loss for PSA. Hence, PSA has an advantage over MEA regarding GHG emissions reduction and energy saving. However, this advantage is negligible when referring to GHG emissions in the whole CCU system because more emissions are caused by the utilization paths than the capture paths (Figure 6.7b).

Table 6.3. Performance of carbon capture for 500 MW NGCC under the optimal operating condition.

		Emissions [ton CO_2/h]	Net electricity output [MW]
CCU	[$NGCC_1$ + MEA]	37.17	400
	[$NGCC_2$ + PSA]	36.66	418
Reference	[$NGCC_1$]	163.50	400
	[$NGCC_2$]	170.80	418

The optimal operating conditions are listed in Table 6.4. To maximize the GHG reduction, the requirements for sub-systems are as follows:

(1) MEA: high recovery rate is preferred.
(2) PSA: in 1st PSA, the P_{L1} should be low enough to enhance capture capacity, while this requirement is not strict for 2nd PSA. Long evacuation is preferred for two PSA columns, and thus sufficient time is allocated to recover the captured CO_2.
(3) MS is favoured over FT.
(4) Heating tends to be fully substituted by low-carbon electricity.
(5) In the reforming process, the ratio of NG/CO_2 is suggested to approach the upper bound, meaning sufficient NG is required to substantially convert CO_2 to CO in the reforming section.

Table 6.4. Initial guess and optimal values (by GA) for decision variables.

	Decision variables θ	Unit	Initial (base case)	Optimal	Decision Index
MEA	r_{CO_2}	-	0.775	0.933	(1)
1st PSA	P_{L1}	bar	0.0075	0.007	(2)
	P_{I1}	bar	0.285	0.406	(3)
	v_{feed1}	m s^{-1}	1.05	0.614	(4)
	t_{ads1}	s	60	68.789	(5)
	t_{bd1}	s	115	32.515	(6)
	t_{evac1}	s	115	183.637	(7)
2nd PSA	P_{L2}	bar	0.0275	0.014	(8)
	P_{I2}	bar	0.285	0.170	(9)
	v_{feed2}	m s^{-1}	1.05	0.534	(10)
	t_{ads2}	s	60	59.411	(11)
	t_{bd2}	s	115	44.544	(12)
	t_{evac2}	s	115	178.820	(13)
CO_2 to FT	z_{FT}	-	0.75	0.027	(14)
FT	T_{FT}	℃	240	247.886	(15)
	P_{FT}	bar	32.5	25.904	(16)
	$tray_{FT}$	-	55	62	(17)
	T_{ref1}	℃	875	876.081	(18)
	P_{ref1}	bar	5	5.073	(19)
	S_{purge}	-	0.1005	0.045	(20)
	Re_{FT}	-	0.5	0.573	(21)
MS	F_{NG}/F_{CO_2}	-	2.85	3.498	(22)
	T_{MS}	℃	200	204.318	(23)
	P_{MS}	bar	65	69.542	(24)
	$Tray_{MS}$	-	55	46	(25)
	T_{ref2}	℃	900	933.319	(26)
	P_{ref2}	bar	5	6.224	(27)
Heating utility	$Frac_{fuel_{ele-CCS}}$	-	0.2	0.997	(28)
	$Frac_{steam_{ele-CCS}}$	-	0.2	0.956	(29)

While determining the optimal conditions, GA tends to replace fossil fuel-based heating with low-carbon electricity generated from sources deploying carbon capture and storage (CCS). However, I anticipate that there might exist several techno-economic limitations towards a complete substitution of heating by decarbonized electricity sources. Hence, I performed a set of scenario analyses for the heating substitution regarding the upper bound for substituting heating utility is set as 0, 25%, 50%, 100% (optimization progresses can be referred to Figure 6.8 and optimal operating condition in Table S17).

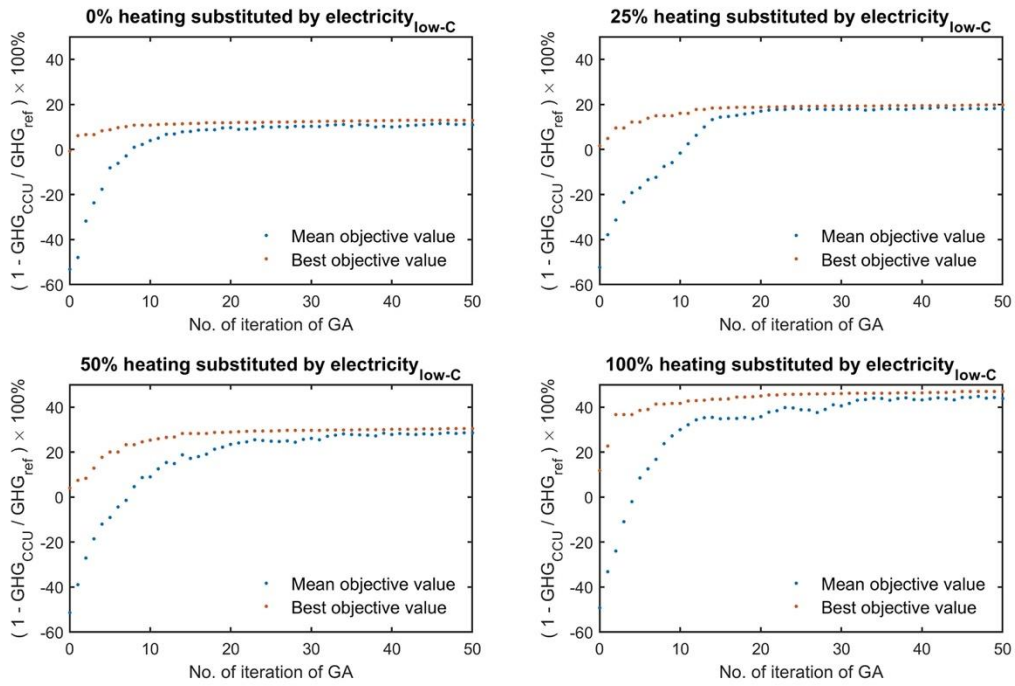


Figure 6.8. Scenario analysis for the optimization progress, regarding 10%, 25%, 50% and 100% heating utility is substituted by electricity.

After optimization, the GHG emissions can be reduced, ranging from 13 to 47%, while all the substitution percentages to low-carbon electricity tend to approach the upper bounds, as shown in Table 6.5.

Table 6.5. Scenario analysis for the optimization result of the industrial park, regarding 0-100% heating utility is substituted by low-carbon electricity (CCS-electricity).

Max substitution [%]	0	25	50	100
GHG reduction [%]	13.0	19.8	30.5	47.0
Fuel sub [%]	0	24.9	49.8	99.7
Steam sub [%]	0	22.3	49.6	95.6

Figure 6.9 shows the breakdowns of sources for GHG emissions in the industrial park. The largest source is heating, followed by NG, CO_2 emissions *via* vent gas and electricity, *etc*. When increasing the heating substitution from 0 to 100%, the GHG emissions can be reduced by 40%. By contrast, GHG emissions are negligible for the cooling, process water and MEA. Yet, even for 100% heating substitution by CCS-electricity, I can spot that heating still holds the most considerable contribution to GHG emissions.

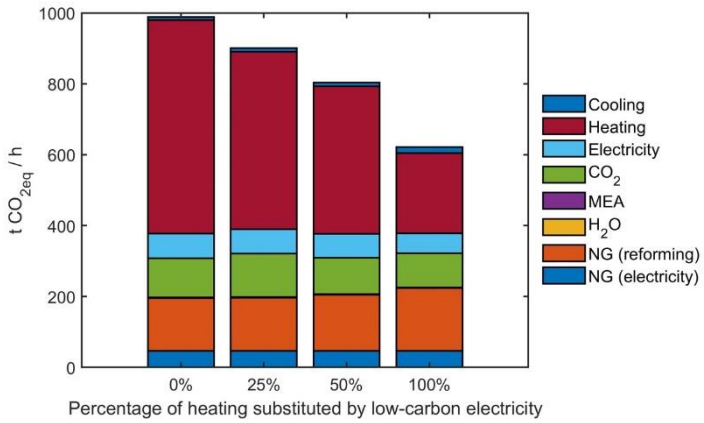


Figure 6.9. Sources of GHG emissions in the industrial park. Results correspond to the optimization result of the industrial park, regarding 0%, 25%, 50% and 100% heating utility are substituted by low-carbon electricity (CCS-electricity).

6.5 Multi-objective optimization regarding emissions reduction and economic gain

Upon exploring the capability of CCU to decarbonize the NG-based power plants and fuels production, I sought to include the economic aspect into the optimization framework. The economic evaluation is under the following assumptions:

(1) The cost calculation considers the operational cost only, since the technology readiness level of CCU is relatively low and its capital cost cannot be quantified accurately [32].
(2) This industrial park is operated in the EU. Economic assessment is based on the prices data for materials/utilities in the first half of 2021. No carbon tax is assumed at this stage of analysis. Since the gas crisis from the second half of 2021 in the EU [180], the prices of petrol products have surged in different levels in the past year, so the latest prices data may cause discrepancies.

Based on the mass and energy balances and economic factors (Appx.B.2.5), the profit of the CCU system is calculated as follows,

$$\text{Profit} = -\sum_{i}\sum_{r}\beta_{r}\cdot F_{i,r} - \sum_{i}\sum_{u}\beta_{u}\cdot U_{i,u} - \sum_{i}F_{i,CO_2}\cdot\gamma_{CO_2} + \beta_{CCS}\cdot E_{electricity} + \sum_{i}\sum_{p}\beta_{p}\cdot F_{i,p} \quad \text{Eq 6.26}$$

where

$F_{i,r}$	Mass flow of raw material r in sub-system i, ton/h
β_r	Cost of raw material r, $\$/\text{ton}_r$
$U_{i,u}$	Consumption of utility u in sub-system i, GJ/h
β_u	Cost of utility u, $\$/\text{GJ}$
F_{i,CO_2}	CO_2 emissions in the vent gas in sub-system i, $\text{ton}_{CO_2\text{eq}}/\text{h}$
β_{CCS}	Price of CCS electricity, $\$/\text{GJ}$
$E_{electricity}$	Net output of electricity from [NGCC + MEA/PSA], GJ/h

β_p	Price of product p, $\$/\mathrm{ton}_p$
γ_{CO_2}	Carbon price ('0' in this section), $\$/\mathrm{ton}_{CO_2}$
Subscript	
i	Notation for sub-systems
r	Notation for raw materials (natural gas, process water, MEA, etc.)
u	Notation for utilities (steam, fuel gas, electricity, cooling, etc.)
p	Notation for products (gasoline, diesel, methanol, etc.)

The formulation of relevant equations and economic data can be referred to Appx.B.2. The optimization is formulated as follows:

$$\max_{\boldsymbol{\theta}}\left[\left(1-\frac{\mathrm{GHG}_{\mathrm{CCU}}}{\mathrm{GHG}_{\mathrm{ref}}}\right),\ \mathrm{profit}\right] \qquad \text{Eq 6.27}$$

$$\text{s.t.} \quad \mathrm{LB} \leq \theta \leq \mathrm{UB} \qquad \text{Eq 6.28}$$

To solve it, I use the non-dominated sorting genetic algorithm-II (NSGA-II), a stochastic optimization algorithm that approximates the Pareto front. Pareto front offers a set of trade-off solutions, where one objective cannot be improved without worsening the other one.

6.5.1 Pareto front

Surrogate-based optimization yields the optimal values for decision variables (Figure S11 - Figure S12). Based on these optimal decisions, rigorous simulations are performed to calculate the two objectives. As shown in Figure 6.10, deviation exists between the surrogate and rigorous simulations. Such deviation is difficult to avoid, because deviation can even occur in a small system, *e.g.*, DCHM process (see Figure 4.7). The overall deviation for the large CCU system results from the sum of sub-systems' deviations.

The trade-off effect can be reflected in the Pareto front. When I set the GHG emissions reduction objective to a high value at 42%, the profit is even negative; yet pursuing a high profit (>3.8e5 $/h) can make the CCU system release even more emissions than conventional processes.

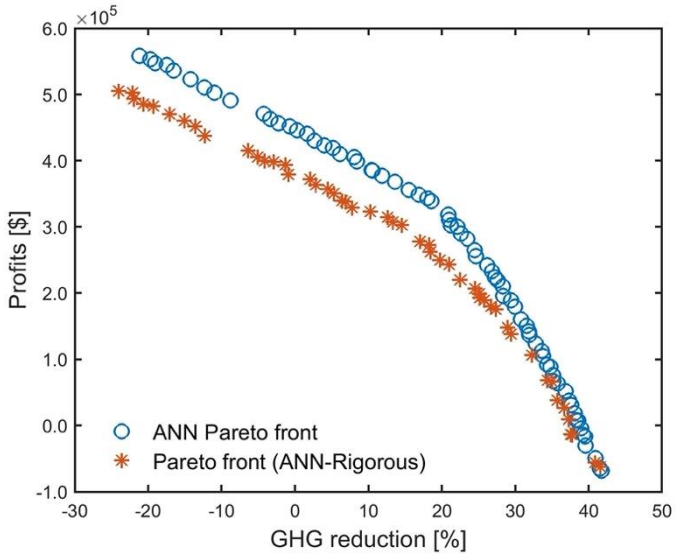


Figure 6.10. Multi-objective optimization of the CCU system: Pareto front between profit and GHG emissions reduction [Note: 'ANN-Rigorous' means that ANN-based optimization yields the optimal values for decision variables, and then rigorous simulations are performed to calculate the two objectives.].

To better understand the trade-off between the two objectives, I refer to the economic breakdowns of several Pareto points, which correspond to GHG emissions reduction values (Figure 6.10) at -24%, 0%, 15%, 30% and 42%, respectively. As shown in Figure 6.11, improving GHG reduction leads to a gradual growth of utility costs and dropping revenue. Table 6.6 indicates that the increasing utility cost is caused by the rising percentage of heating electrification, because the energy price of low-carbon heating can be over four times that of fuel gas or steam (Figure S13). Meanwhile, the shift from FT to MS can further promote the GHG reduction but sacrifice the economic revenue, because the market price of methanol is much lower than FT fuels – gasoline/diesel (Table S16).

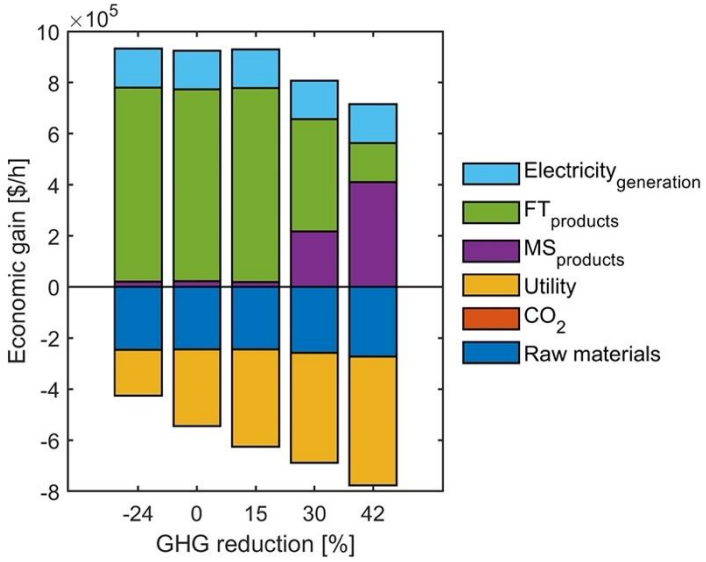


Figure 6.11. The breakdowns of economic gain in several Pareto front points (selected based on GHG emissions reduction at -24%, 0%, 15%, 30% and 42%).

Table 6.6. The trend of selected Pareto front points.

GHG reduction	-24%	0	15%	30%	42%
Profit [$/h]	5.06e5	3.80e5	3.03e5	1.19e5	-6.20e4
z_{FT}	0.963	0.958	0.963	0.565	0.195
$Frac_{fuel_{ele-CCS}}$	0.047	0.579	0.933	0.965	0.984
$Frac_{steam_{ele-CCS}}$	0.631	0.754	0.843	0.940	0.997

Clarification: z_{FT}: split of CO_2 to FT. $1 - z_{FT}$: split of CO_2 to MS. $Frac_{fuel_{ele-CCS}}$: fraction of fuel heating substituted by CCS heating. $Frac_{steam_{ele-CCS}}$: fraction of steam heating substituted by CCS.

6.5.2 Optimal values for decision variables

When referring to the optimal values for the decision variables, multi-objective optimization can recommend the operating conditions for individual processes. As shown in (Figure S11 - Figure S12), each subplot refers to one decision variable, while each circle in a subplot

corresponds to one solution found by NSGA-II (corresponding to a point in Pareto front in Figure 6.10).

Table 6.7 compares the suggested operating conditions by single-objective optimization and multi-objective optimization. On the one hand, both suggest some similar operating conditions. For example, MEA is recommended to approach the upper bound in both cases. On the other hand, two types of optimization differ on some operating conditions: single-objective optimization suggests some extreme conditions (approach either lower or upper bound of decision variables). In contrast, multi-objective optimization offers more moderate operating conditions. For example, single-objective optimization selects the lowest P_L (corresponding to the best recovery for CO_2 and the highest energy consumption [22]) for PSA; MS is also chosen as the main CO_2 utilization pathway; FT is almost neglected in CCU; heating is fully supplied by the low-carbon electricity. By contrast, multi-objective optimization determines a relatively low value for P_L and mixes FT with MS in the utilization pathways. This is because the multi-objective optimization delivers more practical solutions, where GHG reduction is balanced with the economic gain. This balance can be reflected in the utilization pathway selection and heating supply. Specifically, utilization to gasoline/diesel (FT path) can bring in more economic benefits, while utilization to methanol (MS path) and electrifying heating is more environmental-friendly.

In brief, multi-objective optimization can recommend moderate operating conditions for the industrial park. Relating the Pareto front to decision variables can offer an insight into how environmental and economic aspects are affected by operating conditions. By contrast, the extreme operating conditions tend to significantly sacrifice either economic or environmental aspects.

Table 6.7. Best operating conditions (decision variables) found by single-objective *vs.* multi-objective optimization.

	Suggested operating conditions (θ) by		Decision Index
	Single-objective optimization GA (Table 6.4)	Multi-objective optimization NSGA-II (Figure S11Figure S12)	
MEA	High recovery rate	High recovery rate	(1)
1st PSA	P_L approaches the lowest Long adsorption Short desorption for N_2 Long desorption for CO_2	P_L is relatively low Long adsorption Long desorption for N_2 Long desorption for CO_2	(1-7)
2nd PSA	P_L approaches the lowest Long desorption for CO_2	P_L is relatively low Long desorption for CO_2	(8-13)
Utilization pathways	MS is favoured over FT	FT is favoured over MS sometimes FT and MS co-exist sometimes	(14)
FT	(not important, because FT is not selected) FT 248 ℃, 26 bar Distillation 62 trays Reformer 876 ℃, 5.0 bar Purge % at 4.5% More recycle to FT section	(very important, because FT is selected as a key utilization path) FT 244-246 ℃, 28 bar Distillation 55-57 trays Reformer 947 - 950 ℃, 4.2-4.5 bar Purge % at 4.4 – 7.2% More (>80%) recycle to reforming	(15-21)
MS	NG/CO2 = 3.5 MS reactor inlet 204 ℃ MS reactor inlet 70 bar Distillation 46 trays Reformer 933 ℃, 6.2 bar	NG/CO_2 = 3.6-3.7 MS reactor inlet 196-198 ℃ MS reactor inlet 66 bar Distillation 55 trays Reformer 863-881 ℃, 5.6-5.7 bar	(22-27)
Heating	Fuel-gas heating is fully substituted by low-carbon elec. Steam-based heating is fully substituted by low-carbon elec.	Fuel-gas heating is partially substituted by low-carbon elec. Steam-based heating is over 60% substituted by low-carbon elec.	(28-29)

6.5.3 Robustness check for the optimal solutions

NSGA-II is a stochastic optimization technique, so the found solution theoretically cannot guarantee the optimality unless infinite iterations are performed. To check whether the best

solutions found in our case are robust or not, I evaluate two extreme scenarios regarding the selection of utilization pathways – fully employing either FT or MS. As shown in Figure 6.12, either way does not deliver better solutions than the found solution found by NSGA-II. On the one hand, the CO_2 utilization *via* entirely FT tends to bring in a higher profit, but the potential for GHG reduction is limited to 20%. On the other hand, fully MS can enhance GHG reduction to 46% but dramatically lose the economic advantage compared to the original solution found by NSGA-II.

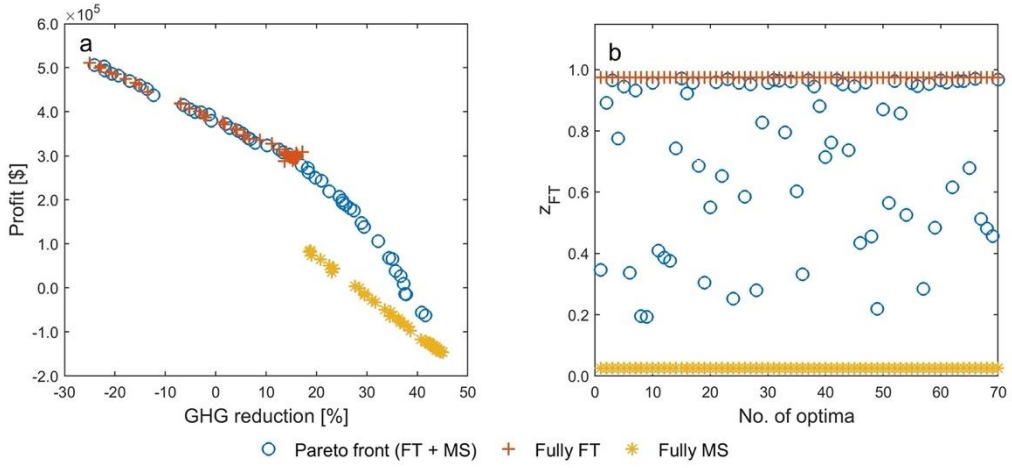


Figure 6.12. Robustness check for multi-objective optimization result – comparing with the scenarios where CO_2 is fully utilized *via* FT or MS: (a) trade-off between profit and GHG reduction for the industrial park; (b) the fraction of CO_2 utilization *via* FT (values of other operating conditions keep the same).

Similarly, I evaluate another extreme scenario, where the heating is fully substituted by low-carbon electricity. As shown in Figure 6.13, such a complete substitution brings in minor improvement on GHG reduction but significantly sacrifices the economic gain.

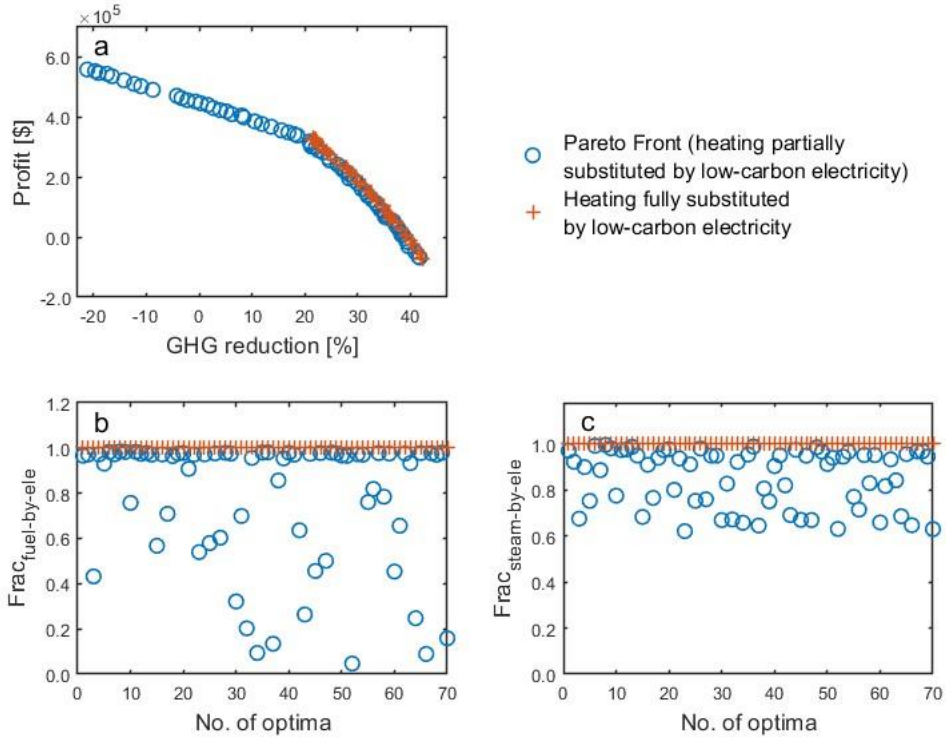


Figure 6.13. Robustness check for multi-objective optimization result – comparing with the scenario where the heating is fully substituted by low-carbon electricity: (a) trade-off between profit and GHG reduction for the industrial park; (b) the fraction of fuel-gas heating substituted by low-carbon electricity; (c) the fraction of steam heating substituted by low-carbon electricity (values of other operating conditions keep the same).

6.5.4 Discussion

The initial focus of this work is to scrutinize the potential for carbon reduction by the simultaneous optimization of the entire industrial park, and thus I focus on a scenario where there exists a high local demand for the CCU products, therefore there are no market-related constraints on how much CCU products can be generated. Nevertheless, the developed methodology does allow to consider market capacity as an optimization constraint. In the future, I will consider the connection to the supply chain for products and the local demand for CCU products, then the actual flowrate of CCU products, transportation / distribution and the size of the industrial park will be taken into account.

Further, there are many uncertainties involved in this proposed industrial park. In the initial conceptual process design, our current work does not consider size / dimension of plants, which allow for more design flexibility for the future and must be carefully evaluated in the next stage – a more robust process design. I assume that reforming can achieve the equilibrium at the different high temperatures, but the conversion efficiencies may not be ideal in practice, especially when coke formation and catalyst deactivation occur. The location of hypothetical industrial park is also essential, because the location choice can affect (1) the compositions of natural gas, which then influence the overall mass balance; (2) prices of raw materials, utilities and products. Additionally, the economic evaluation is subject to external factors, *e.g.*, market dynamics. All these factors can contribute to the deviation of LCA-Economic trade-off curve. A more robust method can be optimization under uncertainty, where the uncertainties are incorporated into the objective function.

6.6 Influence of carbon pricing

Lastly, I sought to examine the influence of carbon pricing on the CCU system. IEA reports that carbon price will significantly increase up to 250 \$/ton-CO_2 by 2050 for advanced economies [1]. As predicted by Nicholson *et al.*, the rising carbon prices can raise the energy cost [172], as a result of an extra financial constraint for the utility emissions, which can be roughly assessed by multiplying the emission factors by the carbon price (Eq 6.29). I embedded different strategies for carbon tax deployment and assumed that carbon pricing is imposed both on emissions resulting from both utility usage and on the life-cycle emissions from the carbon-based raw materials and products.

Now, the economic factors contain two parts: original prices and carbon tax as follows,

$$\beta_u = \beta_{u,0} + \alpha_u \cdot \gamma_{CO_2} \quad \text{Eq 6.29}$$

$$\beta_r = \beta_{r,0} + \alpha_r \cdot \gamma_{CO_2} \quad \text{Eq 6.30}$$

$$\beta_p = \beta_{p,0} + \alpha_p \cdot \gamma_{CO_2} \quad \text{Eq 6.31}$$

where

β Economic factors, $\$/\text{ton}$

α	Lifecycle GHG emission factors, $\mathrm{ton}_{\mathrm{CO}_2}/\mathrm{ton}$
γ_{CO_2}	Carbon price, $\$/\mathrm{ton}_{\mathrm{CO}_2}$
Subscript	
i	Notation for sub-systems
r	Notation for raw materials (natural gas, process water, MEA, etc.)
u	Notation for utilities (steam, fuel gas, electricity, cooling, etc.)
p	Notation for products (gasoline, diesel, methanol, etc.)
0	Notation for original price (no carbon tax applies)

Based on the optimization results for decision variables at no carbon price, the profits are re-calculated under other carbon prices. Figure 6.14 presents the change of the trade-off curves after involving the carbon pricing from 0 to 250 \$/ton-CO_2. With the increase of carbon price, the profit shifts to different directions depending on the GHG reduction. At a low GHG reduction, the profit drops with the carbon tax increase; at a high GHG reduction, the trend is reversed.

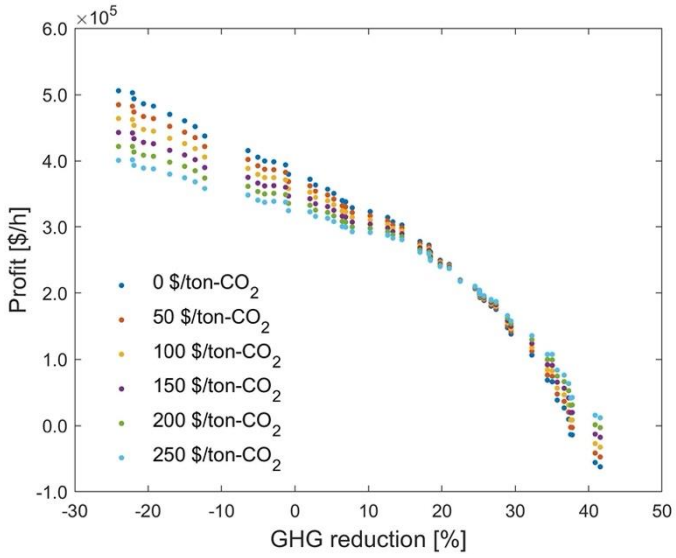


Figure 6.14. Influence of carbon price on the trade-off curve between profit and GHG emissions reduction.

To investigate why the trade-off curves shift to different directions, I pick the points at -24%, 23% and 42% of GHG reduction, under which I check their economic breakdowns. As shown in Figure 6.15, I can find carbon tax has dual effects on this CCU system. On the one hand, the process cost increases with the growth of carbon tax. This is because carbon tax adds an extra burden to any carbon-related materials and energy, including raw materials, utilities and unreacted CO_2 emissions. On the other hand, the revenue from fuel products rises with the growth of carbon tax. This is because the carbon tax is pre-assumed to increase the price of fuel products, which brings in extra credits to the CCU system. MS can reduce more GHG emissions, then the credit for methanol is larger than FT products. Hence, the revenue increase in methanol is much more significant than that in FT products, which reflects that raising carbon tax brings in more revenue at 42% GHG reduction than that at -24% GHG reduction.

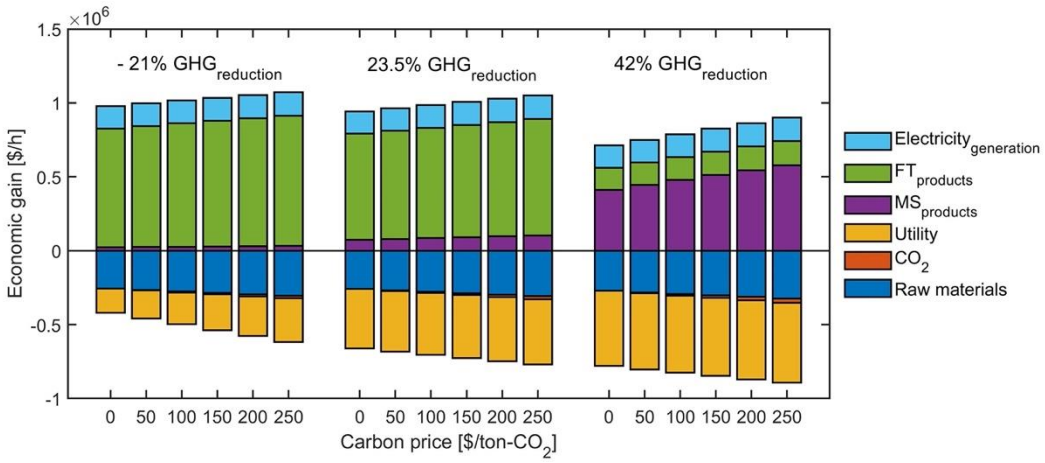


Figure 6.15. Influence of carbon price on the breakdowns of economic gain.

Additionally, carbon price sets a higher penalty for utilities with higher emissions. As shown in Figure 6.16, the cost of utilities with direct emissions surges faster than the low-carbon utilities. This explains why the utility cost at -24% GHG reduction, when the percentage of heating electrification is very low (Table 6.6), grows significantly with the increase of carbon tax (Figure 6.15). By contrast, heating is almost fully substituted by low-carbon electricity at 42% GHG reduction, so the growing carbon tax does not notably change the utility cost.

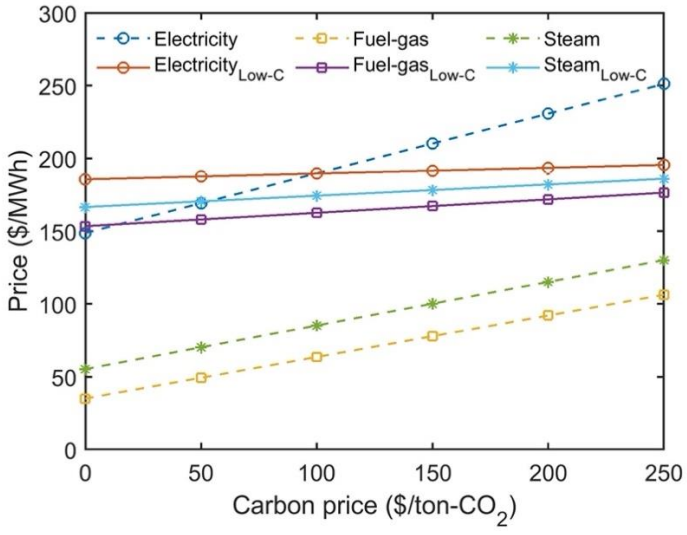


Figure 6.16. Influence of carbon price on the costs of utilities.

Overall, at a higher GHG emissions reduction, the carbon tax promotes a higher growth rate for credit gain and a lower growth rate for the penalty. By contrast, a lower GHG emissions

reduction has an inverse trend. As such, increasing carbon tax brings the trade-off curves in an intersection at 23% GHG reduction, where the growth rate of cost is equivalent to that of product revenue. Notably, I analyzed here only the net profit from the CCU system, without considering how its economic performance would compare to a direct-emission system, which will become significantly less economic under the increasing carbon pricing scenario.

Further, optimization can be done under different carbon tax values. When referring to the values of decision variables (Figure S14 - Figure S15), I do not find a specific trend with the increase of carbon tax (before optimization, I expected that a higher carbon tax would push the optimizer to give a higher percentage for MS path or heating electrification). This may result from the fact that the whole system is nonlinear and complex, and too many parameters (*e.g.*, economic factors) can interact with each other. As a result, the system is not sensitive to the carbon price. Another explanation is that carbon pricing cannot promote low-carbon technologies, as reported in the literature [169, 170]. However, introducing carbon pricing to the CCU system is more or less predicting the future energy cost, which is inevitably subject to a high degree of uncertainty. Uncertainty can also exist in determining a proper carbon pricing scheme, such as allowing extra credit for carbon capture or putting a penalty on producing petrol fuels. To give a more convincing conclusion, more scenarios for different carbon pricing schemes should be analyzed and tested in other regions (*e.g.,* America or Asia has a quite different set of economic factors from the EU, so the influence of the same policy scheme may be quite different).

6.7 Conclusions

In this chapter, I proposed a decarbonization strategy that does not rely on the deployment of renewable energy sources for a hypothesized industrial park – a large CCU system. The advantage of ‘no renewables’ delivers a more robust evaluation on CCU. This CCU system is fully digitalized by ANN-based surrogates and evaluated in a cost-efficient manner. Surrogates can retain the nonlinearity of sub-systems and offer the detailed quantitative understanding of driving factors in GHG emissions sources and economic aspects for CCU. The result shows that CCU can in principle be worse for GHG emissions than the conventional (unabated gas)

process, if one is not careful on operating conditions. Such a ‘negative result’ can help raise the awareness and more discussions in the CCUS field.

After optimization, CCU can cut GHG emissions by 13% compared with the conventional process. This optimization framework avoids sub-optimal solutions by simultaneously optimizing the whole CCU system, and I found that the GHG emissions in utilization dominate the whole CCU system, so optimizing the utilization path can be more rewarding than the capture path. The GHG emissions breakdowns indicate that heating is the most significant contributor to GHG emissions of the whole system. Electrifying heating fully by CCS electricity and fully producing methanol in the utilization pathways can reduce GHG emissions by 47% compared to the conventional process. Still, such extreme conditions will significantly sacrifice the economic benefit. By contrast, multi-objective optimization suggests the production of mixed methanol/gasoline/diesel and partial heating electrification, which can achieve a better trade-off between GHG emissions reduction and economic gain.

This work also discusses the dual effect of the carbon price on this CCU system. On the one hand, carbon pricing puts an extra cost on the raw materials and utilities. On the other hand, the carbon price can also bring in a ‘credit’ effect when reducing GHG emissions in production. The impact of carbon price on the techno-economic performance of CCU is therefore complex to predict. To make it clearly, more scenarios for different carbon pricing schemes should be analyzed.

Since CCU is compatible with existing energy, chemicals and mobility sectors, CCU can be an excellent stepping-stone to renewables.

Chapter 7 Optimization given the interactions of sub-systems

This chapter aims to examine whether surrogate-based sub-systems can capture their interactions. The feasibility will be checked by a case study on a reactor-separator-recycle system. Following this, I will go back to a complex CCU system to check whether the interaction of sub-systems will influence process design and synthesis.

For a large flowsheet, the interaction of sub-systems can exist in two types:

(1) Like circular economy, the waste of one sub-system can be the raw materials of another sub-system [181].
(2) Competition exists between sub-systems. It can be either due to the supply limitation of mutual raw materials or due to the response to the market demand.

The first type is not within the scope of this thesis, and I only present a simple case to demonstrate the capacity of surrogates to capture the interactions between sub-systems. I will focus on the second type of problem – whether the limitation of NG (raw materials) may influence the operation of a CCU industrial park. Financial Times report that the EU experienced the NG crisis in 2021 [182]. The high demand for NG pushed its price to soar all the way from ~ 10 euro/MWh in 2020 to ~ 70 euro/MWh in Sep 2021 [182]. Such gas crisis unavoidably leads to the scenario of limited NG supply. The insufficient supply of raw materials may limit the production of certain products. Similarly, the mismatch of supply/demand can cause price fluctuation, which promotes the production of particular products during a certain period.

7.1 Interactions captured by surrogates

To demonstrate the interactions in process systems, I build a simple reactor-separator-recycle system (Figure 7.1). On the one hand, the reactor outflow has a direct influence on the separator. On the other hand, the separator can affect the reactor inflow *via* the recycle stream. Hence, the interaction exists between the reactor and separator.

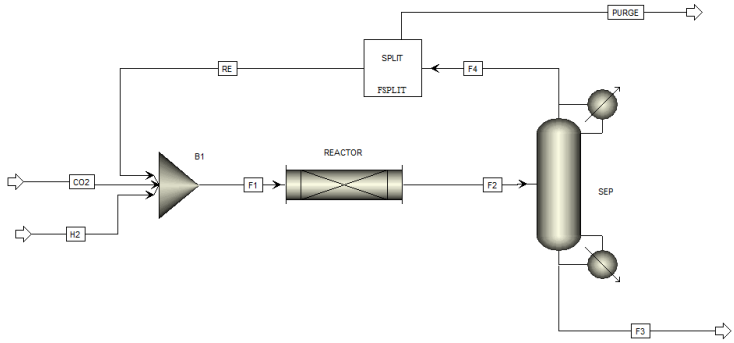


Figure 7.1. Interactions in a reactor-separator-recycle system.

I apply two ANN-based surrogates to represent the reactor and separators (Figure 7.2). Here, a sufficient number of simulations are performed to generate input/output dataset, followed by the surrogate building for the reactor and separators, respectively.

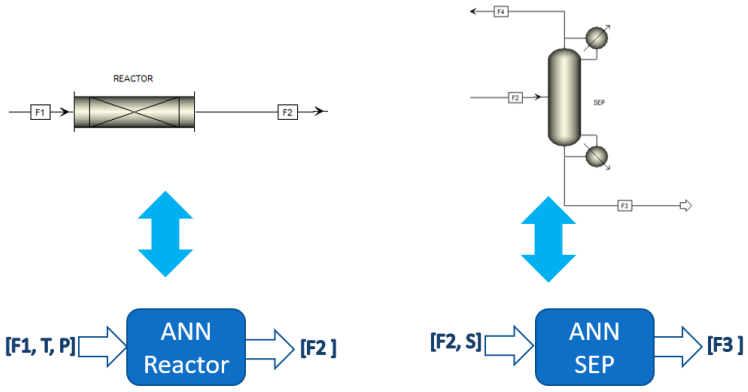


Figure 7.2. ANN-based surrogates for reactor and separator, respectively.

With the obtained surrogates, I establish a surrogate-based flowsheet as shown in Figure 7.3. This is implemented in MATLAB. The convergence of the recycle stream is achieved by the

fsolve function, where the left side of tear stream is required to equal to the right side. (PS. *fsolve* is a default function in MATLAB to solve the system of nonlinear equations.)

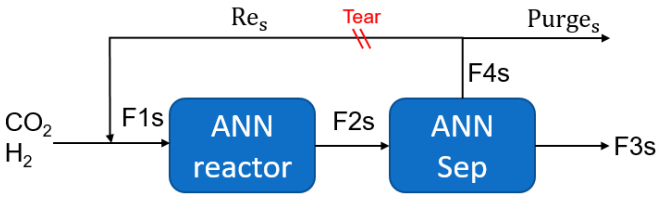


Figure 7.3. ANN-based surrogates for the reactor-separator-recycle system.

To evaluate whether the interactions are captured or not, I examine the relative errors between surrogates and rigorous simulations for the components in several streams. Boxplots are employed to show the distribution of relative errors between surrogate prediction and test data. As shown in Figure 7.4, all components in the different streams can well predicted, with very small relative errors less than 0.3%. Therefore, ANN-based surrogates can capture the interactions between sub-systems.

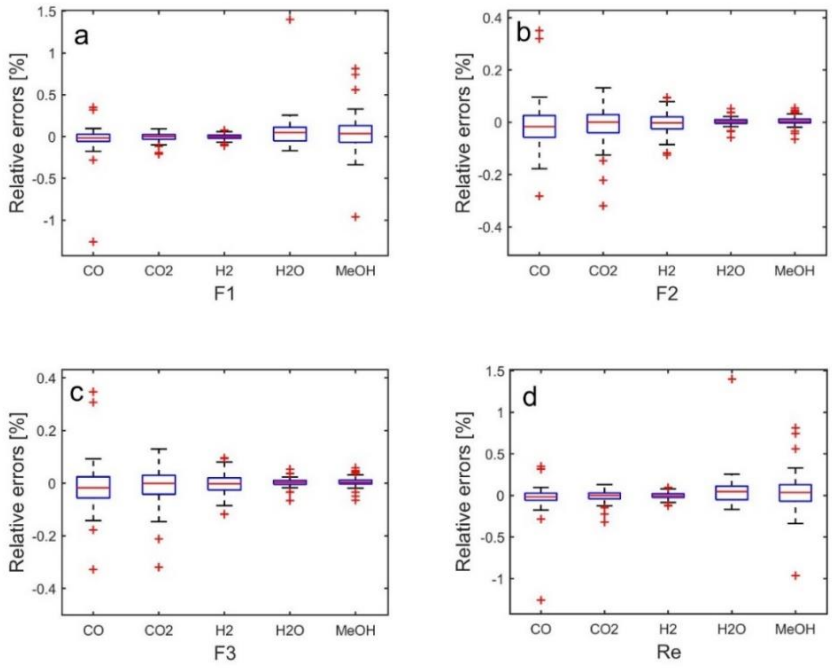


Figure 7.4. Relative errors of stream components by surrogate models for the reactor-separator-recycle system.

7.2 Influence of interactions on CCU

This section will evaluate an interaction effect by competition. In the proposed industrial park, NG is required by both utilization pathways – combined reforming for FT and MS. In other words, the two pathways may compete due to the supply of NG, as shown in Figure 7.5. Based on the analysis in Chapter 6, PSA performs slightly better than MEA, but the difference can be negligible compared to the utilization section. For a fair comparison, two 500 MW power plants (NGCC) employ MEA as the carbon capture technology, forming two [NGCC+MEA] sub-systems. Two utilization pathways exist, and each [NGCC+MEA] can select either FT or MS for the fuel production (x_i=0 for FT; x_i=1 for MS).

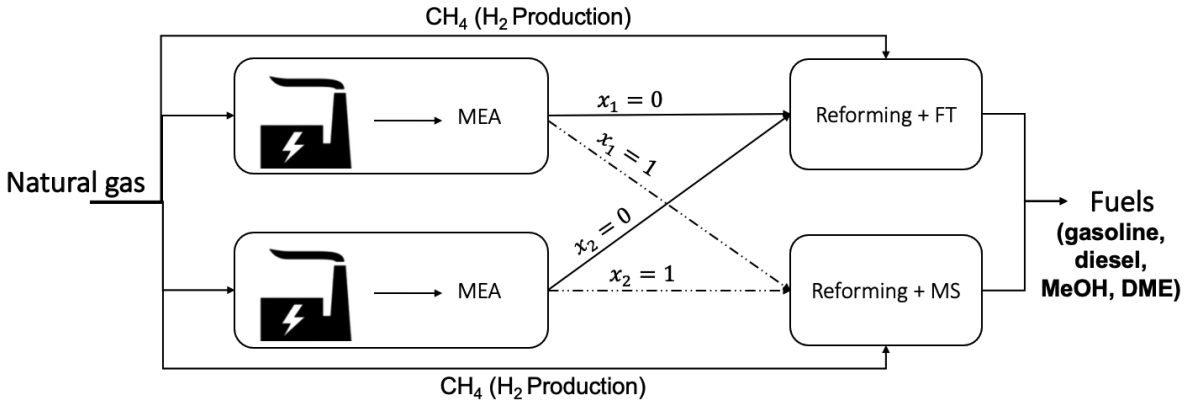


Figure 7.5. NG is required in the various sections in the industrial park of CCU system.

Herein, I set up three scenarios, which depend on the amount of the NG supply. To develop an overall low-carbon system, the carbon efficiency is desired to be maximized under the constraint of NG supply. To solve it, an optimization is formulated as follows:

$$\max_{x_i, FT_i, MS_i} C\% \qquad \text{Eq 7.1}$$

s.t. $NG_{demand} \leq NG_{supply}$ Eq 7.2

$x_i \in [0,1], i = 1,2$ Eq 7.3

$FT_{lb} \leq FT_i \leq FT_{ub}; MS_{lb} \leq MS_i \leq MS_{ub}$ Eq 7.4

Distinct sequence $x_1 \leq x_2;\ FT_1(1) \leq FT_2(1); MS_1(1) \leq MS_2(1)$ Eq 7.5

$$C\% = \frac{\sum_{i=1}^{2}[x_i C_{P_{FT}i} + (1-x_i)C_{P_{MS}i}] \times 100\%}{2C_{NG_{power}} + \sum_{i=1}^{2}[x_i C_{NG_{FT}i} + (1-x_i)C_{NG_{MS}i}]} \quad \text{Eq 7.6}$$

$$NG_{demand} = 2NG_{power} + \sum_{i=1}^{2}[x_i NG_{FT_i} + (1-x_i)NG_{MS_i}] \quad \text{Eq 7.7}$$

In the optimization formulation, the carbon efficiency is maximized (Eq 7.1). The first constraint requires that the consumption of NG (demand) must be smaller than the supply (Eq 7.2). x_i is a binary variable, referring that a utilization option selects either FT or MS (x_i=0 for FT; x_i=1 for MS). Within the individual process, the operating conditions must be constrained within the reasonable range (Eq 7.4). A distinct sequence is guaranteed by the ranking of some decision variables (Eq 7.5). For instance, when one FT (x_i=0) and one MS (x_i=1) are selected, the optimizer will automatically deliver a distinct solution as the first utilization pathway for FT and the second one for MS (because $x_1 \leq x_2$). The formulation of carbon efficiency is calculated by the ratio between carbon mass in the final fuels over that in NG shown in (Eq 7.6). The NG demand is formulated in Eq 7.7.

The optimization framework is the same as Section 6.4. The optimization progress is monitored as shown in Figure 7.6. For an individual subplot, I can find that the average value of the objective gradually approaches the best-found value of objective. This means that GA can iteratively identify better values for decision variables to improve carbon efficiency, under the supply constraint of NG. When comparing among the subplots, I can see that the supply of NG can dramatically affect carbon efficiency. With more supply of NG, the carbon efficiency can be significantly improved.

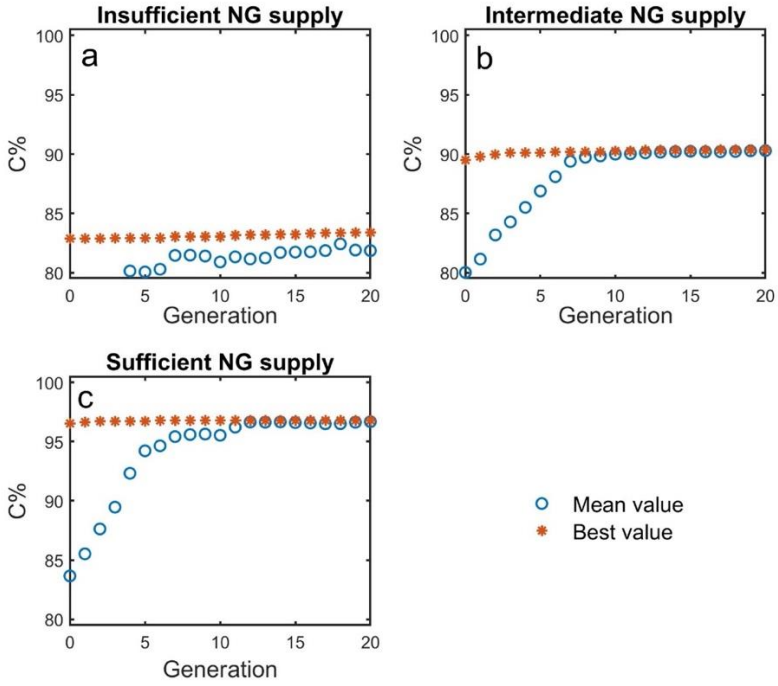


Figure 7.6. Optimization progress regarding three scenarios: (a) insufficient supply of NG - 27,000 kmol/h; (b) intermediate supply of NG - 30,000 kmol/h; (c) sufficient supply of NG (no constraint is set for the optimization algorithm).

I present the best-found value for the objective and decision variables, as shown in Table 7.1. For the objective, the surrogate value is very close to that of the rigorous simulation. Again, this can validate the feasibility of surrogate-based optimization for large systems. For the decision variables, I find that the supply of NG can determine the choice of sub-systems: (1) the limited supply of NG favours two MS; (2) when increasing the supply of NG, the optimization leads to one MS plus one FT; (3) when unlimited NG can be accessed, two FT will be selected by the optimizer to boost the carbon efficiency over 96%.

Table 7.1. Objective and decisions under the three scenarios of NG supply constraint.

Constraint scenarios		Insufficient	Medium	Sufficient
NG supply		[27,000 kmol/h]	[30,000 kmol/h]	[no constraint]
Objective C%	ANN optimization	83.36	90.38	96.79
	Aspen validation	83.28	90.36	96.81
Decisions	x_1	0	0	1
	x_2	0	1	1
	$FT_1(1)$	-	-	215
	$FT_1(2)$	-	-	0.001
	$MS_1(1)$	180	182	-
	$MS_1(2)$	2.4	2.4	-
	$FT_2(1)$	-	247	215
	$FT_2(2)$	-	0.001	0.001
	$MS_2(1)$	181	-	-
	$MS_2(2)$	2.3	-	-
	Working mode	**2 different MS**	**1 MS + 1 FT**	**2 same FT**

Additionally, the NG supply affects the operating conditions: the unlimited NG delivers two FT processes with exactly the same operating conditions, while the limited NG supply delivers two MS processes with different operating conditions. This is because the unlimited NG means no competition exists between two utilization pathways; thus two FT processes can be driven to their individual best performance. By contrast, insufficient NG leads to the competition between two MS processes regarding the NG supply. Consequently, the individual optimal solution cannot be achieved due to the interaction, and one sub-system may give away a little advantage to the other one, resulting in two different operating conditions.

These three scenarios reflect the advantages of surrogate-based optimization for large system over the optimization with rigorous process simulations embedded. If rigorous simulations are used within the optimization iterations, the function evaluations are too expensive and non-convergence issues can also break down the optimization iterations. In addition, if sub-systems may be represented by the simulation result of individual optimal solution, the operating conditions of sub-systems are pre-determined before the overall optimization of whole system. Such a method may solve the problem as Scenario 3 but can never work out Scenario 1 or 2. Scenario 3 is an ideal situation, when the optima of individual sub-systems can be achieved. However, the real-world is not ideal and always comes with constraints (*e.g.*, Scenario 1 or 2). Inspiringly, surrogate-based sub-systems can solve the above-mentioned issues. On the one hand, surrogates are reduced-order models, thus saving excessive computational costs in function evaluations compared to the rigorous process models. On the other hand, surrogate-based optimization can vary critical characteristics of sub-systems during the optimization iterations (maintain a certain nonlinearity), thus adjusting the operating conditions adapting to the real-world interactions.

The developed surrogate-based optimization framework can be employed to adjust the CCU system, according to the availability of energy sources. This is because the developed optimization framework can also be generalized to involve other energy sources, e.g., coal and renewables. In the short term, coal still plays an essential role in the energy security especially under the pressure of gas crisis, and coal can be an energy input for the industrial park. By this flexible methodology, an additional model / surrogate can be added to describe flue gas pre-treatment unit operations for the removal of NO_x, SO_x gases and flying ashes, as typically required at coal-fired plants; more attentions are required to consider the influence of the gas impurities (*e.g.*, sulphur impurities can deactivate the FT catalysts [183]) on the utilization pathways, leading to a different catalyst kinetics and process models / surrogates. In the long term, the supply of renewable energy is expected to increase and the industrial park should gradually introduce more renewable energy inputs, novel unit operations and increasing electrification, in order to enhance the decarbonization capacity in the supply, process and demand aspects.

7.3 Conclusions

This chapter validates the feasibility of using surrogate-based sub-systems to capture the interactions in two case studies: a reactor-separation-recycle system and a CCU system. Regarding the CCU system, our method can recommend optimal CCU configurations together with the operating conditions. When the gas crisis occurs (the NG supply is insufficient), the sub-systems can compete for the conflicting interest, while our method optimizes the sub-systems simultaneously and support decision-making for the whole CCU system.

To generalize the method, the interaction between sub-systems can be formulated as constraints in an optimization problem. Such a framework can be extended to optimizing an even larger system (*e.g.*, a supply chain system) with more process options, and our framework can optimize the process options with the operating conditions simultaneously.

Chapter 8 Conclusions and Outlook

This thesis is divided into two parts: **Part I** is focused on the methodology development: surrogate-based optimization; with developed methodology, **Part II** solves the problem for the optimization of large CCU systems. Conclusions and outlook are summarized from the two parts. Further, I reflect on how the established optimization framework can be extended to other low-carbon process systems.

8.1 Methodology development: surrogate-based optimization

Machine learning-based surrogates belong to black-box models, thus making the gradient information challenging to extract. Without the gradient information, surrogate-based optimization cannot guarantee optimality. Chapter 3 uses a hybrid method to optimize PSA, proving that surrogate-based optimization can closely approach optimality.

After assessing a set of surrogate types, I identify that ANN shows a slightly better regression capacity and more straightforward implementation for multiple outputs of process systems. Data generation for process systems is computationally expensive, so Chapter 5 employs a dynamic sampling method to reduce the data quantity and a classifier to improve the data quality.

Some future works are recommended to improve the accuracy of surrogates. The exploitation-based sampling methods can be introduced to integrate with the sampling workflow in Chapter 5, as to properly increase sampling probability in the nonlinear/complex design space. In this way, the data quality can be further improved. Another solution is to apply feature selection techniques [151, 152] (*i.e.*, automatically adjust input variables) to improve regression performance.

Notably, an improved model accuracy is beneficial to predict the system performance, but it is not necessary to be highly accurate. As shown in Figure 8.1, I make an analogy to computer-aided experiments (*e.g.*, closed-loop optimization [157, 184]): computer simulations are less accurate than experiments, while these simulations can still guide the experiments. The crucial point is to find the improvement direction for decision variables. Similarly, for a large process system, the accuracy inevitably decreases from industrial data to physical model-based simulations, to surrogates. Meanwhile, surrogates accelerate the evaluation on the system. When the surrogates can provide the right improvement direction, the final result can still improve. Further, I reflect that pursuing optimality may not be necessary for conceptual process design. The knowledge of the system is limited, and thus even the physical model has significant uncertainty. Consequently, the optimizer's curse probably occurs because the model error can be larger than the closeness to optimality. Under such a situation, an optimal solution might not be achievable in reality. Hence, exerting extensive effort to pursue optimality might not be wise in the initial process design.

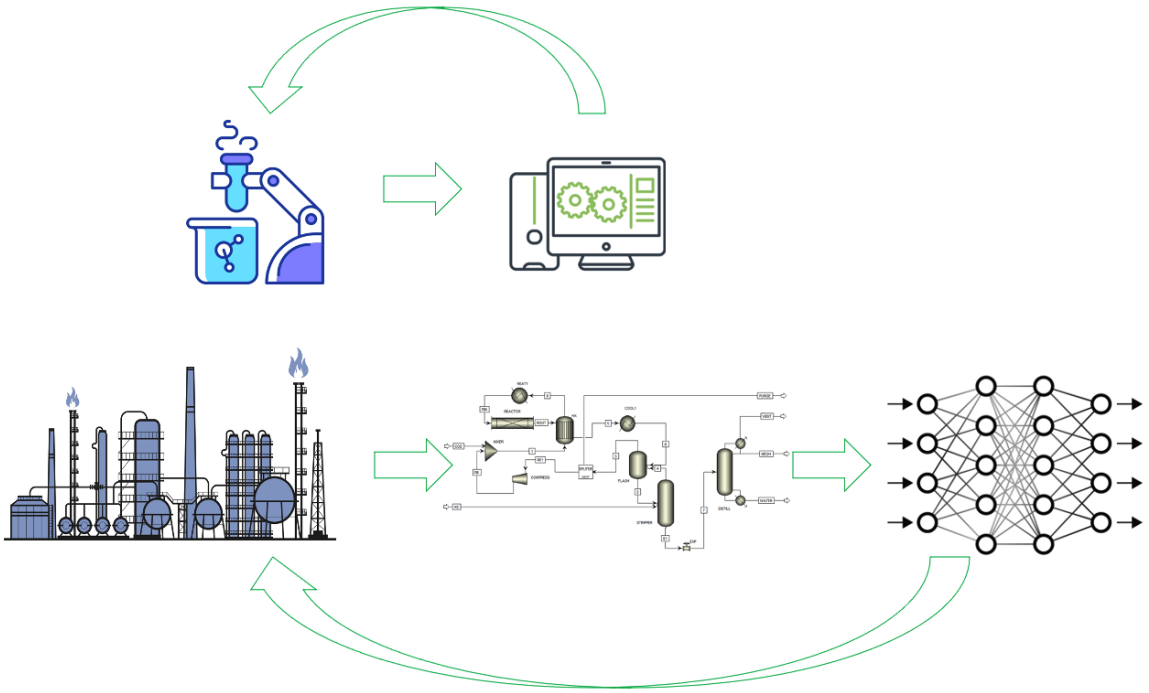


Figure 8.1. Analogy between computer-aided experiments and surrogate-based optimization. From left to right, accuracy decays while efficiency improves.

(Part of image elements in Figure 8.1 come from VectorStock with the standard license EULA https://www.vectorstock.com/faq/member/standard-license)

8.2 Problem solving: optimization of an industrial park

The three-level optimization framework contains rigorous-to-surrogate digitalization and allows for optimization of complex problems with conflicting objectives, as illustrated with the case of CCU system (the proposed industrial park). The initial design of this industrial park does not rely on the deployment of renewable energy sources; hence the proposed decarbonization strategy offers a solution, which is not dependent on the growth of renewables sectors. The sub-systems of the industrial park are fully digitalized by ANN-based surrogates and then simultaneously optimized in a cost-efficient manner. The benefits of surrogates are as follows:

(1) surrogates maintain a certainty complexity/nonlinearity for individual sub-systems;
(2) surrogates keep the interactions between sub-systems;
(3) surrogates represent the sub-systems, and then assemble in a high-level platform to evaluate the whole CCU system;
(4) surrogates accelerate the evaluation for the whole CCU system.

By scrutinizing the interactions between different unit operations proposed for carbon capture and utilization sections, optimization enables to determine a process configuration allowing for substantial reduction of CO_2 emissions. Through comparing the emissions from sub-systems under the optimal solution, I found that the GHG emissions in utilization dominate the whole CCU system, so optimizing the utilization path can be more rewarding than the capture path. This finding benefits from optimizing the sub-systems simultaneously.

Multi-objective optimization can deliver well-rounded solutions than single-objective optimization. Single-objective optimization tends to recommend extreme conditions, *e.g.*, fully electrifying heating by CCS electricity and fully producing methanol in the utilization pathways. Under such conditions, GHG emissions can be reduced by 47% compared to the conventional process, but the economic benefit is dramatically sacrificed. By contrast, multi-objective optimization suggests the production of mixed methanol/gasoline/diesel and partially

heating electrification, which can achieve a better trade-off between GHG reduction and economic profit.

Overall, CCU can be a stepping-stone to renewables. On the one hand, CCU is compatible and capable of decarbonizing the existing energy sector and offering fuels to the mobility sector. On the other hand, since the CCU is mainly designed to decarbonize non-sustainable energy sources (fossil fuels), CCU will not be the ultimate solution.

This part of work inspires several research directions as follows,

(1) The breakdowns analysis indicates that heating plays a dominating role in GHG emissions from CCU systems. A future research direction can be the heat integration for the overall system and the heat supply using low-carbon utility, *e.g.*, solar reforming [101, 185, 186]. Developing low-cost heating is also recommended to enhance economic viability.
(2) CCU is a regional problem, and its economic analysis is subject to the choice of locations. EU is selected in this work. A comprehensive conclusion can be obtained by changing the locations to Asia and America, since the emissions reduction is a global issue.
(3) This work briefly discusses the dual effects of the carbon price on this CCU system: raising the cost of raw materials and utilities and gaining credits when emissions are reduced in producing valued products. The current work is based on the system boundary of ‘cradle-to-gate’. In the future research, the system boundary can be expanded to ‘cradle-to-grave’, where the fuels consumption is probably be influenced by carbon price.
(4) Another research direction is how to balance Economic-LCA criteria in a proper way. LCA is closely related to the technical aspect, so the assessment result will not change remarkably for the future trend. By contrast, the economic aspect relies on market dynamics - the supply/demand relationship and its evaluation is involved with remarkable uncertainty for the future. Therefore, the next stage of research work may allow a higher weight in LCA than the economic aspect.
(5) A flexible system structure is recommended and leaves a broader decision space. This allows the possibility of renewable inputs in the future. For example, electrolysis can

offer an extra feedstock of H_2 for CO_2 utilization. Gradually raising the proportion of renewables can further enhance the capacity of emissions reduction.

8.3 Optimization and other low-carbon process systems

This thesis shows that how digitalization and optimization can be powerful tools to explore the potential of CCU. This can be extended to other low-carbon process systems. Low-carbon process systems have two aspects: low-carbon technology and its maximum potential in carbon reduction. In a certain sense, the low-carbon technology type has determined the maximum potential, which can be achieved by optimization. Figure 8.2 lists emission factors of several power generation technologies [187], and the type of technology has determined the upper and lower bound of emissions. Specifically, the scope of this work is [gas + CCUS], which integrates the gas-fired power plants with CCU, as well as the heating utility partially substituted by CCS electricity. When the fuels are converted to the equivalent energy, Eq 8.1 can deliver the range of emission factor per energy generation of [Gas + CCUS], see the red column in Figure 8.2. Single-objective optimization can reach the lower bound for the emission factor. In contrast, multi-objective optimization offers moderate solutions with slightly more emissions but dramatically improves other objectives, such as the economic aspect.

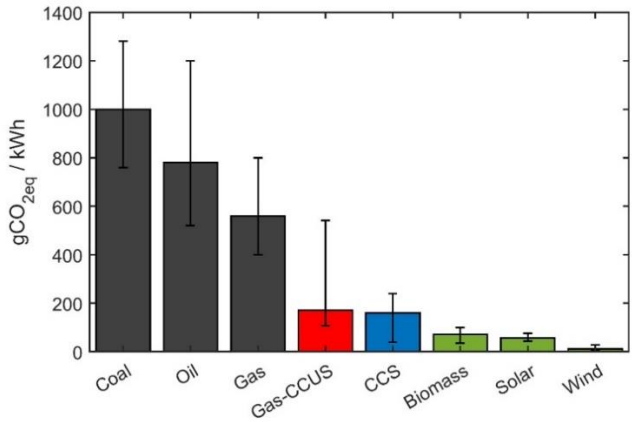


Figure 8.2. Lifecycle GHG emission factors of various power generation technologies. The scope of this thesis is [gas-CCUS], and its emission factor is calculated as Eq 8.1. Other values can be found in Weisser [187].

$$\alpha_{CCU} = \frac{GHG_{CCU}}{E_{electricity} + \sum_i \sum_p e_p \cdot F_{i,p}} \quad \text{Eq 8.1}$$

where

α_{CCU}	Lifecycle GHG emission factor of CCU: ton_{CO_2eq}/GJ
GHG_{CCU}	GHG emissions of the whole CCU system, ton_{CO_2eq}/h
F	Mass flow, ton/h
e	Energy density: GJ/ton
$E_{electricity}$	Net output of electricity from [NGCC + MEA/PSA], GJ/h
Subscript	
i	Notation for sub-systems
p	Notation for products (gasoline, diesel, methanol, *etc.*)

Further, this thesis is not against 'renewable power-to-X' systems. Introducing the renewables to CCU can enhance the potential of carbon reduction (*i.e.*, the lower bound of [gas-CCUS] can be even lower in Figure 8.2). However, this requires a much higher renewable installed capacity, and then there exists more surplus electricity to be converted to chemicals for the energy storage purpose. Another requirement is to cope with the intermittency of renewable energy sources by the feasibility of dynamic operation for the subsequent processes, which also relies on (dynamic) optimization.

Low-carbon process systems will be consistent with the circular economy, regarding material recycling/reuse and energy integration within a large system. As such, interactions between sub-systems are necessary to be captured. The digitalization tool developed in this thesis lays a solid foundation in the interaction capture.

In the long term, net zero needs various low-carbon pathways. While their decarbonization performances are enhanced by optimization, the overall progress of net zero will be accelerated.

Appendix A Process description

A.1 Pressure swing adsorption

A.1.1 A simulation result of PSA

The full model of PSA can be the sum of four stages multiplied by corresponding binary variables (Eq. 3.19). Through varying binary variables $[Y_1, Y_2, Y_3, Y_4]$, the four stages of PSA model can be simulated continuously. The simulation of PSA requires the numerical integration of a series of initial value problems (IVP). After completing four stages, re-initialize the cycle time (t_{cycle}) as 0 and then start the simulation of another one cycle of PSA. The PSA cycle is simulated iteratively until a cyclic steady state (CSS) is reached. Theoretically, when a CSS is reached, the column profile is expected to be the same between the same step in two subsequent cycles. In the mathematical language, when $\left|\mathrm{x(t)} - \mathrm{x(t + t_{cycle})}\right| < \delta$, PSA is deemed to be under CSS. The dynamic simulation of PSA can be found in Figure S1.

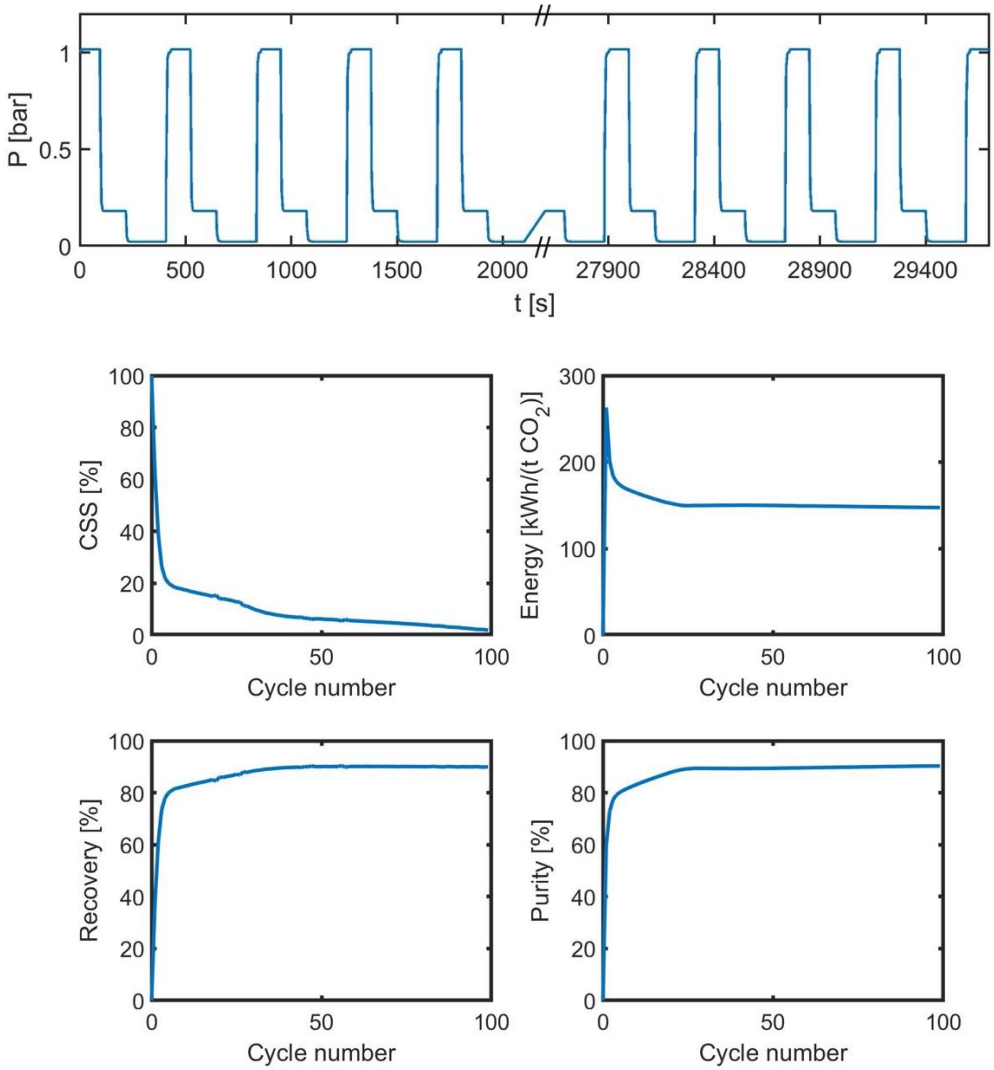


Figure S1. The dynamic behaviour of PSA. The simulation is based on the operating condition $[t_{ads}, t_{bd}, t_{evac}, P_I, P_L, v_{feed}, y_{CO_2}]$ = [94.89 s, 122.84 s, 189.46 s, 0.18 bar, 0.02 bar, 0.58 m/s, 0.15].

A full dynamic simulation can contain much more results than shown here, but those results might not be relevant in the process analysis and optimization. This is a motivation to develop a reduced-order surrogate.

A.1.2 An example of one input-output data point for PSA surrogate

According to the simulation result as shown in Figure S1, I can collect one input-output data point in Table S1. Obviously, this data point has much less information than a dynamic simulation result in Dymola, but this data point has contained the essential features for the PSA system regarding the operating condition and the corresponding process output. A sufficient number of such effective data points (by varying the values of input variables and simulating the values for output variables) can be used to train a surrogate for PSA process.

Table S1. An example of one input-output data point for PSA surrogate.

Input variables	Value	Unit	Notes
t_{ads}	94.89	[s]	Duration of adsorption stage
t_{bd}	122.84	[s]	Duration of blowdown stage
t_{evac}	189.46	[s]	Duration of evacuation stage
P_I	0.18	[bar]	Setpoint of intermediate pressure
P_L	0.02	[bar]	Setpoint of low pressure
v_{feed}	0.58	[m/s]	Inlet flowrate
y_{CO_2}	0.15	[-]	Inlet molar fraction of CO_2
Output variables			
Recovery	0.8967	[-]	Recovery rate of CO_2
Purity	0.8934	[-]	Purity of CO_2 in the product flow
Energy	150.22	[kWh/ton-CO_2]	Energy usage per ton CO_2 captured

A.1.3 Validation of simulation results

As shown in Table S3, this model is validated by comparison to the energy consumption in the literature [23]. Given the same operating condition, the simulation result can be well-reproduced.

Table S2. Validation of simulation results of energy consumption by comparison with the literature values reported by Haghpanah *et.al.*[23]

Energy [kWh/(t CO_2)]	Literature	simulation reproduce 1	simulation reproduce 2	simulation reproduce 3
Operating condition I	213.22	213.01	213.01	213.01
Operating condition II	176.14	177.96	177.96	177.96
Operating condition III	148.96	150.22	150.22	150.22

Operating condition I: $[t_{ads}, t_{bd}, t_{evac}, P_I, P_L, v_{feed}, y_{CO_2}]$ = [89.55, 34.54, 120.70, 0.26, 0.005, 0.91, 0.15]; operating condition II: $[t_{ads}, t_{bd}, t_{evac}, P_I, P_L, v_{feed}, y_{CO_2}]$ = [80.24, 54.05, 101.83, 0.22, 0.01, 0.86, 0.15]; operating condition III: $[t_{ads}, t_{bd}, t_{evac}, P_I, P_L, v_{feed}, y_{CO_2}]$ = [94.89, 122.84, 189.46, 0.18, 0.02, 0.58, 0.15]. The units are as follows: , $t_{ads}\ [s], t_{bd}[s], t_{evac}[s]$, $P_I\ [bar], P_L\ [bar], v_{feed}\ [m/s], y_{CO_2}\ [-]$.

A.1.4 Optimization set-up of PSA on DyOS

A set of values (In Chapter 3, TSEMO offers these values) for decision variables are required to initialize DyOS. One cycle of full-order PSA model is programmed in the Modelica language, which is then compiled to Functional Mockup Unit (FMU) as a model input to DyOS. DyOS calls the FMU repeatedly until CSS, the objective (recovery) and the constraint (purity) are evaluated in the last cycle. Full state mapping links the state variables between two subsequent cycles, which can overcome the discrete/continuous issue of PSA. Through the integrator and NLP solver within DyOS, the optimal values for decision variables can be determined.

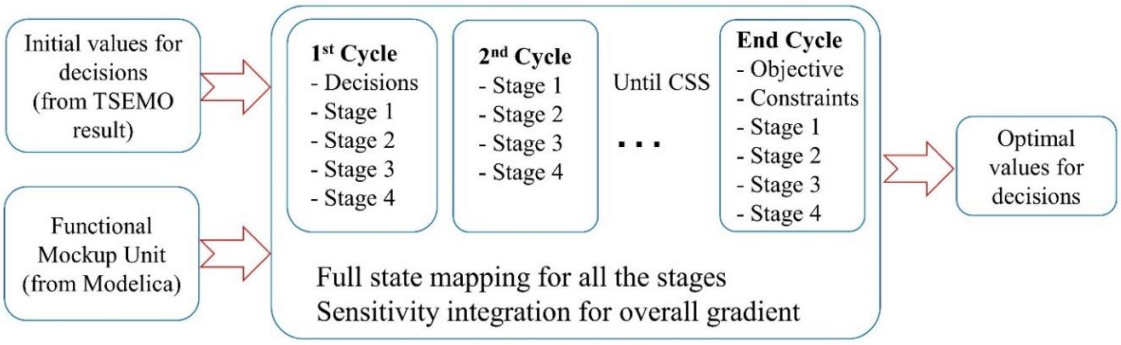


Figure S2. Optimization set-up of PSA on DyOS.

A.1.5 Values of decision variables by the hybrid approach

Table S3 shows the values of the decision variables corresponding to the Pareto front shown in Figure 3.5. The evacuation pressure (P_L) and blowdown pressure (P_I) are driven to the lower bound following the gradient in DyOS, while the inlet flowrate changes little. Haghpanah *et al.* reported that a lower evacuation pressure (P_L) can remove side-products and improve CO_2 recovery [23], which is consistent with the result in this thesis.

Table S3. hybrid approach for the multi-objective optimization of PSA: the corresponding decision variables.

TSEMO decisions – 600 simulations						DyOS decisions		
$100P_L$	$10P_I$	v_{feed}	$tads/100$	$tbd/100$	$tevac/100$	$100P_L$	$10P_I$	v_{feed}
0.500000	0.880487	0.347208	0.601150	0.999232	1.89142	0.500000	0.700000	0.229774
0.500000	0.700000	0.228040	0.930513	0.342909	1.257679	0.500000	0.700000	0.228040
0.779682	0.700341	0.281475	0.833384	1.700676	1.261054	0.500000	0.700000	0.294035
0.684084	0.700000	0.611681	0.391217	1.149653	1.635953	0.500000	0.700000	0.632977
0.576999	0.700000	0.488979	0.572768	0.725114	1.472715	0.500000	0.700000	0.493956
0.657376	0.840852	0.740873	0.494670	0.755455	1.835819	0.500000	0.700000	0.608736
0.500000	1.001157	0.710936	0.673961	0.300000	2.000000	0.500000	0.700000	0.474782
0.500000	0.700000	0.926407	0.449104	1.051403	1.782759	0.500000	0.700000	0.926407
0.572962	0.700000	1.131029	0.450885	1.903985	2.000000	0.500000	0.700000	1.141140
0.515198	0.700000	1.387293	0.360134	0.300000	1.426807	0.500000	0.700000	1.388694
1.334989	0.700000	1.121346	0.411061	1.665885	1.988835	0.500000	0.700000	1.304453
0.500000	0.700000	0.891859	0.669812	1.677351	1.405318	0.500000	0.700000	0.891859
0.706421	0.700000	1.693612	0.361687	1.868023	1.959067	0.500000	0.700000	1.735485
0.647267	0.700000	1.613504	0.439415	0.868861	2.000000	0.500000	0.700000	1.638038
0.500000	0.727163	1.191254	0.685667	0.677946	1.892645	0.500000	0.700000	1.153165
0.817640	0.700000	1.017771	0.719202	0.332629	1.345802	0.507720	0.700000	1.043587

0.500000	0.726966	2.000000	0.431704	1.612660	1.892487	0.500000	0.700000	1.937147
0.660234	0.720004	1.265342	0.745474	1.487010	2.000000	0.500000	0.700000	1.249492
0.500000	0.702484	1.668763	0.571970	1.904343	1.981857	0.500000	0.700000	1.663736
0.890000	0.708067	0.881114	0.999932	1.630597	1.444556	0.533980	0.700000	0.897656
1.750526	0.770418	1.376968	0.806094	1.733595	2.000000	1.130510	0.705417	1.371464
1.361080	0.700000	1.512352	0.733896	1.844054	1.862760	0.654310	0.712248	1.650399

The units are as follows: $P_L\ [bar], P_I\ [bar], v_{feed}\ [m/s], t_{ads}\ [s], t_{bd}[s], t_{evac}[s]$.

A.1.6 Reproducing the multi-objective optimization result *via* TSEMO

As shown in Figure S3, I ran the TSEMO three times. In trial 1 (A1, A2) and trial 2 (B1, B2), I started the TSEMO from the same initial sampling points. In trial 3 (C1, C2), I initialized TSEMO from different initial sampling points. In the several initial iterations, the deviation of results is significant, while the deviation becomes smaller with the increase of iterations.

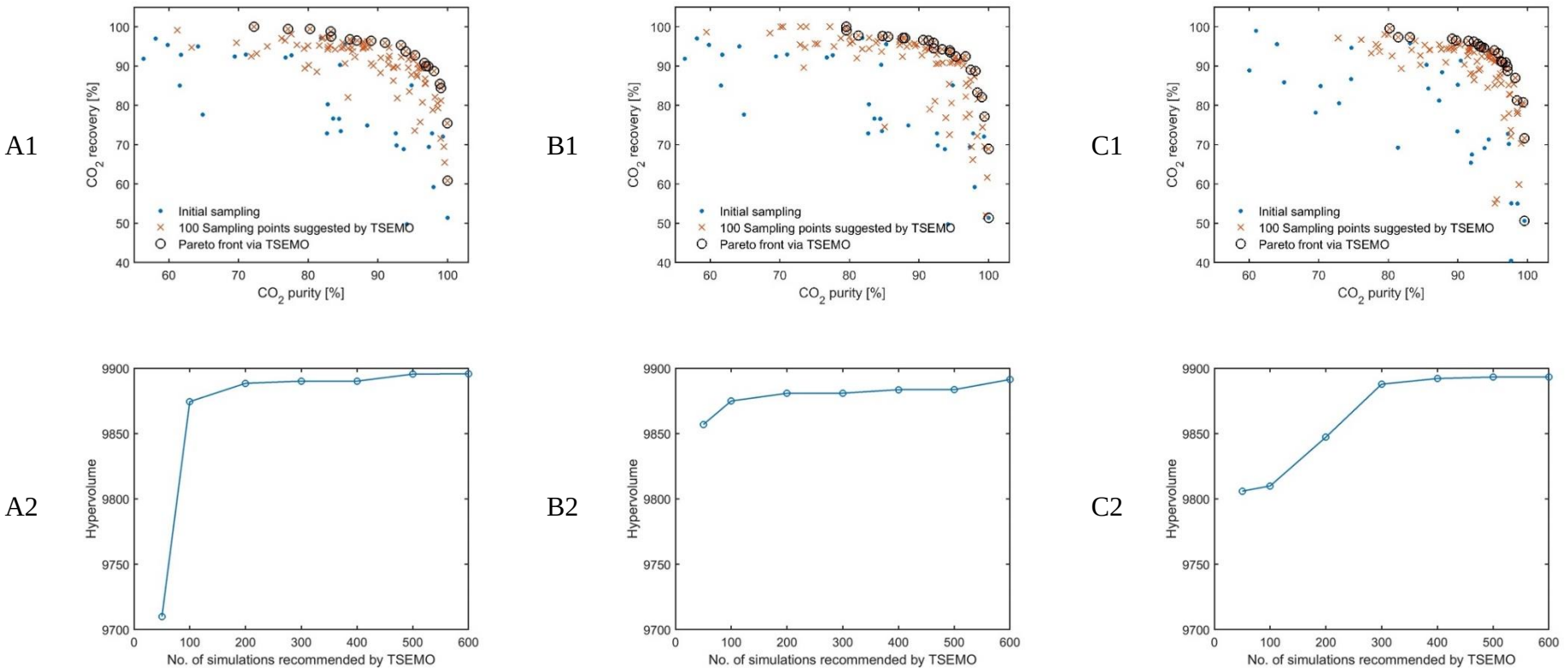


Figure S3. Multi-objective optimization of PSA *via* TSEMO. (A1, B1, C1) optimization results through 100 simulations recommended by TSEMO: to initialize TSEMO, LHS generated 30 simulations, shown as the blue points; the algorithm recommended additional 100 simulations, shown as the red crosses. The estimated Pareto front was evolved, shown as the black circles. (A2, B2, C2) hypervolume quantification (reference point is [0, 0]) varying from 50 to 600 simulations recommended by TSEMO.

A.2 Direct CO_2 hydrogenation to methanol (DCHM)

A.2.1 Reaction kinetics

The Graaf's kinetics [175] for DCHM is used in the reactor modelling, whereas the kinetic parameters come form An *et al.* [93].

Table S4. Reaction kinetics for CO_2 hydrogeneration to methanol [175].

Kinetics	r_1: CO + 2H₂ ↔ CH₃OH r_2: CO₂ + 3H₂ ↔ CH₃OH + H₂O r_3: CO₂ + H₂ ↔ CO + H₂O (RWGS)
Rate expression from Graaf *et al.* [27]	$r_1 = \frac{k_1 K_{CO}\left[p_{CO}p_{H_2}^{1.5} - \frac{p_{CH_3OH}}{p_{H_2}^{0.5}K_{eq1}}\right]}{\left(1 + K_{CO}p_{CO} + K_{CO_2}p_{CO_2}\right)\left[p_{H_2}^{0.5} + \frac{K_{H_2O}}{K_{H_2}^{0.5}}p_{H_2O}\right]}$ $r_2 = \frac{k_2 K_{CO_2}\left[p_{CO_2}p_{H_2}^{1.5} - \frac{p_{CH_3OH}p_{H_2O}}{p_{H_2}^{1.5}K_{eq2}}\right]}{\left(1 + K_{CO}p_{CO} + K_{CO_2}p_{CO_2}\right)\left[p_{H_2}^{0.5} + \frac{K_{H_2O}}{K_{H_2}^{0.5}}p_{H_2O}\right]}$ $r_3 = \frac{k_3 K_{CO_2}\left[p_{CO_2}p_{H_2} - \frac{p_{CO}p_{H_2O}}{K_{eq3}}\right]}{\left(1 + K_{CO}p_{CO} + K_{CO_2}p_{CO_2}\right)\left[p_{H_2}^{0.5} + \frac{K_{H_2O}}{K_{H_2}^{0.5}}p_{H_2O}\right]}$

Where $K_{eq1}, K_{eq2}, K_{eq3}$ are equilibrium constants of three reactions; $K_{CO}, K_{CO_2}, K_{H_2}, K_{H_2O}$ are adsorption constants of components; k_1, k_2, k_3 are rate constants; p_i refers to partial pressure of component i.

The equilibrium constants listed in Table S4 are calculated from empirical expressions, see Eqs. S1-S3.

$$K_{eq2} = K_{eq1} \cdot K_{eq3} \quad \text{S1}$$

$$\log_{10}(K_{eq1}) = \frac{5139}{T} - 12.621 \quad \text{S2}$$

$$\log_{10}(K_{eq3}) = \frac{-2073}{T} + 2.029 \quad \text{S3}$$

The rate constants are based on the Arrhenius equation, see Eq. S4. The adsorption constants are obtained by Eq. S5, and the relevant values come from An's work [93], see Table S5.

$$k_i = A_i \exp\left(\frac{-E_i}{RT}\right) \quad \text{S4}$$

$$K_i = B_i \exp\left(\frac{-\Delta_b H_i}{RT}\right) \quad \text{S5}$$

Table S5. Values for kinetic parameters [93].

Symbol	A_i or B_i	E_i or $\Delta_b H_i$
k_1	4.06×10^{-6}	1.17×10^{4}
k_2	1.52×10^{-33}	2.66×10^{5}
k_3	9.04×10^{8}	1.13×10^{5}
K_{CO}	8.40×10^{-11}	-1.18×10^{5}
K_{CO_2}	1.72×10^{-10}	-8.13×10^{4}
$K_{H_2O}K_{H_2}^{-0.5}$	4.37×10^{-12}	-1.15×10^{5}

A.2.2 A simulation result

I present a simulation result of flowrates and utility consumptions as shown in Table S6 - Table S7. This simulation is based on the input [F_{H_2},T_R, P_R,T_F, P_F, N_{trays}, $Split_{purge}$] = [1254.32 kmol/h, 236 ℃, 55 bar, 36 ℃, 48 bar, 24, 0.0037]. A full Aspen Plus simulation can contain much more information than shown here, but those might not be relevant in the process analysis and optimization. As a result, the execution of rigorous simulation can bring in excessive computation costs in the iteration of simulation-based optimization.

Table S6. One simulation result for the flowrates of MS by direct CO_2 hydrogenation.

		Inlet flows		**Outlet flows**			
	Units	CO2	H2	MEOH	PURGE	VENT	WATER
CO	kmol/h	0	0	0	0.22	0	0
CO_2	kmol/h	390	0	0	0.73	0	0
H_2	kmol/h	0	1254.32	0.02	86.17	0.77	0
H_2O	kmol/h	0	3	3.87	0.06	0	388.34
CH_3OH	kmol/h	0	0	384.87	0.30	0.11	3.78
Total	kmol/h	390.00	1257.32	388.76	87.48	0.88	392.12
flow	kg/h	17163.90	2577.51	12401.44	222.20	5.15	7116.98
$\overline{Mw}$	kg/kmol	44.01	2.05	31.90	2.54	5.85	18.15

Simulation input [F_{H_2},T_R, P_R,T_F, P_F, N_{trays}, $Split_{purge}$] = [1254.32 kmol/h, 236 ℃, 55 bar, 36 ℃, 48 bar, 24, 0.0037]

Table S7. One simulation result for the utility of MS by direct CO_2 hydrogeneration.

	Unit	Cooler	Heater	Flash	Compressor	Condenser	Reboiler	Reactor
Duty	Gcal/h	-15.2633	18.6137	-11.2195	0.787219	-6.48332	6.13132	-5.441

Simulation input [F_{H_2},T_R, P_R,T_F, P_F, N_{trays}, $Split_{purge}$] = [1254.32 kmol/h, 236 ℃, 55 bar, 36 ℃, 48 bar, 24, 0.0037]

A.3 Gas-to-liquid (GTL) process or [Reforming + FT]

As shown in Figure S4, the Gas-to-Liquid (GTL) process is modelled in Aspen Plus, by referring to the prior works of Ha *et al*. [109], Lee *et al*. [111] and Zhang *et al*. [30, 188]. This process starts with the combined reforming (CO_2 + H_2O) of natural gas to syngas, followed by FT synthesis for fuels. Since the upgrading section has little influence on the overall performance [109], I use a distillation column to simplify it. To deal with a petrochemical system like GTL, Peng-Robinson is recommended as the thermodynamic method [146].

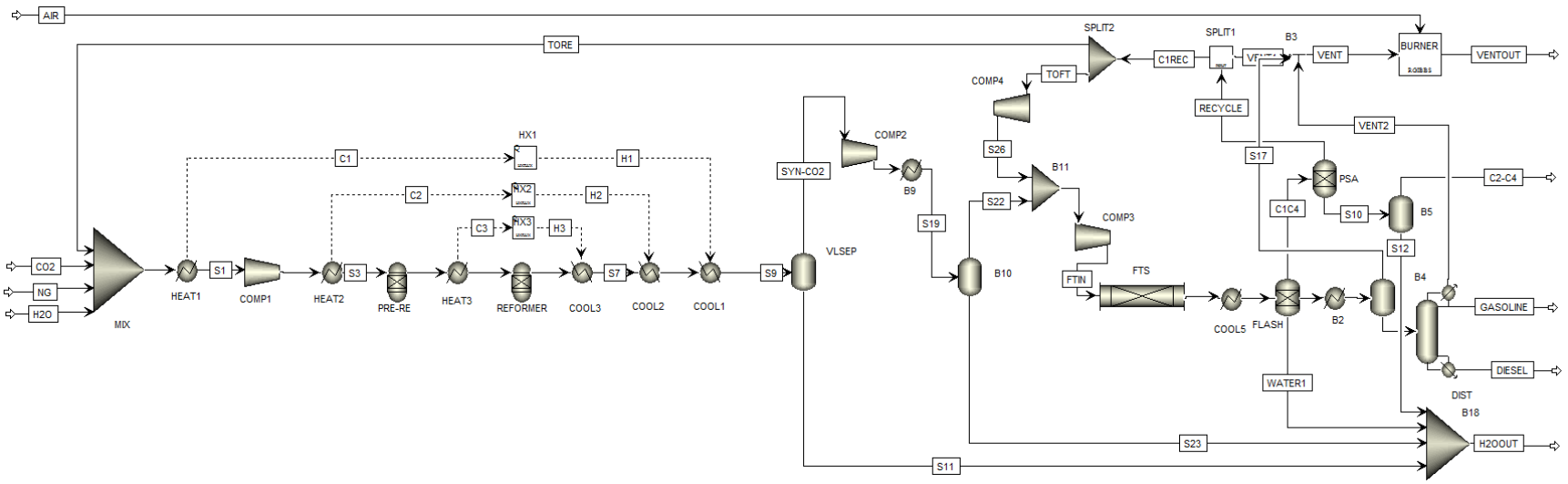


Figure S4. Flowsheet for GTL built in Aspen Plus. *HXFlux* is used to model the heat exchangers: unlike a typical *HeatX* involved with mass flows, *HXFlux* only deals with heat flows (dashed lines).

A.3.1 Combined reforming section

In the reforming section, GTL starts with NG, water and CO_2. A typical composition of NG can be referred to Bao's work (Table S8) [112]. The water is heated to steam before reforming. In the pre-reformer, all the carbon components are converted to CO and CH_4. In the reformer, CH_4 is converted to syngas with the assistance of CO_2 and steam. The reforming is performed at a high temperature between 700 and 1000 ℃, so it is assumed to reach equilibrium. In Aspen Plus, the reformer is modelled by an RGibbs reactor, where the total Gibbs energy is minimized to the reach the equilibrium ('Restricted Chemical Equilibrium' is set for the combined reforming reactions). A flowsheet option is set to vary the flowrate of H_2O and NG to guarantee the ratio of CO : H_2 falls in a range of 2 – 2.2 in the reformer outlet. Before the FT section, the mixed stream is cooled down and split into a gas stream (mainly syngas and CO_2) and a liquid stream (primarily water).

Table S8. The compositions of NG.

Components	Mol %	Range
CH_4	95.39	94 – 96
C_2H_6	3.91	*
C_3H_8	0.03	*
CO_2	0.59	*
N_2	0.08	*
Total	100	100

* CH_4% is regarded as an uncertainty in this work. When CH_4% changes, the ratio of C_2H_6 : C_3H_8 : CO_2 : N_2 is assumed to keep unchanged.

A.3.2 Fischer-Tropsch section

A.3.2.1 FT kinetics

For FT kinetics, Yates *et al*. developed a simple but reliable expression 30 years ago for the consumption rate of CO as Equation S6 [189].

$$r_{CO} = \frac{F\ a\ P_{CO}\ P_{H_2}}{(1 + b\ P_{CO})^2} \quad \text{S6}$$

where, F: catalyst improvement factor compared to the catalyst tested in 1991 (F=1);

P_i: partial pressure of component i;

a: reaction rate coefficient;

b: adsorption coefficient.

Vervloet *et al*. suggested the catalyst improvement factor could fall in the range of 1 – 10 [107], while, a decade ago, Guettel *et al*. claimed that a more promising value could be up to 20 in the future [106]. Here, I assume that the value of the catalyst improvement factor is 10 for this work. The variables a and b can be expressed by the following equations,

$$a = a_0 \exp\left[\frac{E_a}{R}\left(\frac{1}{493.15} - \frac{1}{T}\right)\right] \quad \text{S7}$$

$$b = b_0 \exp\left[\frac{\Delta_b H}{R}\left(\frac{1}{493.15} - \frac{1}{T}\right)\right] \quad \text{S8}$$

where, a_0: pre-exponential factor;

E_a: activation energy;

b_0: adsorption coefficient;

$\Delta_b H$: adsorption entahlpy.

Table S9 gives all the relevant values used in GTL.

Table S9. Values for FT kinetic parameters used in this work [103].

Symbol	Value	Unit	Note
F	10	-	Catalyst improvement factor
a_0	8.9e-3	$\mathrm{mol \cdot s^{-1} \cdot kg_{cat}^{-1} \cdot bar^{-2}}$	Pre-exponential factor
E_a	3.7e4	$\mathrm{J \cdot mol^{-1}}$	Activation energy
b_0	2.2	$\mathrm{bar^{-1}}$	Adsorption coefficient
$\Delta_b H$	-6.8e4	$\mathrm{J \cdot mol^{-1}}$	Adsorption enthalpy

A.3.2.2 Product distribution

Yates' kinetics only describes the consumption rate of CO, but no more information is given concerning the products. The Anderson-Schulz-Flory mechanism can be used to explain the distribution of FT products [103]. Since most FT products are linear hydrocarbons, the reaction can be regarded as the polymerization process to grow a long chain. Figure S5a demonstrates how a small carbon chain extends to a long chain. The chain growth probability, α, is a parameter to denote whether the chain continues to grow or not: α for growth, whereas $(1-\alpha)$ for termination. Figure S5b shows how α can determine the product distribution.

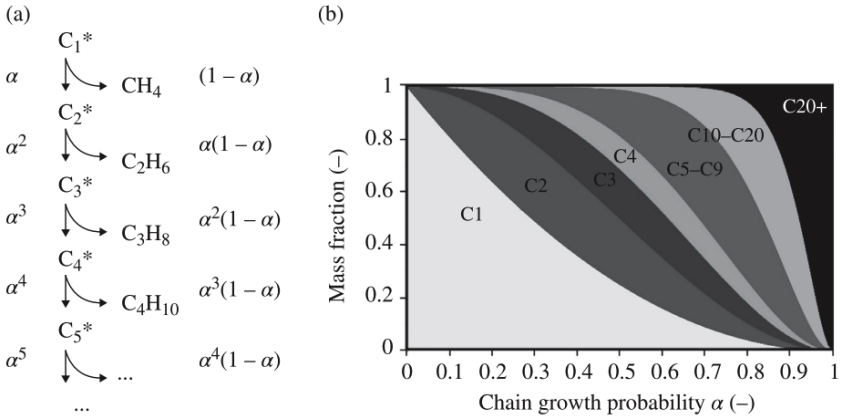


Figure S5. (a) Chain growth by the Anderson-Schulz-Flory mechanism. (b) The product distribution based on the α. This figure is taken from the cited work [103].

Typically, FT produces long hydrocarbons, which can be used as waxes and be easily cracked for short chains. Hence, the value of α should be high enough, and typical value can be around 0.9 [103]. In our work, I assumed that $\alpha = 0.93$ and plotted the corresponding product distribution as Figure S6.

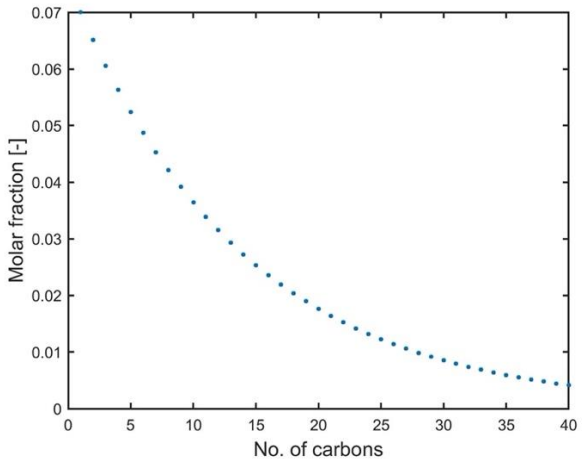


Figure S6. Molar distribution of FT products, assuming $\alpha = 0.93$.

Since the properties of hydrocarbons (HCs) are similar, I used several components to represent the whole range of HCs for simplification during the simulation as the Table S10. Based on the distribution of FT products, the molar fractions of representative HCs can be calculated from the sum of molar fraction of corresponding FT products. In this work, C_8H_{18} is used to approximate gasoline, while $C_{16}H_{34}$ is regarded as diesel. For further simplification, wax is not taken into consideration in the system. Since just a few components are considered, it is a simplified approach for the simulation of FT.

Table S10. Components used in the simulation of FT.

C range	Molar fraction	Representations
C_1	0.07	CH_4
$C_2 \sim C_4$	0.18	C_3H_8
$C_5 \sim C_{12}$	0.30	C_8H_{18} (gasoline)
C_{13+}	0.45	$C_{16}H_{34}$ (diesel)
Total	1.00	

In the FT section, the syngas is pressurized before entering the FT reactor. RPlug is chose to simulate the multi-tubular fixed-bed reactor for FT in Aspen Plus. Yates' kinetics is used for the overall consuming rate of CO (r_{CO}). Since the properties of hydrocarbons (HCs) are similar, I employed four reactions ($R1 \sim R4$) to represent the whole range of HCs for simplification during the simulation as the Table S11. The sum of CO consuming rates of individual reactions should be equal to the overall consuming rate of CO (r_{CO}). Then the individual reaction rates are obtained from the reaction stoichiometry. CO_2 is reported not to react on the Cobalt-based catalyst and can be regarded as an inert gas in the FT reaction [108, 110]. Thus, no reaction associated with CO_2 is listed in the FT reactor. Based on the information mentioned above, I inserted the kinetics into RPlug.

Table S11. Components and reactions used in the simulation.

C range	Molar fraction	Representations	Representative reactions	R_{CO}
C_1	0.07	CH_4	$R1$: $3H_2 + CO \rightarrow CH_4 + H_2O$	$0.007r_{CO}$
$C_2 \sim C_4$	0.18	C_3H_8	$R2$: $7H_2 + 3CO \rightarrow C_3H_8 + 3H_2O$	$0.053r_{CO}$
$C_5 \sim C_{12}$	0.30	C_8H_{18} (gasoline)	$R3$: $17H_2 + 8CO \rightarrow C_8H_{18} + 8H_2O$	$0.234r_{CO}$
C_{13+}	0.45	$C_{16}H_{34}$ (diesel)	R4: $33H_2 + 16CO \rightarrow C_{16}H_{34} + 16H_2O$	$0.706r_{CO}$
Total	1.00			r_{CO}

A.3.3 Separation and recovery

Following the FT reactor, a three-phase flash is used to split the mixed stream into gas, liquid HCs and wastewater. As a simplification to the upgrading system, I used a distillation column (RadFrac) to separate gasoline from diesel. For the gas mixture in the simulation, I used an ideal separator (in the real world, PSA can be an option) to recycle all the C1 components. The GTL system contains inert gas (N_2), which must be purged (otherwise, it will gradually accumulate in the recycle stream, making the convergence impossible to achieve). Herein, the recycle stream is split to *vent* (to purge ranging from 0.001 to 0.2 of the recycle stream) and *C1REC*, which is followed by splitting into reforming section and FT sections, respectively.

A.3.4 Utilities and their integration

The heating utility is supplied by high-pressure steam and fuel gas. The cooling utility is provided by air and cooling water. Pumps and compressors are powered by electricity.

The reforming is highly endothermic, and the reforming section can account for over 50% of energy consumption in the overall GTL process. A high temperature is required for the reforming reaction, where the heat is supplied by the combustion of natural gas or electricity. Thus, the reformer outflow has an extremely high temperature and needs to be cooled before the FT process. I built three heat exchangers to gradually cool down the reformer outflow, while the recycled heat is used to pre-heat the mixed stream to the gas form (>100 ℃), an intermediate temperature for pre-reformer (~ 500 ℃) and a high temperature for reformer (700 ~ 1000 ℃).

Additionally, the purge stream contains CO and CH_4, which will bring in considerable greenhouse emissions if the direct emissions apply. With the assistance of air, a burner is used to deal with these C1 components. An RGibbs reactor operated in 600 ℃ is used to simulate the burner. Due to the exothermic reactions, the burner will release heat, while the waste heat recovery technology[190, 191] can be used to recover partial heat from the burner (utilization efficiency is assumed at $\eta_{burner} = 60\%$) to reduce the heating utility of steam or fuel gas.

A.3.5 Convergence strategy

The modelled GTL system has two flowsheet options (design specifications) to vary the flow rates of H_2O and NG, as well as two recycle streams flowing to reforming/FT sections. As a result, the flowsheet convergence can be difficult. After trails and errors, I would suggest a two-stage strategy for a fast convergence for this case. In the first stage, the convergence of mass flows is performed. *Newton* algorithm is selected to converge the two design specifications, while *Broyden* or *Wegstein* algorithm is used to tear the stream *FTIN*. Because the recycle loop and design specifications can interact with each other in our case, the tear stream has to be solved simultaneously with the two design specifications (*Outside-Simultaneous* is recommended for sequencing). After the mass flow is converged, the heat flow is then converged in the second stage. *Wegstein* algorithm is used to tear the heating streams *H1, H2* and *H3,* which can be made as a convergence block. The heating convergence block is sequenced after the first-stage convergence. Meanwhile, this explains why I used *HXFlux* to model the heat exchangers. Unlike a typical *HeatX* involved with mass flows (solid lines), *HXFlux* only deals with heat flows (dashed lines as shown in Figure S4).

A.3.6 A simulation result of Aspen Plus for GTL

I present a typical simulation result of flowrates and utility consumptions as shown in Table S12 - Table S13. Simulation input is set as follows [F_{CO_2}, F_{CH_4}, x_{CH_4}, T_{FT}, P_{FT}, N_{trays}, $T_{reformer}$, $P_{reformer}$, $Split_{vent}$, $Split_{FT}$] = [6923 kmol/h, 18787 ℃, 0.96, 254 ℃, 20 bar, 55, 948 ℃, 4 bar, 0.086, 0.40]. For the inlet flow for FT, the ratio of H_2 : CO = 2.16 (FTIN) falls in the desired range for the FT reaction. A full Aspen Plus simulation can contain much more information than shown here, but those might not be relevant for optimization. This is a motivation to develop a reduced-order surrogate.

Table S12. One simulation result for the flowrates of GTL.

		Inlet flows				**Outlet flows**					**Inside**				
	Units	CO2	NG	H2O	AIR	C2-C4	GASOLINE	DIESEL	VENTOUT	H2OOUT	FTIN	FTOUT	RECYCLE	C1RECYLE	TOFT
CO	kmol/h	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	24911.10	0.00	0.00	0.00	0.00
H2	kmol/h	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.04	53701.40	1437.11	1436.92	1313.24	522.06
CO_2	kmol/h	6922.51	101.15	0.00	0.00	0.00	2.31	0.00	1773.30	1.02	18056.40	18056.40	17998.20	16449.00	6539.08
C_2H_6	kmol/h	0.00	670.31	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
C_3H_8	kmol/h	0.00	5.14	0.00	0.00	436.60	1.38	0.00	0.00	0.00	0.00	442.89	0.00	0.00	0.00
H_2O	kmol/h	0.00	0.00	50639.00	0.00	19.66	19.59	0.00	358.97	61413.20	462.56	25374.00	0.00	0.00	0.00
C_8H_{18}	kmol/h	0.00	0.00	0.00	0.00	159.16	553.86	0.00	0.00	0.00	0.00	728.81	0.00	0.00	0.00
$C_{16}H_{34}$	kmol/h	0.00	0.00	0.00	0.00	0.13	5.49	1093.19	0.00	0.00	0.00	1098.83	0.00	0.00	0.00
CH_4	kmol/h	0.00	17996.30	0.00	0.00	0.00	0.00	0.00	0.00	0.00	147.06	318.05	317.76	290.41	115.45
N_2	kmol/h	0.00	13.71	0.00	1277.28	0.00	0.00	0.00	1290.98	0.00	158.45	158.45	158.38	144.75	57.54
O_2	kmol/h	0.00	0.00	0.00	339.53	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.00
Total	kmol/h	6922.51	18786.60	50639.00	1616.81	615.54	582.63	1093.19	3423.27	61414.30	97436.90	47614.50	19911.30	18197.40	7234.14
$\overline{Mw}$	kg/kmol	44.01	16.71	18.02	28.85	61.44	111.61	226.45	35.25	18.02	16.58	33.94	40.41	40.41	40.41

Table S13. One simulation result for the utility of GTL.

Utility ID	Unit	ELE	Steam	U1000	WATER	FT reactor	AIR	Burner
Utility type		Electricity	Steam	Fuel gas	Cooling water	Cooling water	Cooling air	Waste heat recovery
Duty	GJ/h	1356.58	195.18	4513.24	2346.89	3637.57	30.61	67.15

Simulation input for Table S12 - Table S13 is set as $[F_{CO_2}, F_{CH_4}, x_{CH_4}, T_{FT}, P_{FT}, N_{trays}, T_{reformer}, P_{reformer}, Split_{vent}, Split_{FT}]$ = [6923 kmol/h, 18787 ℃, 0.96, 254 ℃, 20 bar, 55, 948 ℃, 4 bar, 0.086, 0.40].

A.3.7 An example of one input-output data point for GTL surrogate

According to the simulation result as shown in Table S12 - Table S13, I can collect one input-output data point as shown in Table S14. This data point has much less information than an Aspen Plus simulation result, but it has contained the essential features for the GTL process regarding the operating condition and corresponding process output. A sufficient number of such effective data points are required to train a surrogate for the GTL process.

Table S14. An example of one input-output data point for GTL surrogate.

Input variables	Value	Unit	Notes
F_{CO_2}	6923	[kmol/h]	Inlet flowrate of CO_2
F_{NG}	18787	[kmol/h]	Inlet flowrate of natural gas (NG)
x_{CH_4}	0.96	[-]	Inlet molar fraction of CH_4
T_{FT}	253	[℃]	Temperature in FT reactor
P_{FT}	20	[bar]	Pressure in FT reactor
N_{trays}	55	[-]	No. of trays in distillation column
$T_{reformer}$	948	[℃]	Temperature in reformer reactor
$P_{reformer}$	4	[bar]	Pressure in reformer reactor
$Split_{vent}$	0.086	[-]	Split fraction to vent stream (the other goes to recycle)
$Split_{FT}$	0.40	[-]	Split fraction to FT reactor (the other goes to reformer)
Output variables			
$F_{gasoline}$	713	[kmol/h]	Sum of [C_8H_{18}] in outlet flows of C2-C4, GASOLINE, DIESEL
F_{diesel}	1099	[kmol/h]	Sum of [$C_{16}H_{34}$] in outlet flows of GASOLINE, DIESEL
F_{gas}	438	[kmol/h]	Sum of [C_3H_8] in all outlet flows of C2-C4, GASOLINE
$F_{H_2O_{net}}$	11172	[kmol/h]	Outflow [H_2O] – inflow [H_2O]
$vent_{CO_2}$	1773	[kmol/h]	Flowrate of CO_2 in VENTOUT
Electricity	1357	[GJ/h]	Electricity consumption for pumps and compressors
U_{air}	31	[GJ/h]	Cooling by air
U_{1000}	4473	[GJ/h]	Heating by 1000 ℃ fuel gas – waste heat recovery· η_{burner}
U_{Steam}	195	[GJ/h]	Heating by high-pressure steam
U_{water}	5984	[GJ/h]	Cooling by cooling water

A.4 [Reforming + MS]

The [Reforming + MS] sub-system contains two sections: (1) combined reforming, which generates the syngas with the composition ratio as $\frac{2CO+3CO_2}{H_2} = 1$ (Figure S7); (2) methanol synthesis (MS) converted the syngas to methanol. The MS process model is taken out from the Aspen Plus model library [178].

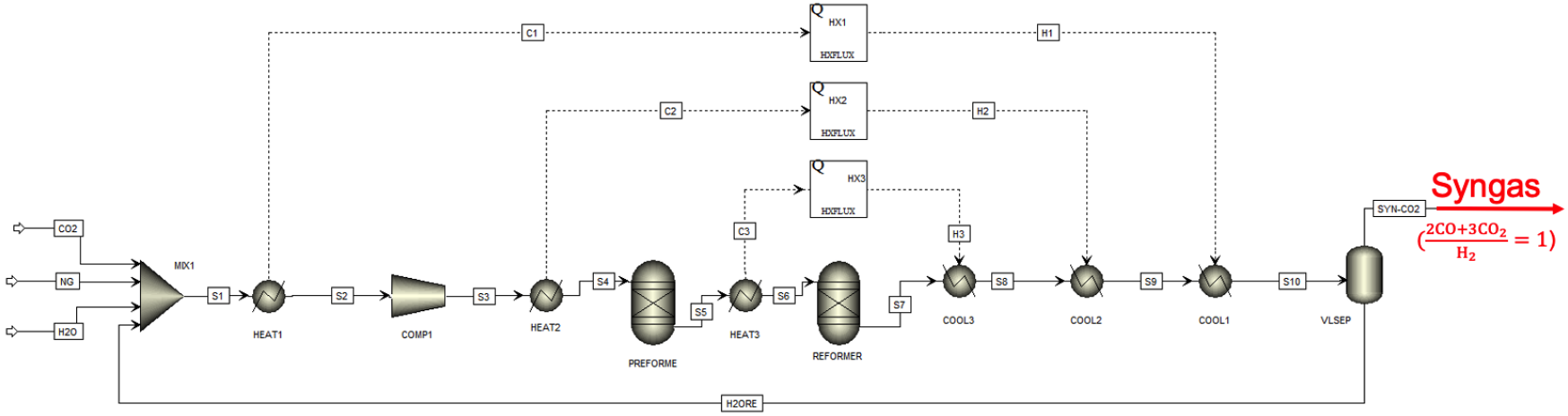


Figure S7. Reforming section for [Reforming + MS].

A.4.1 Combined reforming

The combined reforming section is similar to Appx.A.3.1; flowsheet options are set to manipulate the flowrates of NG and inlet water, in order to guarantee the optimal syngas ratio as $\frac{2CO+3CO_2}{H_2} = 1$ for MS reaction (Figure S7).

A.4.2 Methanol synthesis

For the MS section, the Aspen Plus Model library offers an industrial-scale process model, where the thermodynamics, reaction kinetics and mass balance are validated. The process model is reported to be capable of simulating the most common industrial methanol process - ICI Synetix low pressure methanol process (LPM). A four-stage quench reactor is used to perform the heat integration between the inlet stream and exothermic MS reactions. More detailed information about the process models can be referred to the Aspen Plus documents [178].

$$CO_2 + 3H_2 \leftrightarrow CH_3OH + H_2O$$

$$CO_2 + H_2 \leftrightarrow CO + H_2O \text{ (RWGS)}$$

Vanden Bussche and Froment kinetics [176] are used for MS.

$$r_{MS} = \frac{k_{MS} p_{CO_2} p_{H_2} \left(1 - \frac{p_{CH_3OH} p_{H_2O}}{K_{MS} p_{H_2}^3 p_{CO_2}}\right)}{\left(1 + K_A \frac{p_{H_2O}}{p_{H_2}} + K_B p_{H_2}^{0.5} + K_C p_{H_2O}\right)^3} (W_{cat} F_{cat}) \quad \text{S9}$$

$$r_{RWGS} = \frac{k_{RWGS} p_{CO_2} \left(1 - \frac{p_{CO} p_{H_2O}}{K_{RWGS} p_{CO_2} p_{H_2}}\right)}{\left(1 + K_A \frac{p_{H_2O}}{p_{H_2}} + K_B p_{H_2}^{0.5} + K_C p_{H_2O}\right)} (W_{cat} F_{cat}) \quad \text{S10}$$

where the rate constants can be expressed as follows:

$$k_{MS} = k_{MS,ref} \exp\left(-\frac{E_{MS}}{R}\left(\frac{1}{T} - \frac{1}{T_{ref}}\right)\right) \quad \text{S11}$$

$$k_{RWGS} = k_{RWGS,ref}\exp\left(-\frac{E_{RWGS}}{R}\left(\frac{1}{T}-\frac{1}{T_{ref}}\right)\right) \quad \text{S12}$$

where, r_{MS}: rate of methanol synthesis, kmol/s;

p_i: partial pressure of component i, bar ($i = H_2, H_2O, CH_3OH, CO, CO_2$);

k_j: rate constant of reaction j, kmol/kg-cat/s ($j = MS, RWGS$)

T_{ref}: reference temperature, 501.57 K

$k_{j,ref}$: rate constant of reaction j at T_{ref}, kmol/kg-cat/s ($j = MS, RWGS$)

E_j: activation energy of reaction j, kmol/kg-cat/s ($j = MS, RWGS$)

K_j: equilibrium constant of reaction j ($j = MS, RWGS$), expressed as $\ln K_j = A_j + \frac{B_j}{T}$

W_{cat}: rate constant of reaction j, kmol/kg-cat/s ($j = MS, RWGS$)

F_{cat}: catalyst activity factor, F_{cat}=1 at fresh catalyst.

A.4.3 Separation and recovery

I did not modify the separation part of the original process model. More detailed information about the process models can be referred to the Aspen Plus document [178].

A.4.4 Utilities and their integration

This section is the same as GTL in the Appx.A.3.4.

Appendix B Supplementary information for the large CCU system

Appendix B provides the supplementary information mainly for Chapter 6. Appendix B delivers emissions and economic quantification, followed by single/multi-objective optimization of the whole CCU system.

B.1 Evaluation of GHG emissions for the industrial park

B.1.1 System boundary in this work: cradle-to-gate

The GHG emissions for the whole CCU system are evaluated based on the life cycle assessment (LCA). LCA is mainly used in comparative assessments. The cradle-to-gate (from raw materials to manufacturing) is sufficient to compare the emissions for different process configurations, because the downstream emissions are identical [76]. The cradle-to-gate approach for the CCU system will quantify the GHG emissions of raw materials and utility as well as CO_2 emissions in the process (*e.g.*, uncaptured CO_2 and CO_2 emissions via vent gas).

B.1.2 System expansion strategy to compare CCU with a reference process.

The ‘system expansion’ strategy is particularly useful for LCA of a system with multiple functions [76]. The CCU system achieves multiple functions, *i.e.*, the co-production of several fuels and the generation of low-carbon electricity. For a proper comparison, the ‘system expansion’ strategy expands a reference process to include all the functions as the original process. As shown in Figure S8, the reference system can generate electricity (no capture technology) and fuels by conventional process. Meanwhile, the amount of electricity generation and the production of fuels are equivalent in the two systems for a fair comparison.

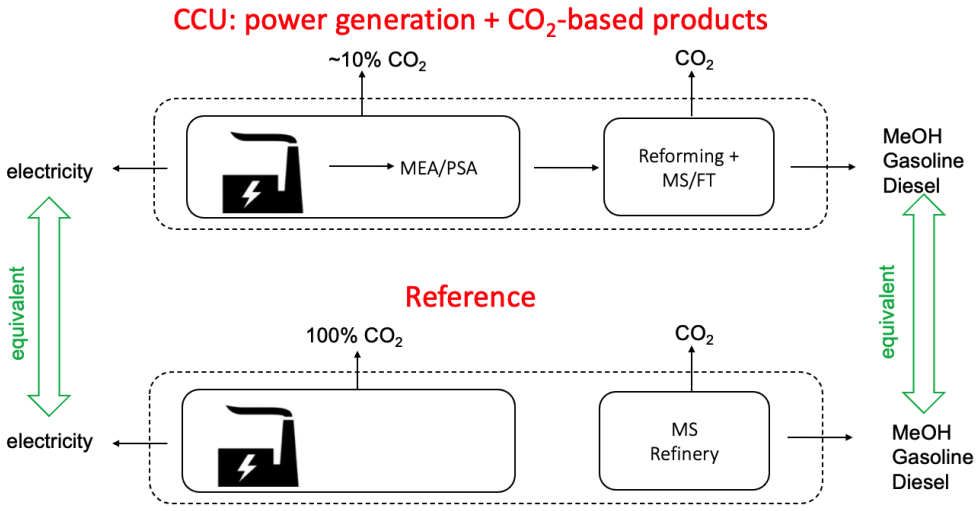


Figure S8. Expansion strategy for a fair comparison between CCU system and a conventional system.

B.1.3 The reduction of GHG emissions

Based on Figure S9, GHG emissions reduction can be calculated in Eqs. S13-S15.

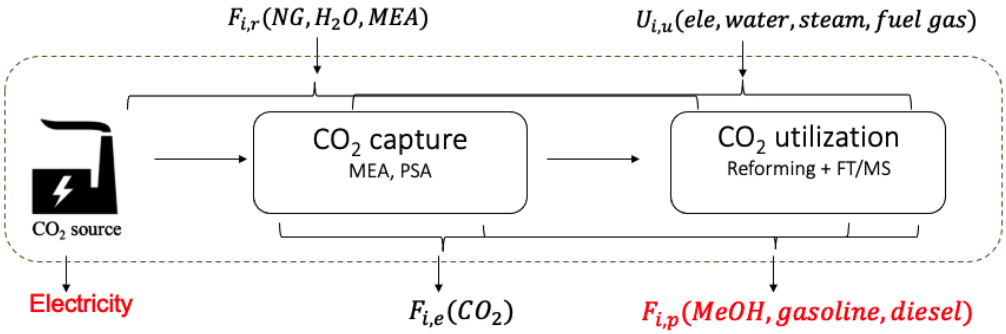


Figure S9. Sources of GHG emissions in the cradle-to-gate system boundary.

$$GHG_{CCU} = \sum_{i}\sum_{r} \alpha_r \cdot F_{i,r} + \sum_{i}\sum_{u} \alpha_u \cdot U_{i,u} + \sum_{i} F_{i,CO_2} \quad \text{S13}$$

$$GHG_{ref} = \alpha_{NGCC} \cdot E_{electricity} + \sum_{i}\sum_{p} \alpha_p \cdot F_{i,p} \quad \text{S14}$$

$$\mathrm{GHG_{reduction}} = 1 - \frac{\mathrm{GHG_{CCU}}}{\mathrm{GHG_{ref}}} \quad \text{S15}$$

where

F: mass flow, $\mathrm{ton/hour}$

U: utility, $\mathrm{GJ/hour}$

α_r: emission factor per raw material r generation: $\mathrm{ton_{CO_2eq}/ton_r}$

α_u: emission factor per utility u generation: $\mathrm{ton_{CO_2eq}/GJ}$

α_p: emission factor per product p generation: $\mathrm{ton_{CO_2eq}/ton_p}$

Subscript

i: notation for sub-systems

r: notation for raw materials (natural gas, process water, MEA)

u: notation for utilities (steam, fuel gas, electricity, cooling water)

p: notation for products.

B.1.4 Calculation of emission factors for low-carbon electricity

In this work, the emission factor is defined as the GHG emission in generating a material (per ton) or a utility (per GJ). In the proposed industrial park, two 500 MW power stations generate electricity by natural gas combined cycle (NGCC) technology. When no carbon capture is applied, as shown in Equation S16, the emission factor per unit of electricity generation is calculated by the sum of emissions in generating raw materials (NG, water) together with the direct emissions. As such, the emission factor of NGCC electricity is calculated as 0.41 $\mathrm{kg_{CO_2eq}/kWh}$, which falls in the range reported by Weisser [187].

$$\alpha_{NGCC} = \frac{\sum_r \alpha_r \cdot F_{NGCC,r} + F_{NGCC,CO_2e}}{E_{NGCC}} \quad \text{S16}$$

In our work, the low-carbon electricity specifically refers to CCS electricity (NGCC integrated with CCS). MEA decarbonizes one NGCC, while PSA decarbonizes the other one. The emission factors of low-carbon electricity are approximated by the average value between [NGCC-PSA-storage] and [NGCC-MEA-storage]. Eqs S17 - S19 show how the emission factor for low-carbon electricity is calculated. As such, the emission factor of low-carbon electricity is estimated at 0.098 kg_{CO_2}/kwh (value may slightly change subject to the amount of CO_2 captured), which is in agreement with the literature value [187].

$$\alpha_{CCS} = (\alpha_{PSA} + \alpha_{MEA})/2 \quad \text{S17}$$

$$\alpha_{PSA} = \frac{\sum_r \alpha_r \cdot F_{NGCC,r} + F_{PSA,CO_2e}}{E_{NGCC-PSA}} \quad \text{S18}$$

$$\alpha_{MEA} = \frac{\sum_r \alpha_r \cdot F_{NGCC,r} + \sum_r \alpha_r \cdot F_{MEA,r} + F_{MEA,CO_2e}}{E_{NGCC-MEA}} \quad \text{S19}$$

E_{NGCC}: power generation of NGCC, GJ/h

$E_{NGCC-PSA}$: net power out for a NGCC coupled with PSA, GJ/h

$E_{NGCC-MEA}$: net power out for a NGCC coupled with MEA, GJ/h

F_{PSA,CO_2e}: emissions for a NGCC coupled with PSA, ton_{CO_2}/h

F_{MEA,CO_2e}: emissions for a NGCC coupled with MEA, ton_{CO_2}/h

F_{PSA,CO_2s}: the amount of stored CO_2 for a NGCC coupled with PSA, ton_{CO_2}/h

F_{MEA,CO_2s}: the amount of stored CO_2 for a NGCC coupled with MEA, ton_{CO_2}/h

$F_{MEA,r}$: mass flow of raw materials (MEA, H_2O) in MEA absorption process, $ton/hour$

Subscript:

CO_{2e}, CO_2 emissions to environment

CO_{2s}, CO_2 storage to underground.

B.1.5 Calculation of GHG emission factor for low-carbon heating

In this work, heating is proposed to be partially substituted by CCS electricity. As such, low-carbon heating is used in the CCU system. Herein, the GHG emission factor for low-carbon heating is calculated as follows,

$$\alpha_{\mathrm{fuel}_{\mathrm{low-C}}} = \epsilon \cdot \alpha_{\mathrm{CCS}} + (1-\epsilon)\alpha_{\mathrm{fuel}} \quad \text{S20}$$

$$\alpha_{\mathrm{steam}_{\mathrm{low-C}}} = \varepsilon \cdot \alpha_{\mathrm{CCS}} + (1-\varepsilon)\alpha_{\mathrm{steam}} \quad \text{S21}$$

where,

$\alpha_{\mathrm{fuel}_{\mathrm{low-C}}}$: GHG emission factor for low-carbon heating (partially by CCS electricity and partially by fuel gas), ton/GJ

α_{fuel}: GHG emission factor for heating by fuel gas, ton/GJ

$\alpha_{\mathrm{steam}_{\mathrm{low-C}}}$: GHG emission factor for low-carbon heating (partially by CCS electricity and partially by steam), ton/GJ

α_{steam}: GHG emission factor for heating by steam, ton/GJ

B.1.6 Data for GHG emissions factors

α_{r}: emission factor per raw material r generation: $\mathrm{ton}_{\mathrm{CO_2eq}}/\mathrm{ton}_{\mathrm{r}}$

α_{u}: emission factor per utility u generation: $\mathrm{ton}_{\mathrm{CO_2eq}}/\mathrm{GJ}$

α_{p}: emission factor per product p generation (in reference process): $\mathrm{ton}_{\mathrm{CO_2eq}}/\mathrm{ton}_{\mathrm{p}}$

Table S15. GHG emissions factors for materials (α_r or α_p) and utilities (α_u).

	GHG emissions factors	Unit	Sources
Natural gas	0.354	$\mathrm{ton}_{\mathrm{CO_2eq}}/\mathrm{ton}_{\mathrm{NG}}$	[32]
Process water	5.4e-4	$\mathrm{ton}_{\mathrm{CO_2eq}}/\mathrm{ton}_{\mathrm{water}}$	[32]
Methanol	0.762	$\mathrm{ton}_{\mathrm{CO_2eq}}/\mathrm{ton}_{\mathrm{MEOH}}$	[32]
Gasoline	0.802	$\mathrm{ton}_{\mathrm{CO_2eq}}/\mathrm{ton}_{\mathrm{gasoline}}$	[32]
Diesel	0.663	$\mathrm{ton}_{\mathrm{CO_2eq}}/\mathrm{ton}_{\mathrm{diesel}}$	[32]
MEA	3.40	$\mathrm{ton}_{\mathrm{CO_2eq}}/\mathrm{ton}_{\mathrm{MEA}}$	[192]
Ethanol	3.74	$\mathrm{ton}_{\mathrm{CO_2eq}}/\mathrm{ton}_{\mathrm{EtOH}}$	[193]
C2-C4	1.11	$\mathrm{ton}_{\mathrm{CO_2eq}}/\mathrm{ton}_{\mathrm{C3}}$	*
Electricity	0.114	$\mathrm{ton}_{\mathrm{CO_2eq}}/\mathrm{GJ}_{\mathrm{NGCC}}$	Eq. S16
Fuel gas	0.079	$\mathrm{ton}_{\mathrm{CO_2eq}}/\mathrm{GJ}_{\mathrm{fuel-gas}}$	[32]
Steam	0.083	$\mathrm{ton}_{\mathrm{CO_2eq}}/\mathrm{GJ}_{\mathrm{steam}}$	[32]
Cooling water	8.04e-3	$\mathrm{ton}_{\mathrm{CO_2eq}}/\mathrm{GJ}_{\mathrm{cooling}}$	**

*Average the emission factors of propene and propane in the software Umberto (method: ReCiPe Midpoint (H) w/o LT).

**The emission factor of cooling is calculated by water emission factor times its required amount (based on $\Delta T = 20℃$, heat transfer efficiency $\eta = 0.8$).

B.2 Evaluation of economic aspect for the industrial park

B.2.1 The calculation of profit

$$\mathrm{Profit_{CCU}} = -\sum_{i}\sum_{r}\beta_r \cdot F_{i,r} - \sum_{i}\sum_{u}\beta_u \cdot U_{i,u} - \sum_{i} F_{i,CO2} \cdot \gamma_{CO_2} + \beta_{CCS} \cdot E_{electricity} + \sum_{i}\sum_{p}\beta_p \cdot F_{i,p} \quad \text{S22}$$

where

F: mass flow, ton/hour

U: utility, GJ/hour

β_r: cost of raw material r, \$/ton

β_u: cost of utility u, \$/GJ

γ_{CO_2}: carbon tax (carbon price), $\$/\mathrm{ton_{CO_2}}$

β_{CCS}: cost of low-carbon electricity (equivalent to CCS electricity), \$/GJ

β_p: price of product p, \$/ton

Subscript

i: notation for sub-systems

r: notation for raw materials (natural gas, process water, MEA)

u: notation for utilities (steam, fuel gas, electricity, cooling water)

p: notation for products.

B.2.2 Calculation of economic factors for low-carbon electricity

Economic factors refer to the costs of raw materials and utilities as well as the prices of products. The emission factor of low-carbon electricity is approximated by the average value between [NGCC-PSA-storage] and [NGCC-MEA-storage].

$$\beta_{CCS} = (\beta_{PSA} + \beta_{MEA})/2 \quad \text{S23}$$

$$\beta_{PSA} = \frac{\beta_{NGCC} \cdot E_{NGCC} + \gamma_{CO_2} \cdot F_{PSA,CO_2e} + \delta_{CO_2} \cdot F_{PSA,CO_2s}}{E_{NGCC-PSA}} \quad \text{S24}$$

$$\beta_{MEA} = \frac{\beta_{NGCC} \cdot E_{NGCC} + \sum_r \beta_r \cdot F_{MEA,r} + \gamma_{CO_2} \cdot F_{MEA,CO_2e} + \delta_{CO_2} \cdot F_{MEA,CO_2s}}{E_{NGCC-MEA}} \quad \text{S25}$$

β_{CCS}: cost of low-carbon electricity (CCS electricity), $\$/GJ$

β_{PSA}: cost of electricity from [NGCC-PSA-storage] (the value can slightly change based on the carbon price and the amount of captured CO_2), $\$/GJ$

β_{NGCC}: cost of electricity from NGCC, $\$/GJ$

β_{MEA}: cost of electricity from [NGCC-MEA-storage] (the value can slightly change based on the carbon price and the amount of captured CO_2), $\$/GJ$

δ_{CO_2}: cost of the CO_2 transportation and storage in underground (obtained in IECM, assuming a 50 km pipeline is used for transportation), $\$/ton_{CO_2s}$

E_{NGCC}: electricity generation from NGCC, GJ/h

F_{PSA,CO_2e}: emissions for a NGCC coupled with PSA, ton_{CO_2}/h

F_{PSA,CO_2s}: the amount of stored CO_2 for a NGCC coupled with PSA, ton_{CO_2}/h

F_{MEA,CO_2e}: emissions for a NGCC coupled with MEA, ton_{CO_2}/h

F_{MEA,CO_2s}: the amount of stored CO_2 for a NGCC coupled with MEA, ton_{CO_2}/h.

B.2.3 Calculation of economic factors for low-carbon heating

In this work, heating is proposed to be partially substituted by CCS electricity. As such, low-carbon heating is used in the CCU system. Herein, the economic factor for low-carbon heating is calculated as follows,

$$\beta_{\mathrm{fuel_{low-c}}} = \epsilon \cdot \beta_{\mathrm{CCS}} + (1-\epsilon)\beta_{\mathrm{fuel}} \quad \text{S26}$$

$$\beta_{\mathrm{steam_{low-c}}} = \varepsilon \cdot \beta_{\mathrm{CCS}} + (1-\varepsilon)\beta_{\mathrm{steam}} \quad \text{S27}$$

where,

$\beta_{\mathrm{fuel_{low-c}}}$: economic factor for low-carbon heating (partially by CCS electricity and partially by fuel gas), ton/GJ

β_{fuel}: economic factor for heating by fuel gas, ton/GJ

$\beta_{\mathrm{steam_{low-c}}}$: economic factor for low-carbon heating (partially by CCS electricity and partially by steam), ton/GJ

β_{steam}: economic factor for heating by steam, ton/GJ

B.2.4 Calculation of economic factors involved with carbon tax

The economic factors contain two parts: original prices and carbon tax.

$$\mathrm{B_r} = \beta_{\mathrm{r,0}} + \alpha_{\mathrm{r}} \cdot \gamma_{\mathrm{CO_2}} \quad \text{S28}$$

$$\beta_{\mathrm{u}} = \beta_{\mathrm{u,0}} + \alpha_{\mathrm{u}} \cdot \gamma_{\mathrm{CO_2}} \quad \text{S29}$$

$$\beta_{\mathrm{p}} = \beta_{\mathrm{p,0}} + \alpha_{\mathrm{p}} \cdot \gamma_{\mathrm{CO_2}} \quad \text{S30}$$

where

β	Economic factors, $\$/\mathrm{ton_r}$

α	Emission factors, $\mathrm{ton_{CO_2}/ton_p}$
γ_{CO_2}	Carbon price, $\$/\mathrm{ton_{CO_2}}$
Subscript	
i	Notation for sub-systems
r	Notation for raw materials (natural gas, process water, MEA, *etc.*)
u	Notation for utilities (steam, fuel gas, electricity, cooling, *etc.*)
p	Notation for products (gasoline, diesel, methanol, *etc.*)
0	Notation for original price.

B.2.5 Data for economic factors

I used the prices data for the key materials in the first half of 2021.

Table S16. Economic factors for materials ($\beta_{r,0}$ or $\beta_{p,0}$), utilities ($\beta_{u,0}$), CO_2 storage (δ_{CO_2}).

	Economic factors [$ / ton]	Unit	Sources
Natural gas	475.4	$\$/\text{ton}_{\text{NG}}$	EU price in the first half of 2021 [194]
Process water	0.036	$\$/\text{ton}_{\text{water}}$	[195]
Methanol	475.6	$\$/\text{ton}_{\text{MEOH}}$	EU price in the first half of 2021 [145]
Gasoline	2254.8	$\$/\text{ton}_{\text{gasoline}}$	EU price in April 2021 [196]
Diesel	1808.6	$\$/\text{ton}_{\text{diesel}}$	EU price in April 2021 [196]
MEA	1100.0	$\$/\text{ton}_{\text{MEA}}$	[144]
Ethanol	705.7	$\$/\text{ton}_{\text{EtOH}}$	[197]
C2-C4	1067.2	$\$/\text{ton}_{\text{C3}}$	[197]
Electricity	41.34	$\$/\text{GJ}_{\text{NGCC}}$	EU price in the first half of 2021 [194]
Fuel gas	9.76	$\$/\text{GJ}_{\text{fuel-gas}}$	EU price in the first half of 2021 [194]
Steam	15.35	$\$/\text{GJ}_{\text{steam}}$	*
Cooling water	0.029	$\$/\text{ton}_{\text{water}}$	[195]
Cooling utility	0.43	$\$/\text{GJ}_{\text{cooling}}$	**
CO_2 storage	5.56	$\$/\text{ton}_{\text{CO}_2\text{s}}$	***

* calculated in TLV [198].

**The cost of cooling is calculated by cooling water emission factor times its required amount (based on $\Delta T = 20℃$, heat transfer efficiency $\eta = 0.8$).

***δ_{CO_2}: cost of the CO_2 transportation and storage in underground is obtained in IECM (assuming a 50 km pipeline is used for transportation) [177].

B.3 Single-objective (LCA) optimization of the industrial park

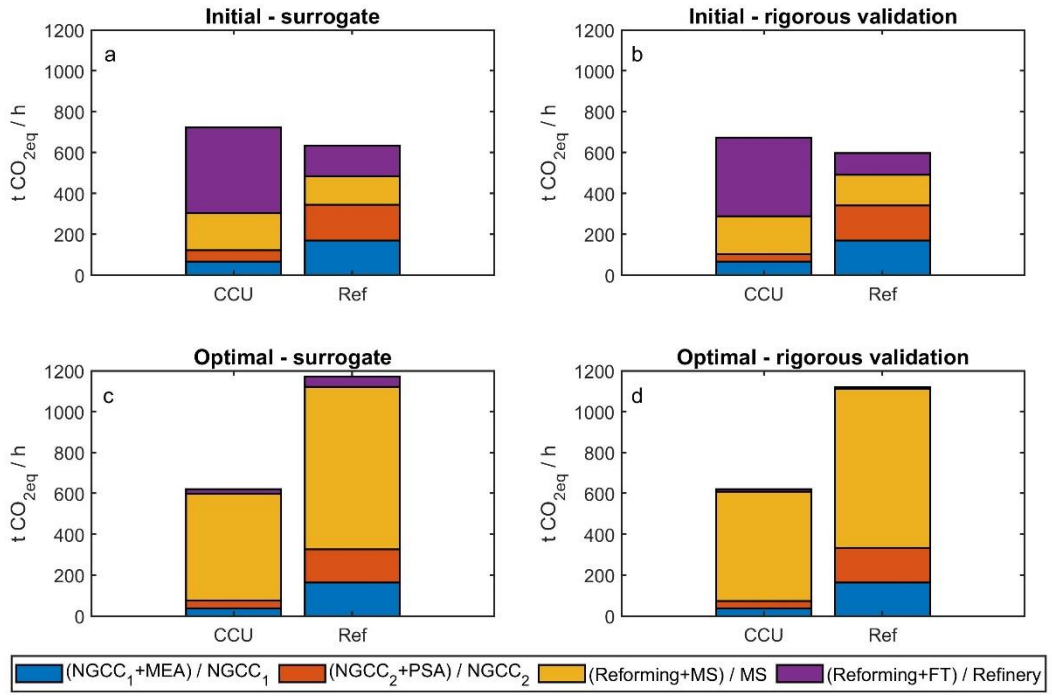


Figure S10. Validation of surrogate models by rigorous simulation for the industrial park regarding: (a) initial operating condition based on surrogates; (b) initial operating condition based on rigorous process models; (c) optimal operating condition based on surrogates; (d) optimal operating condition based on rigorous process models. For the legends: left of '/' for CCU system, right of '/' for the reference system.

Table S17. Scenario analysis for optimal values of decision variables, regarding maximum 0%, 25%, 50% and 100% heating utility is substituted by low-carbon electricity.

	Design variables	Unit	Max heating % substituted by CCS-elec. 0	0.25	0.5	1
MEA	r_{CO_2}	-	0.925	0.917	0.935	0.933
1st PSA	P_{L1}	bar	0.006	0.006	0.008	0.007
	P_{I1}	bar	0.203	0.447	0.375	0.406
	v_{feed1}	m/s	0.857	1.265	0.614	0.614
	t_{ads1}	s	69.800	51.744	80.648	68.789
	t_{bd1}	s	42.274	50.743	54.996	32.515
	t_{evac1}	s	190.152	194.448	179.226	183.637
2nd PSA	P_{L2}	bar	0.019	0.011	0.011	0.014
	P_{I2}	bar	0.267	0.432	0.266	0.170
	v_{feed2}	m/s	0.316	0.917	0.917	0.534
	t_{ads2}	s	60.379	59.411	46.559	59.411
	t_{bd2}	s	82.815	35.890	36.050	44.544
	t_{evac2}	s	176.463	191.707	178.820	178.820
CO_2 to FT	z_{FT}	-	0.027	0.046	0.027	0.027
FT	T_{FT}	℃	257.929	254.435	263.910	247.886
	P_{FT}	bar	42.624	46.535	17.334	25.904
	$tray_{FT}$	-	52	52	52	62
	T_{ref1}	℃	858.981	774.820	876.081	876.081
	P_{ref1}	bar	3.129	4.730	5.130	5.073
	S_{purge}	-	0.102	0.194	0.069	0.045
	Re_{FT}	-	0.468	0.660	0.828	0.573
MEOH	F_{NG}/F_{CO}	-	3.656	3.656	3.643	3.498
	T_{MS}	℃	189.011	187.149	198.258	204.318
	P_{MS}	bar	77.341	75.294	78.283	69.542
	$Tray_{MS}$	-	64	60	61	46
	T_{ref2}	℃	876.951	913.001	864.478	933.319
	P_{ref2}	bar	5.176	5.474	4.615	6.224
Heating utility	$Frac_{fuel_{ele-CCS}}$	-	0.000	0.250	0.498	0.997
	$Frac_{steam_{ele-CCS}}$	-	0.000	0.223	0.496	0.956

B.4 Multi-objective (LCA-Economic) optimization of the industrial park

B.4.1 Optimal values for decisions at carbon price = 0

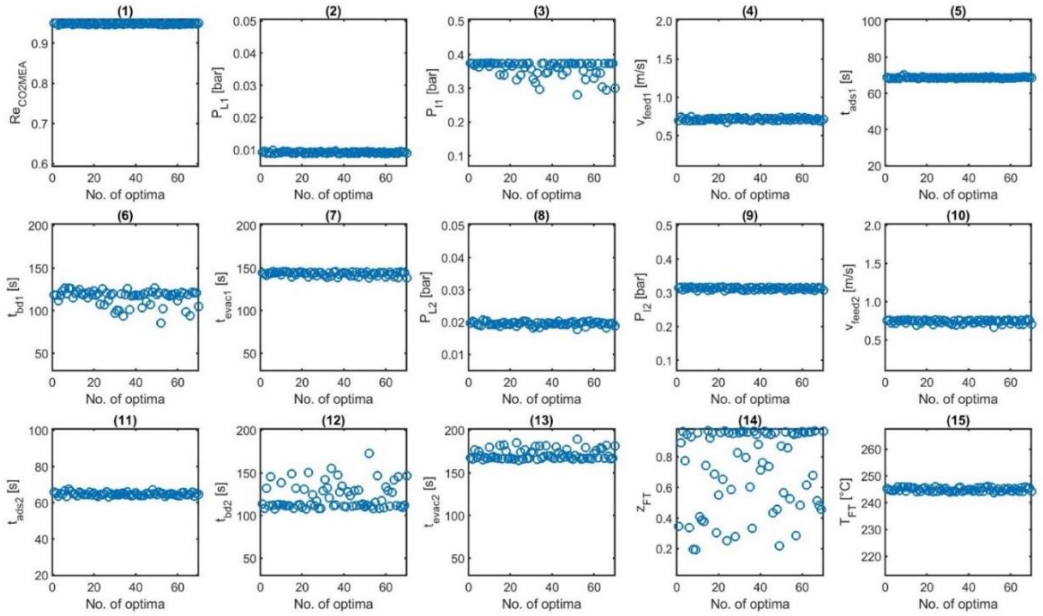


Figure S11. Optimal values of decision variables $x_1 \sim x_{15}$ after multi-objective optimization at $C_{tax} = 0$ \$/ton-CO_2 (corresponding to the Pareto front in Figure 6.10).

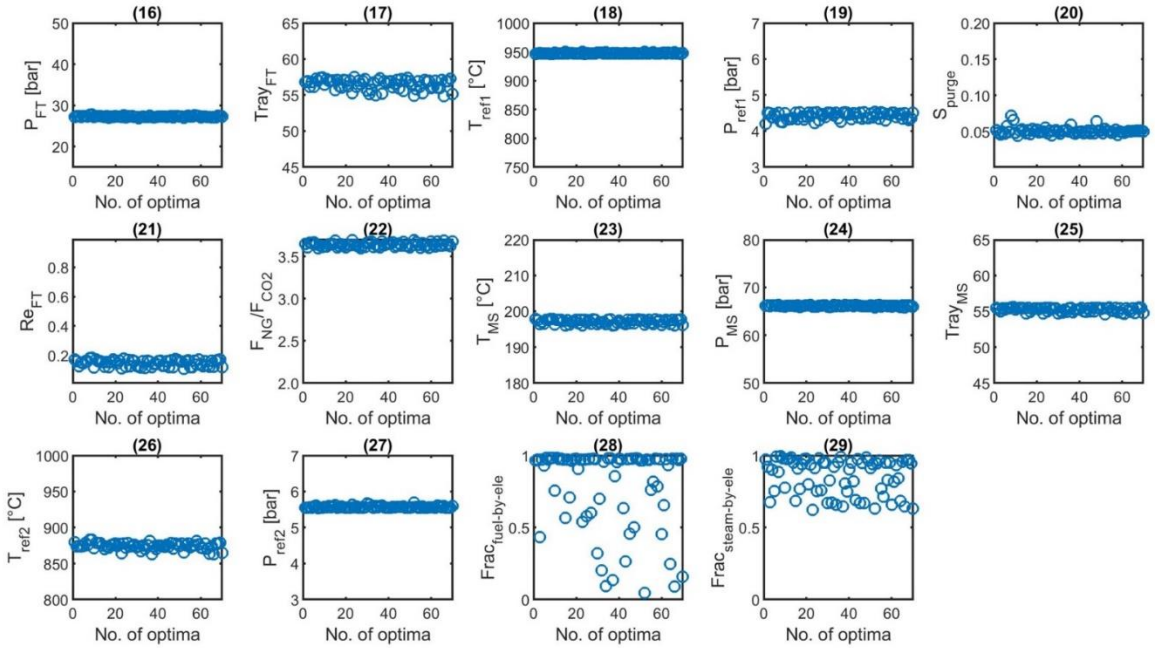


Figure S12. Optimal values of decision variables x_{16}~x_{29} after multi-objective optimization at $C_{tax} = 0$ \$/ton-CO_2 (corresponding to the Pareto front in Figure 6.10).

B.4.2 Evaluation of utilities

Under multi-objective optimization, I checked the range of utilities on both economic aspects and GHG emissions. With the optimal values for the decision variables, the prices and emissions of utilities can be estimated based on the equations in Appendix B.2.2 - B.2.3 (Figure S13). On the one hand, switching to the low-carbon electricity increases the energy cost, by 25% for electricity and by average 337% for fuel-gas heating as well as 201% for steam heating. On the other hand, switching to the low-carbon electricity dramatically reduces GHG emissions, by 76% for electricity and by 52% for fuel-gas heating as well as 57% for steam heating.

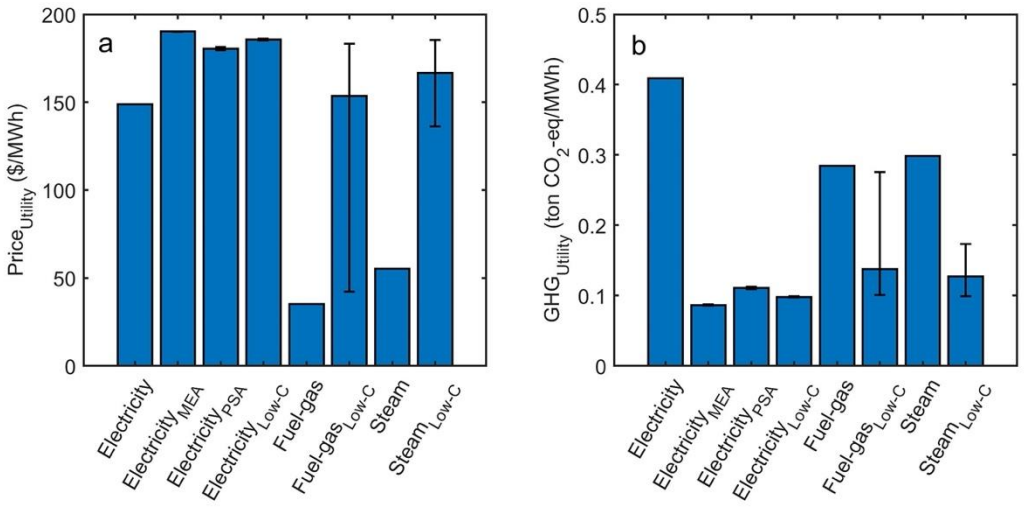


Figure S13. Range of optimized utilities under multi-objective optimization. (a) Price of utilities. (b) GHG emissions of utilities. Clarification for the x-axis label - (1) $Electricity$: NG-based power plant with direct emissions; (2) $Electricity_{MEA}$: NG-based power plant coupled with MEA; (3) $Electricity_{PSA}$: NG-based power plant coupled with PSA; (4) $Electricity_{Low-C}$: low-carbon electricity, which is approximated by the average value between $Electricity_{MEA}$ and $Electricity_{PSA}$ (the cost of CO_2 storage is included); (5) $Fuel\text{-}gas$: heating provided by fuel gas; (6) $Fuel\text{-}gas_{Low-C}$: fuel-gas heating partially substituted by low-carbon electricity; (7) $Steam$: heating provided by steam; (8) $Steam_{Low-C}$: steam heating partially substituted by low-carbon electricity.

B.5 Different carbon prices - perform multi-objective optimization

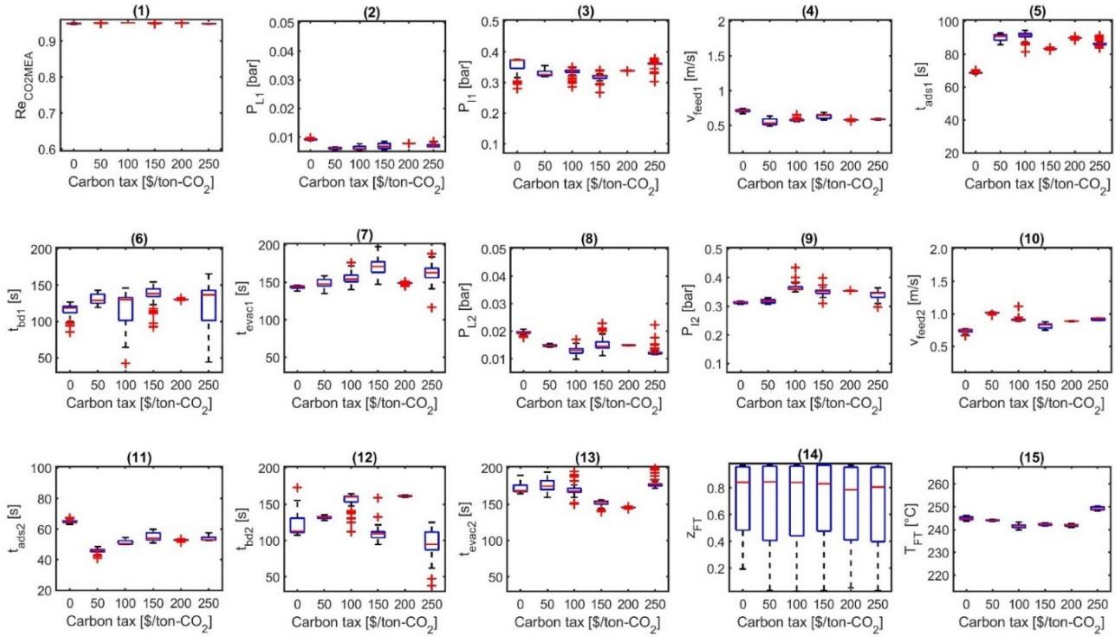


Figure S14. Optimal values of decision variables $x_1 \sim x_{15}$ after multi-objective optimization at $C_{tax} = [0,50,100,150,200,250]$\$/ton-CO_2.

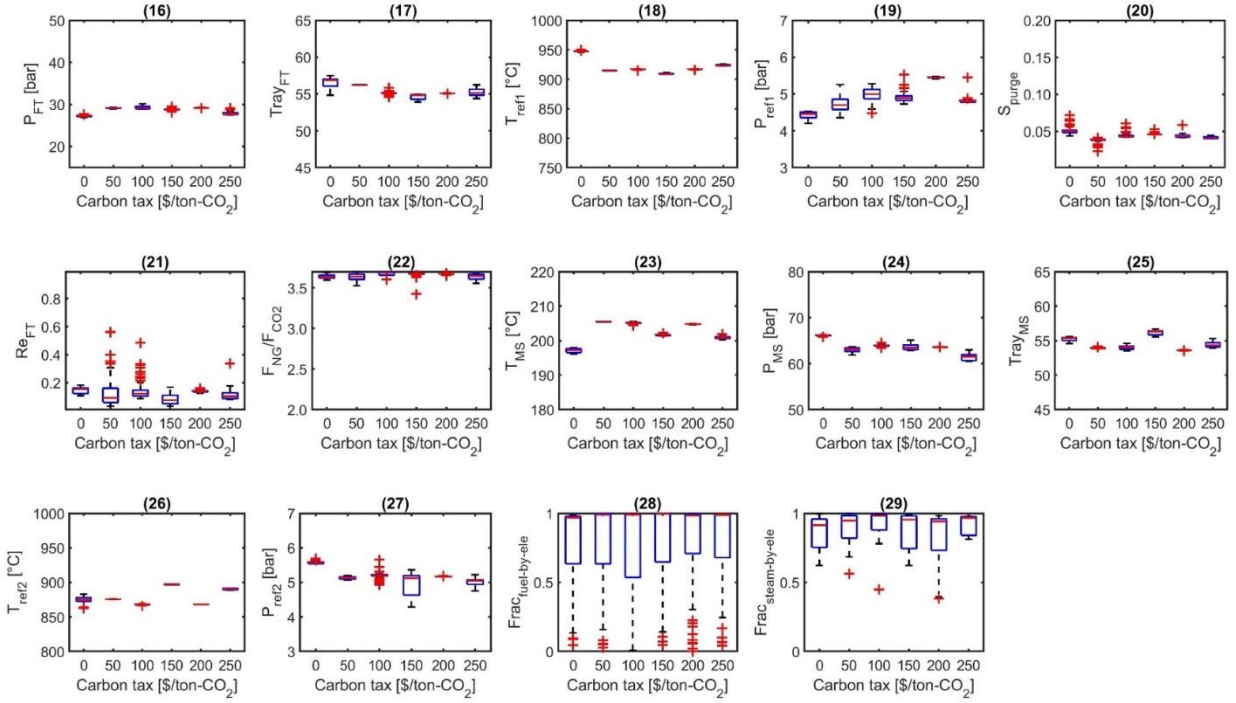


Figure S15. Optimal values of decision variables x_{16}~x_{29} after multi-objective optimization at $C_{tax} = [0,50,100,150,200,250]$\$/ton-CO_2.

Appendix C Algorithms-related information

C.1 Flowchart of TSEMO

TSEMO is an in-house algorithm to solve multi-objective optimization problems [54]. This algorithm aims to identify Pareto front between multiple objectives of expensive-to-evaluate models. First, a small dataset of simulations is collected using a space-filling design method (*e.g.*, Latin Hypercube Sampling). Subsequently, GP surrogate models are trained for the objectives. Then, TSEMO takes random samples by GPs and employs NSGA-II for optimization based on GP. Among the final population of the genetic algorithm, TSEMO selects the new sampling inputs based on the expected hypervolume improvement. In each iteration, given the new sampling inputs, the robust simulations (or experiments) are performed to obtain the outputs and Pareto front. The algorithm terminates when the allocated computational budget is reached. The framework of TSEMO can be referred to Figure S16.

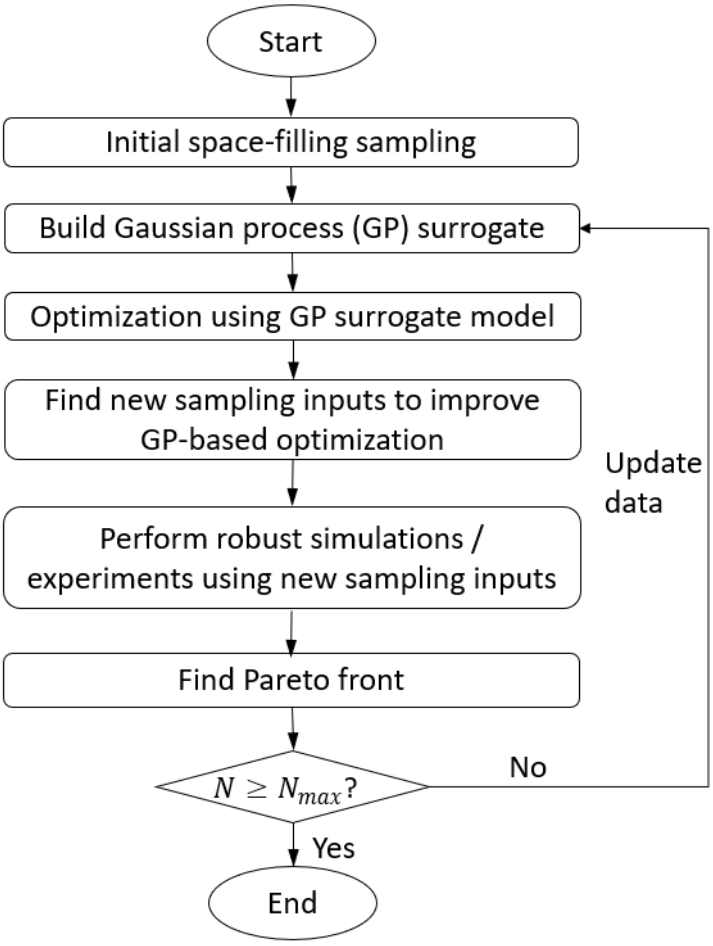


Figure S16. Framework of TSEMO. Adapted from [54].

C.2 Framework of DyOS

DyOS is a framework for adaptive direct sequential multi-stage dynamic optimization [138]. DyOS integrates different non/linear equation solvers, integration, optimization NLP solvers, and is designed for large-scale multi-stage dynamic optimization problems. Based on direct adaptive shooting algorithms, DyOS is tailored to DAEs, and it can integrate multi-stage process models continuously. Initial guesses are given to the decision variables. Several integrators are available for the DAEs to integrate time-dependent variables and gradient over the time horizon of all stages. Following this, function values and gradient values are passed to

NLP solver for optimization. DyOS can be set up in either MATLAB or Python. The framework of DyOS is shown in Figure S17.

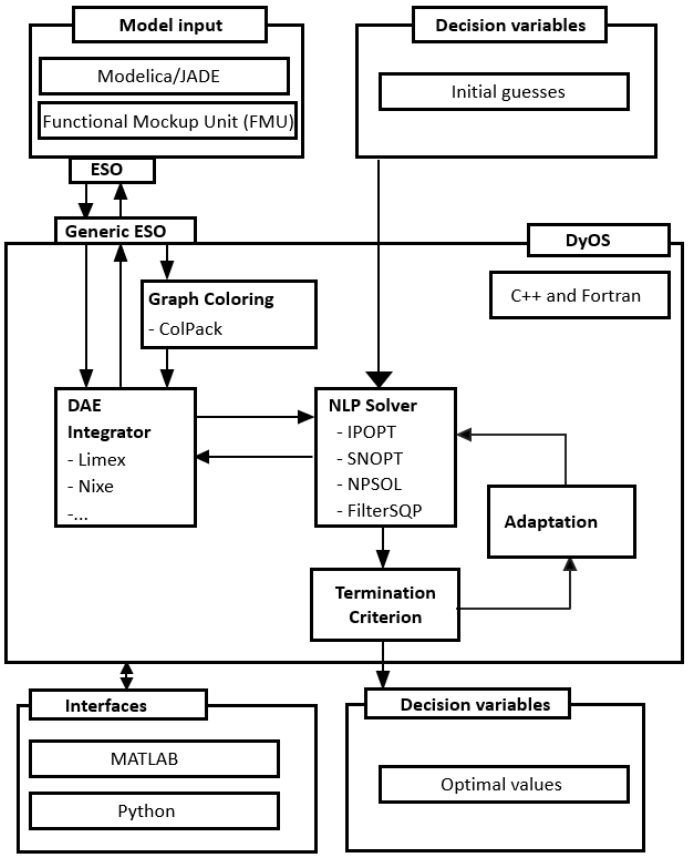


Figure S17. Framework of DyOS. Adapted from [138].

C.3 Reproducibility of the slowdown sampling for peaks function

Peaks function is used as a numerical example to demonstrate the reproducibility of the slowdown sampling. Peaks function consists of two variables (Figure S18), which can be accessed by the function ‘peaks’ in MATLAB. The mathematical form is as follows,

$$Z = 3(1-x)^2 \exp(-x^2) - (y+1)^2 - 10\left(\frac{x}{5} - x^3 - y^5\right)\exp(-x^2 - y^2) - \frac{1}{3}\exp\left(-(x+1)^2 - y^2\right) \quad \text{S31}$$

The slowdown sampling is applied to generate data sequentially for the surrogate construction (Figure S19). The termination criterion is set as $|\text{relative slople}| < 0.02$, when the slope is regarded as 0. With four trails, the total number of sampled data falls between 190 and 220.

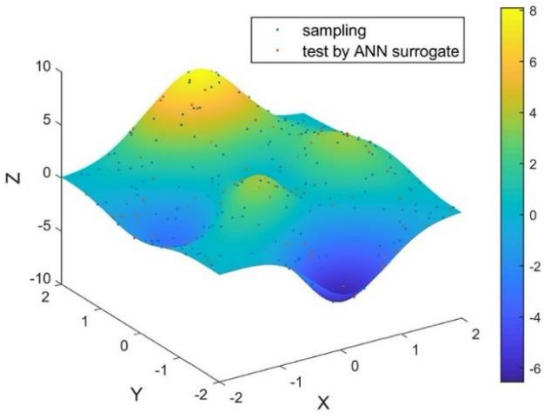


Figure S18. Sampling for the surrogate building of peaks function.

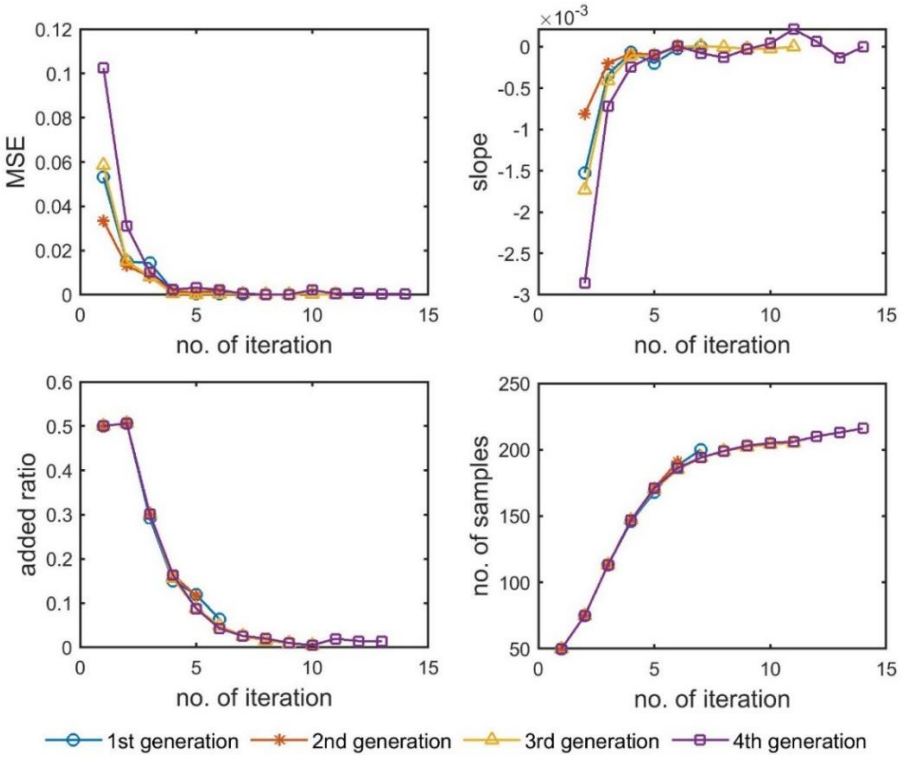


Figure S19. Reproducibility of slow-down sampling for peaks function.

References

[1] International Energy Agency, Net Zero by 2050: A roadmap for the global energy sector, 2021.
[2] United Nations, The global coalition for net-zero emissions is growing, https://www.un.org/en/climatechange/net-zero-coalition, 2021 (accessed: 2021-12-10).
[3] J. Nocedal, S. Wright, Numerical optimization, Springer Science and Business Media, 2006.
[4] W.R. Esposito, C.A. Floudas, Deterministic global optimization in nonlinear optimal control problems, J. Global Optim. 17 (2000) 97-126.
[5] S. Lee, I.E. Grossmann, A global optimization algorithm for nonconvex generalized disjunctive programming and applications to process systems, Comput. Chem. Eng. 25 (2001) 1675-1697.
[6] H.S. Ryoo, N.V. Sahinidis, Global optimization of nonconvex NLPs and MINLPs with applications in process design, Comput. Chem. Eng. 19 (1995) 551-566.
[7] I. Boussaïd, J. Lepagnot, P. Siarry, A survey on optimization metaheuristics, Inf. Sci. 237 (2013) 82-117.
[8] L.T. Biegler, New directions for nonlinear process optimization, Curr. Opin. Chem. Eng. 21 (2018) 32-40.
[9] L.T. Biegler, Chapter 5 - Newton methods for equality constrained optimization, in: Nonlinear programming: concepts, algorithms, and applications to chemical processes, Society for Industrial and Applied Mathmatics, 2010, pp. 91-132.
[10] M. Bui, C.S. Adjiman, A. Bardow, E.J. Anthony, A. Boston, S. Brown, P.S. Fennell, S. Fuss, A. Galindo, L.A. Hackett, J.P. Hallett, H.J. Herzog, G. Jackson, J. Kemper, S. Krevor, G.C. Maitland, M. Matuszewski, I.S. Metcalfe, C. Petit, G. Puxty, J. Reimer, D.M. Reiner, E.S. Rubin, S.A. Scott, N. Shah, B. Smit, J.P.M. Trusler, P. Webley, J. Wilcox, N. Mac Dowell, Carbon capture and storage (CCS): the way forward, Energy Environ. Sci. 11 (2018) 1062-1176.
[11] B. Page, G. Turan, A. Zapantis, J. Burrows, C. Consoli, J. Erikson, I. Havercroft, D. Kearns, H. Liu, D. Rassool, The global status of CCS 2020: vital to achieve net zero, https://www.globalccsinstitute.com/wp-content/uploads/2021/03/Global-Status-of-CCS-Report-English.pdf, 2021 (accessed: 2021-11-01.).
[12] B. Rego de Vasconcelos, J.-M. Lavoie, Recent advances in power-to-X technology for the production of fuels and chemicals, Front. Chem. 7 (2019) 392.
[13] Siemens-Energy, Power-to-X: The crucial business on the way to a carbon-free world, https://www.siemens-energy.com/global/en/offerings/technical-papers/download-power-to-x.html, 2021 (accessed: 2021-11-10.).
[14] V. Dieterich, A. Buttler, A. Hanel, H. Spliethoff, S. Fendt, Power-to-liquid via synthesis of methanol, DME or Fischer–Tropsch-fuels: a review, Energy Environ. Sci. 13 (2020) 3207-3252.

[15] L.T. Biegler, I.E. Grossmann, Retrospective on optimization, Comput. Chem. Eng. 28 (2004) 1169-1192.
[16] S.H. Kim, F. Boukouvala, Machine learning-based surrogate modeling for data-driven optimization: a comparison of subset selection for regression techniques, Optim. Lett. 14 (2020) 989-1010.
[17] K. Hornik, M. Stinchcombe, H. White, Multilayer feedforward networks are universal approximators, Neural Netw. 2 (1989) 359-366.
[18] S.G. Subraveti, Z.K. Li, V. Prasad, A. Rajendran, Machine learning-based multiobjective optimization of pressure swing adsorption, Ind. Eng. Chem. Res. 58 (2019) 20412-20422.
[19] F. Boukouvala, M.M.F. Hasan, C.A. Floudas, Global optimization of general constrained grey-box models: new method and its application to constrained PDEs for pressure swing adsorption, J. Global Optim. 67 (2017) 3-42.
[20] A. Gonzalez-Garay, G. Guillen-Gosalbez, SUSCAPE: A framework for the optimal design of SUStainable ChemicAl ProcEsses incorporating data envelopment analysis, Chem. Eng. Res. Des. 137 (2018) 246-264.
[21] A.M. Schweidtmann, E. Esche, A. Fischer, M. Kloft, J.-U. Repke, S. Sager, A. Mitsos, Machine learning in chemical engineering: a perspective, Chem-Ing-Tech 93 (2021) 2029-2039.
[22] H.A. Balogun, D. Bahamon, S. AlMenhali, L.F. Vega, A. Alhajaj, Are we missing something when evaluating adsorbents for CO_2 capture at the system level?, Energy Environ. Sci. (2021) 6360-6380.
[23] R. Haghpanah, A. Majumder, R. Nilam, A. Rajendran, S. Farooq, I.A. Karimi, M. Amanullah, Multiobjective optimization of a four-step adsorption process for postcombustion CO_2 capture via finite volume simulation, Ind. Eng. Chem. Res. 52 (2013) 4249-4265.
[24] Z. Hao, A. Caspari, A.M. Schweidtmann, Y. Vaupel, A.A. Lapkin, A. Mhamdi, Efficient hybrid multiobjective optimization of pressure swing adsorption, Chem. Eng. J. 423 (2021) 130248.
[25] M.M.F. Hasan, R.C. Baliban, J.A. Elia, C.A. Floudas, Modeling, simulation, and optimization of postcombustion CO_2 capture for variable feed concentration and flow rate. 1. chemical absorption and membrane processes, Ind. Eng. Chem. Res. 51 (2012) 15642-15664.
[26] W. Chung, J.H. Lee, Input-output surrogate models for efficient economic evaluation of amine scrubbing CO_2 capture processes, Ind. Eng. Chem. Res. 59 (2020) 18951-18964.
[27] A. González-Garay, M.S. Frei, A. Al-Qahtani, C. Mondelli, G. Guillén-Gosálbez, J. Pérez-Ramírez, Plant-to-planet analysis of CO_2-based methanol processes, Energy Environ. Sci. 12 (2019) 3425-3436.
[28] C. Chen, A. Yang, Power-to-methanol: The role of process flexibility in the integration of variable renewable energy into chemical production, Energy Convers. Manag. 228 (2021) 113673.
[29] Y.H. Kim, K.-W. Jun, H. Joo, C. Han, I.K. Song, A simulation study on gas-to-liquid (natural gas to Fischer–Tropsch synthetic fuel) process optimization, Chem. Eng. J. 155 (2009) 427-432.
[30] C. Zhang, K.-W. Jun, K.-S. Ha, Y.-J. Lee, S.C. Kang, Efficient utilization of greenhouse gases in a gas-to-liquids process combined with CO_2/steam-mixed reforming and Fe-Based Fischer–Tropsch synthesis, Environ. Sci. Technol. 48 (2014) 8251-8257.
[31] J. Burre, D. Bongartz, S. Deutz, C. Mebrahtu, O. Osterthun, R. Sun, S. Völker, A. Bardow, J. Klankermayer, R. Palkovits, A. Mitsos, Comparing pathways for electricity-based

production of dimethoxymethane as a sustainable fuel, Energy Environ. Sci. 14 (2021) 3686-3699.
[32] K. Roh, A.S. Al-Hunaidy, H. Imran, J.H. Lee, Optimization-based identification of CO_2 capture and utilization processing paths for life cycle greenhouse gas reduction and economic benefits, AIChE J. 65 (2019) e16580.
[33] K. Roh, R. Frauzem, T.B.H. Nguyen, R. Gani, J.H. Lee, A methodology for the sustainable design and implementation strategy of CO_2 utilization processes, Comput. Chem. Eng. 91 (2016) 407-421.
[34] R.M. Smith, B. Linnhoff, The design of separators in the context of overall processes, Chem. Eng. Res. Des. 66 (1988) 195-228.
[35] H. Lasi, P. Fettke, H.-G. Kemper, T. Feld, M. Hoffmann, Industry 4.0, Bus. Inf. Syst. Eng. 6 (2014) 239-242.
[36] R.S. Kenett, J. Bortman, The digital twin in Industry 4.0: A wide-angle perspective, Qual. Reliab. Eng. Int. (2021).
[37] L. Cao, D. Russo, K. Felton, D. Salley, A. Sharma, G. Keenan, W. Mauer, H. Gao, L. Cronin, A.A. Lapkin, Optimization of formulations using robotic experiments driven by machine learning DoE, Cell Rep. Phys. Sci. 2 (2021) 100295.
[38] K.K. Saripella, N.C. Loka, R. Mallipeddi, A.M. Rane, S.H. Neau, A quality by experimental design approach to assess the effect of formulation and process variables on the extrusion and spheronization of drug-loaded pellets containing polyplasdone® XL-10, AAPS PharmSciTech 17 (2016) 368-379.
[39] C. Zhang, Y. Amar, L. Cao, A.A. Lapkin, Solvent selection for mitsunobu reaction driven by an active learning surrogate model, Org. Process Res. Dev. 24 (2020) 2864-2873.
[40] C.A. Henao, C.T. Maravelias, Surrogate-based superstructure optimization framework, AIChE J. 57 (2011) 1216-1232.
[41] A. Bhosekar, M. Ierapetritou, Advances in surrogate based modeling, feasibility analysis, and optimization: A review, Comput. Chem. Eng. 108 (2018) 250-267.
[42] K. McBride, K. Sundmacher, Overview of surrogate modeling in chemical process engineering, Chem-Ing-Tech 91 (2019) 228-239.
[43] E.A. del Rio-Chanona, F. Fiorelli, D. Zhang, N.R. Ahmed, K. Jing, N. Shah, An efficient model construction strategy to simulate microalgal lutein photo-production dynamic process, Biotechnol. Bioeng. 114 (2017) 2518-2527.
[44] F. Boukouvala, M.G. Ierapetritou, Surrogate-based optimization of expensive flowsheet modeling for continuous pharmaceutical manufacturing, Journal Pharm. Innov. 8 (2013) 131-145.
[45] X. Wan, J.F. Pekny, G.V. Reklaitis, Simulation based optimization of supply chains with a surrogate model, in: A. Barbosa-Póvoa, H. Matos (Eds.) Computer Aided Chemical Engineering, Elsevier 2004, pp. 1009-1014.
[46] W. Ye, F. You, A computationally efficient simulation-based optimization method with region-wise surrogate modeling for stochastic inventory management of supply chains with general network structures, Comput. Chem. Eng. 87 (2016) 164-179.
[47] R. Yondo, E. Andrés, E. Valero, A review on design of experiments and surrogate models in aircraft real-time and many-query aerodynamic analyses, Prog. Aerosp. Sci. 96 (2018) 23-61.
[48] T.W. Simpson, T.M. Mauery, J.J. Korte, F. Mistree, Kriging models for global approximation in simulation-based multidisciplinary design optimization, AIAA J. 39 (2001) 2233-2241.

[49] N.V. Queipo, R.T. Haftka, W. Shyy, T. Goel, R. Vaidyanathan, P. Kevin Tucker, Surrogate-based analysis and optimization, Prog. Aerosp. Sci. 41 (2005) 1-28.
[50] S. Recker, M. Skiborowski, C. Redepenning, W. Marquardt, A unifying framework for optimization-based design of integrated reaction–separation processes, Comput. Chem. Eng. 81 (2015) 260-271.
[51] A.I.J. Forrester, A.J. Keane, Recent advances in surrogate-based optimization, Prog. Aerosp. Sci. 45 (2009) 50-79.
[52] I. Fahmi, S. Cremaschi, Process synthesis of biodiesel production plant using artificial neural networks as the surrogate models, Comput. Chem. Eng. 46 (2012) 105-123.
[53] D. Duvenaud, Automatic model construction with Gaussian processes, University of Cambridge, 2014.
[54] E. Bradford, A.M. Schweidtmann, A. Lapkin, Efficient multiobjective optimization employing Gaussian processes, spectral sampling and a genetic algorithm, J. Global Optim. 71 (2018) 407-438.
[55] A.M. Schweidtmann, A. Mitsos, Deterministic global optimization with artificial neural networks embedded, Journal of Optimization Theory and Applications 180 (2019) 925-948.
[56] A.R. Omondi, J.C. Rajapakse, FPGA implementations of neural networks, Springer2006.
[57] J. Bergstra, Y. Bengio, Random search for hyper-parameter optimization, J. Mach. Learn. Res. 13 (2012) 281–305.
[58] P. Boyle, M. Frean, Multiple output Gaussian process regression, (2005).
[59] A.G. Wilson, D.A. Knowles, Z. Ghahramani, Gaussian process regression networks, arXiv preprint:1110.4411 (2011).
[60] D. Bongartz, Deterministic global flowsheet optimization for the design of energy conversion processes, RWTH Aachen University, 2020.
[61] S. Cheng, B. Liu, T.O. Ting, Q. Qin, Y. Shi, K. Huang, Survey on data science with population-based algorithms, Big Data Anal. 1 (2016) 3.
[62] V.S. Vassiliadis, W. Kähm, E.A. del Rio Chanona, Y. Yuan, Optimization for chemical and biochemical engineering: Theory, algorithms, modeling and applications, Cambridge University Press, Cambridge, 2021.
[63] L.T. Biegler, Nonlinear programming: concepts, algorithms, and applications to chemical processes, SIAM, 2010.
[64] D. Fouskakis, D. Draper, Stochastic optimization: a review, Int. Stat. Rev. 70 (2002) 315-349.
[65] A.E. Eiben, J.E. Smith, Introduction to evolutionary computing, Springer 2003.
[66] C. Junying, Q. Zheng, L. Yu, L. Jiang, Particle swarm optimization with local search, 2005 International Conference on Neural Networks and Brain, 2005, pp. 481-484.
[67] G. Folino, C. Pizzuti, G. Spezzano, Parallel hybrid method for SAT that couples genetic algorithms and local search, IEEE Trans. Evol. Comput. 5 (2001) 323-334.
[68] C.K. Chak, G. Feng, Accelerated genetic algorithms: combined with local search techniques for fast and accurate global search, Proceedings of the 1995 IEEE International Conference on Evolutionary Computation. Part 1 (of 2), 1995, pp. 378-383.
[69] H. Ritchie, M. Roser, R. Pablo, CO_2 and greenhouse gas emissions, 2020. https://ourworldindata.org/emissions-by-sector. (Accessed 2021-11-10).
[70] J. Deutch, Is net zero carbon 2050 possible?, Joule 4 (2020) 2237-2240.
[71] BP, BP Energy Outlook: 2020 edition, https://www.bp.com/en/global/corporate/energy-economics/energy-outlook.html, 2020 (accessed: 2021-11-10).
[72] D. Brower, Chevron to spend $10bn on clean energy push, Financial Times,

https://www.ft.com/content/5e284144-3abc-4bee-8e8c-2cc2bac039cd, 2021 (accessed: 2021-09-15).
[73] T. Bruhn, H. Naims, B. Olfe-Kräutlein, Separating the debate on CO_2 utilization from carbon capture and storage, Environ. Sci. Policy 60 (2016) 38-43.
[74] S. Deutz, A. Bardow, Life-cycle assessment of an industrial direct air capture process based on temperature–vacuum swing adsorption, Nature Energy 6 (2021) 203-213.
[75] B.S. Koelbl, M.A. van den Broek, A.P.C. Faaij, D.P. van Vuuren, Uncertainty in carbon capture and storage (CCS) deployment projections: a cross-model comparison exercise, Clim. Change 123 (2014) 461-476.
[76] A. Zimmermann, L. Müller, Y. Wang, T. Langhorst, J. Wunderlich, A. Marxen, K. Armstrong, G. Buchner, A. Kätelhön, M. Bachmann, Techno-economic assessment & life cycle assessment guidelines for CO_2 utilization (Version 1.1), 2020.
[77] M.H. Sahraei, L.A. Ricardez-Sandoval, An integration framework for CO_2 capture processes, Process Systems and Materials for CO_2 Capture, 2017, pp. 523-543.
[78] C. Alie, L. Backham, E. Croiset, P.L. Douglas, Simulation of CO_2 capture using MEA scrubbing: a flowsheet decomposition method, Energy Convers. Manag. 46 (2005) 475-487.
[79] U.S. P.R.Arachchige, M.C. Melaaen, Aspen Plus simulation of CO_2 removal from coal and gas fired power plants, Energy Procedia 23 (2012) 391-399.
[80] S. Sircar, Pressure swing adsorption, Ind. Eng. Chem. Res. 41 (2002) 1389-1392.
[81] R.V. Jasra, N.V. Choudary, S.G.T. Bhat, Separation of gases by pressure swing adsorption, Sep. Sci. Technol. 26 (1991) 885-930.
[82] R. Haghpanah, R. Nilam, A. Rajendran, S. Farooq, I.A. Karimi, Cycle synthesis and optimization of a VSA process for postcombustion CO_2 capture, AIChE J. 59 (2013) 4735-4748.
[83] J.W. Carter, M.L. Wyszynski, The pressure swing adsorption drying of compressed air, Chem. Eng. Sci. 38 (1983) 1093-1099.
[84] D.M. Ruthven, S. Farooq, Air separation by pressure swing adsorption, Gas Sep. Purif. 4 (1990) 141-148.
[85] J.-G. Jee, J.-S. Lee, C.-H. Lee, Air separation by a small-scale two-bed medical O_2 pressure swing adsorption, Ind. Eng. Chem. Res. 40 (2001) 3647-3658.
[86] A.H. Malek, S. Farooq, Hydrogen purification from refinery fuel gas by pressure swing adsorption, AIChE J. 44 (1998) 1985-1992.
[87] S. Sircar, T.C. Golden, Purification of hydrogen by pressure swing adsorption, Sep. Sci. Technol. 35 (2000) 667-687.
[88] A.L. Chaffee, G.P. Knowles, Z. Liang, J. Zhang, P. Xiao, P.A. Webley, CO_2 capture by adsorption: materials and process development, Int. J. Greenh. Gas Con. 1 (2007) 11-18.
[89] L.Y. Liu, H. Gong, Z. Wang, G. Li, T. Du, Application of pressure swing adsorption technology to capture CO_2 in highly humid flue gas, Prog. Chem. 30 (2018) 872-878.
[90] P.I. Barton, C.C. Pantelides, Modeling of combined discrete/continuous processes, AIChE J. 40 (1994) 966-979.
[91] K.Z. House, C.F. Harvey, M.J. Aziz, D.P. Schrag, The energy penalty of post-combustion CO_2 capture & storage and its implications for retrofitting the U.S. installed base, Energy Environ. Sci. 2 (2009) 193-205.
[92] S. Dang, H. Yang, P. Gao, H. Wang, X. Li, W. Wei, Y. Sun, A review of research progress on heterogeneous catalysts for methanol synthesis from carbon dioxide hydrogenation, Catal. Today 330 (2019) 61-75.

[93] X. An, Y. Zuo, Q. Zhang, J. Wang, Methanol synthesis from CO_2 hydrogenation with a Cu/Zn/Al/Zr fibrous catalyst, Chinese J. Chem. Eng. 17 (2009) 88-94.
[94] R. Guil-López, N. Mota, J. Llorente, E. Millán, B. Pawelec, J.L.G. Fierro, R.M. Navarro, Methanol synthesis from CO_2: A review of the latest developments in heterogeneous catalysis, Materials 12 (2019) 3902.
[95] P. Brinckerhoff, Accelerating the uptake of CCS: industrial use of captured carbon dioxide, Global CCS Institute, https://www.globalccsinstitute.com/archive/hub/publications/14026/accelerating-uptake-ccs-industrial-use-captured-carbon-dioxide.pdf, 2011, (accessed: 2021-11-01).
[96] M. Götz, J. Lefebvre, F. Mörs, A. McDaniel Koch, F. Graf, S. Bajohr, R. Reimert, T. Kolb, Renewable power-to-gas: a technological and economic review, Renew. Energy 85 (2016) 1371-1390.
[97] J.-M. Lavoie, Review on dry reforming of methane, a potentially more environmentally-friendly approach to the increasing natural gas exploitation, Front. Chem. 2 (2014).
[98] J. Baltrusaitis, W.L. Luyben, Methane conversion to syngas for gas-to-liquids (GTL): is sustainable CO_2 reuse via dry methane reforming (DMR) cost competitive with SMR and ATR processes?, ACS Sustain. Chem. Eng. 3 (2015) 2100-2111.
[99] Y. Lim, C.-J. Lee, Y.S. Jeong, I.H. Song, C.J. Lee, C. Han, Optimal design and decision for combined steam reforming process with dry methane reforming to reuse CO_2 as a raw material, Ind. Eng. Chem. Res. 51 (2012) 4982-4989.
[100] K. Wittich, M. Krämer, N. Bottke, S.A. Schunk, Catalytic dry reforming of methane: insights from model systems, ChemCatChem 12 (2020) 2130-2147.
[101] Y. Sun, T. Ritchie, S.S. Hla, S. McEvoy, W. Stein, J.H. Edwards, Thermodynamic analysis of mixed and dry reforming of methane for solar thermal applications, J. Nat. Gas Chem. 20 (2011) 568-576.
[102] M.E. Dry, The Fischer–Tropsch process: 1950–2000, Catal. Today 71 (2002) 227-241.
[103] J.R. van Ommen, J. Grievink, Synthesis gas utilization for transportation fuel production, Biomass as a Sustainable Energy Source for the Future2014, pp. 525-546.
[104] R.M. de Deugd, F. Kapteijn, J.A. Moulijn, Trends in Fischer–Tropsch reactor technology - opportunities for structured reactors, Top. Catal. 26 (2003) 29-39.
[105] P. Azadi, G. Brownbridge, I. Kemp, S. Mosbach, J.S. Dennis, M. Kraft, Microkinetic modeling of the Fischer–Tropsch synthesis over Cobalt catalysts, ChemCatChem 7 (2015) 137-143.
[106] R. Guettel, T. Turek, Comparison of different reactor types for low temperature Fischer–Tropsch synthesis: a simulation study, Chem. Eng. Sci. 64 (2009) 955-964.
[107] D. Vervloet, F. Kapteijn, J. Nijenhuis, J.R. van Ommen, Fischer-Tropsch reaction-diffusion in a Cobalt catalyst particle: aspects of activity and selectivity for a variable chain growth probability, Catal. Sci. Technol. 2 (2012) 1221-1233.
[108] P. Kaiser, R.B. Unde, C. Kern, A. Jess, Production of liquid hydrocarbons with CO_2 as carbon source based on reverse water-gas shift and Fischer-Tropsch synthesis, Chem-Ing-Tech 85 (2013) 489-499.
[109] K.-S. Ha, J.W. Bae, K.-J. Woo, K.-W. Jun, Efficient utilization of greenhouse gas in a gas-to-liquids process combined with carbon dioxide reforming of methane, Environ. Sci. Technol. 44 (2010) 1412-1417.
[110] C.G. Visconti, L. Lietti, E. Tronconi, P. Forzatti, R. Zennaro, E. Finocchio, Fischer–Tropsch synthesis on a Co/Al_2O_3 catalyst with CO_2 containing syngas, Appl. Catal. A Gen 355 (2009) 61-68.

[111] C.-J. Lee, Y. Lim, H.S. Kim, C. Han, Optimal gas-to-liquid product selection from natural gas under uncertain price scenarios, Ind. Eng. Chem. Res. 48 (2009) 794-800.
[112] B. Bao, M.M. El-Halwagi, N.O. Elbashir, Simulation, integration, and economic analysis of gas-to-liquid processes, Fuel Process Technol. 91 (2010) 703-713.
[113] A.A. Kiss, J.J. Pragt, H.J. Vos, G. Bargeman, M.T. de Groot, Novel efficient process for methanol synthesis by CO_2 hydrogenation, Chem. Eng. J. 284 (2016) 260-269.
[114] N. von der Assen, A. Bardow, Life cycle assessment of polyols for polyurethane production using CO_2 as feedstock: insights from an industrial case study, Green Chem. 16 (2014) 3272-3280.
[115] A. Sanna, M. Uibu, G. Caramanna, R. Kuusik, M.M. Maroto-Valer, A review of mineral carbonation technologies to sequester CO_2, Chem. Soc. Rev. 43 (2014) 8049-8080.
[116] AspenTech, Aspen Plus, https://www.aspentech.com/en/products/engineering/aspen-plus.
[117] Dassault Systems, Dymola, https://www.3ds.com/products-services/catia/products/dymola.
[118] Process Systems Enterprise, gPROMS, https://www.psenterprise.com/products/gproms.
[119] Modelica-Consortium, OpenModelica, https://www.openmodelica.org.
[120] AVEVA, PRO/II Simulation, https://www.aveva.com/en/products/pro-ii-simulation.
[121] U. Lee, J. Burre, A. Caspari, J. Kleinekorte, A.M. Schweidtmann, A. Mitsos, Techno-economic optimization of a green-field post-combustion CO_2 capture process using superstructure and rate-based models, Ind. Eng. Chem. Res. 55 (2016) 12014-12026.
[122] P. Mahapatra, J. Ma, B. Ng, D. Bhattacharyya, S.E. Zitney, D.C. Miller, Integrated dynamic modeling and advanced process control of carbon capture systems, Energy Procedia 63 (2014) 1354-1367.
[123] Y. Tian, S.E. Demirel, M.M.F. Hasan, E.N. Pistikopoulos, An overview of process systems engineering approaches for process intensification: state of the art, Chem. Eng. Process. 133 (2018) 160-210.
[124] L. Jiang, L.T. Biegler, V.G. Fox, Simulation and optimization of pressure swing adsorption systems for air separation, AIChE J. 49 (2003) 1140-1157.
[125] Y.Q. Ding, M.D. LeVan, Periodic states of adsorption cycles III. convergence acceleration for direct Determination, Chem. Eng. Sci. 56 (2001) 5217-5230.
[126] O.J. Smith, A.W. Westerberg, Acceleration of cyclic steady state convergence for pressure swing adsorption models, Ind. Eng. Chem. Res. 31 (1992) 1569-1573.
[127] F. Capra, M. Gazzani, L. Joss, M. Mazzotti, E. Martelli, MO-MCS, A derivative-free algorithm for the multiobjective optimization of adsorption processes, Ind. Eng. Chem. Res. 57 (2018) 9977-9993.
[128] K.T. Leperi, D. Yancy-Caballero, R.Q. Snurr, F. You, 110th Anniversary: surrogate models based on artificial neural networks to simulate and optimize pressure swing adsorption cycles for CO_2 capture, Ind. Eng. Chem. Res. 58 (2019) 18241-18252.
[129] C. Tsay, R.C. Pattison, M. Baldea, A pseudo-transient optimization framework for periodic processes: pressure swing adsorption and simulated moving bed chromatography, AIChE J. 64 (2018) 2982-2996.
[130] A. Agarwal, L.T. Biegler, S.E. Zitney, Simulation and optimization of pressure swing adsorption systems using reduced-order modeling, Ind. Eng. Chem. Res. 48 (2009) 2327-2343.
[131] D. Buche, N.N. Schraudolph, P. Koumoutsakos, Accelerating evolutionary algorithms with Gaussian process fitness function models, IEEE Trans. Syst. Man Cybern. Part C-Appl. Rev. 35 (2005) 183-194.

[132] Z.T. Wilson, N.V. Sahinidis, The ALAMO approach to machine learning, Comput. Chem. Eng. 106 (2017) 785-795.
[133] H. Mobahi, J.W. Fisher, Coarse-to-fine minimization of some common nonconvexities, in: E. Bae, T.F. Chan, M. Lysaker (Eds.) Energy Minimization Methods in Computer Vision and Pattern Recognition, Emmcvpr 2015, pp. 71-84.
[134] J. Yao, A. Al-Dahle, Coarse-to-fine optimization for speech enhancement, CoRR abs/1908.08044 (2019).
[135] C. Raphael, Coarse-to-fine dynamic programming, IEEE Trans. Pattern Anal. Mach. Intell. 23 (2001) 1379-1390.
[136] L. Estupiñan Perez, P. Sarkar, A. Rajendran, Experimental validation of multi-objective optimization techniques for design of vacuum swing adsorption processes, Sep. Purif. Technol. 224 (2019) 553-563.
[137] S. Krishnamurthy, V.R. Rao, S. Guntuka, P. Sharratt, R. Haghpanah, A. Rajendran, M. Amanullah, I.A. Karimi, S. Farooq, CO_2 capture from dry flue gas by vacuum swing adsorption: a pilot plant study, AIChE J. 60 (2014) 1830-1842.
[138] A. Caspari, A.M. Bremen, J.M.M. Faust, F. Jung, C.D. Kappatou, S. Sass, Y. Vaupel, K. Hannemann-Tamas, A. Mhamdi, A. Mitsos, DyOS - a framework for optimization of large-scale differential algebraic equation systems, in: A.A. Kiss, E. Zondervan, R. Lakerveld, L. Ozkan (Eds.) 29th European Symposium on Computer Aided Process Engineering, Pt A2019, pp. 619-624.
[139] B. Shahriari, K. Swersky, Z. Wang, R.P. Adams, N.d. Freitas, Taking the human out of the loop: a review of bayesian optimization, Proc. IEEE 104 (2016) 148-175.
[140] P. Fritzson, P. Bunus, Modelica - a general object-oriented language for continuous and discrete-event system modeling and simulation, 35th Annual Simulation Symposium, Proceedings 2002, pp. 365-380.
[141] R. Wu, G. Mavromatidis, K. Orehounig, J. Carmeliet, Multiobjective optimisation of energy systems and building envelope retrofit in a residential community, Appl. Energy 190 (2017) 634-649.
[142] A. Auger, J. Bader, D. Brockhoff, E. Zitzler, Hypervolume-based multiobjective optimization: theoretical foundations and practical implications, Theor. Comput. Sci. 425 (2012) 75-103.
[143] E. Zitzler, D. Brockhoff, L. Thiele, The hypervolume indicator revisited: on the design of Pareto-compliant indicators via weighted integration, in: S. Obayashi, K. Deb, C. Poloni, T. Hiroyasu, T. Murata (Eds.) Evolutionary Multi-Criterion Optimization, Springer Berlin Heidelberg, Berlin, Heidelberg, 2007, pp. 862-876.
[144] P. Brandl, M. Bui, J.P. Hallett, N. Mac Dowell, Beyond 90% capture: possible, but at what cost?, Int. J. Greenh. Gas Con. 105 (2021) 103239.
[145] Methanex, Methanex methanol price sheet, https://www.methanex.com/our-business/pricing, 2021 (accessed: 2021-08-01).
[146] E.C. Carlson, Don't gamble with physical properties for simulations, Chem. Eng. Prog. 92 (1996) 35-46.
[147] C. Cortes, V. Vapnik, Support-vector networks, Mach. Learn. 20 (1995) 273-297.
[148] Y. Cao, M.A. Brubaker, D.J. Fleet, A. Hertzmann, Efficient optimization for sparse Gaussian process regression, Adv. Neural Inf. Process. Syst. 26 (2013).
[149] E. Tsymbalov, S. Makarychev, A. Shapeev, M. Panov, Deeper connections between neural networks and Gaussian processes speed-up active learning, arXiv preprint arXiv:1902.10350 (2019).

[150] M. Lilley, M. Frean, Neural networks: a replacement for gaussian processes?, International Conference on Intelligent Data Engineering and Automated Learning, Springer, 2005, pp. 195-202.
[151] J. Cai, J. Luo, S. Wang, S. Yang, Feature Selection in Machine Learning: A New Perspective, Neurocomputing 300 (2018) 70-79.
[152] S. Khalid, T. Khalil, S. Nasreen, A Survey of feature selection and feature extraction techniques in machine learning, 2014 Science and Information Conference, 2014, pp. 372-378.
[153] Z. Hao, C. Zhang, A.A. Lapkin, Efficient surrogates construction of chemical processes: Case studies on pressure swing adsorption and gas-to-liquids, AIChE J. 68 (2022) e17616.
[154] S.S. Garud, I.A. Karimi, M. Kraft, Design of computer experiments: A review, Comput. Chem. Eng. 106 (2017) 71-95.
[155] D. Ibrahim, M. Jobson, J. Li, G. Guillén-Gosálbez, Surrogate models combined with a support vector machine for the optimized design of a crude oil distillation unit using genetic algorithms, in: A. Espuña, M. Graells, L. Puigjaner (Eds.), Computer Aided Chemical Engineering, Elsevier2017, pp. 481-486.
[156] Y. Kim, E. Kim, E. Antono, B. Meredig, J. Ling, Machine-learned metrics for predicting the likelihood of success in materials discovery, Npj Comput. Mater. 6 (2020) 131.
[157] C. Houben, N. Peremezhney, A. Zubov, J. Kosek, A.A. Lapkin, Closed-loop multitarget optimization for discovery of new emulsion polymerization recipes, Org. Process Res. Dev. 19 (2015) 1049-1053.
[158] A. Cozad, N.V. Sahinidis, D.C. Miller, Learning surrogate models for simulation-based optimization, AIChE J. 60 (2014) 2211-2227.
[159] D. Busby, C.L. Farmer, A. Iske, Hierarchical nonlinear approximation for experimental design and statistical data fitting, SIAM J. Sci. Comput. 29 (2007) 49-69.
[160] G. Li, V. Aute, S. Azarm, An accumulative error based adaptive design of experiments for offline metamodeling, Struct. Multidiscipl. Optim. 40 (2010) 137-155.
[161] A.T. Penteado, E. Esche, J. Weigert, J.-U. Repke, A framework for stochastic and surrogate-assisted optimization using sequential modular process simulators, in: S. Pierucci, F. Manenti, G.L. Bozzano, D. Manca (Eds.), Computer Aided Chemical Engineering, Elsevier2020, pp. 1903-1908.
[162] Z. Hao, M.H. Barecka, A.A. Lapkin, Accelerating net zero from the perspective of optimizing a carbon capture and utilization system, Energy Environ. Sci. 15 (2022) 2139-2153.
[163] M.H. Barecka, J.W. Ager, A.A. Lapkin, Economically viable CO_2 electroreduction embedded within ethylene oxide manufacturing, Energy Environ. Sci. 14 (2021) 1530-1543.
[164] A. Sternberg, A. Bardow, Power-to-What? – Environmental assessment of energy storage systems, Energy Environ. Sci. 8 (2015) 389-400.
[165] VoltaChem, Power-to-Fuels. https://www.voltachem.com/applications/power-to-fuels, (accessed: 2021-11-10).
[166] C. Chen, Electrification & renewable energy integration: a case study of methanol production, University of Oxford, 2021.
[167] N.M. Haegel, R. Margolis, T. Buonassisi, D. Feldman, A. Froitzheim, R. Garabedian, M. Green, S. Glunz, H.-M. Henning, B. Holder, I. Kaizuka, B. Kroposki, K. Matsubara, S. Niki, K. Sakurai, R.A. Schindler, W. Tumas, E.R. Weber, G. Wilson, M. Woodhouse, S. Kurtz, Terawatt-scale photovoltaics: Trajectories and challenges, Science 356 (2017) 141-143.
[168] D.C. Miller, D. Agarwal, D. Bhattacharyya, J. Boverhof, Y.-W. Cheah, Y. Chen, J. Eslick, J. Leek, J. Ma, P. Mahapatra, B. Ng, N.V. Sahinidis, C. Tong, S.E. Zitney, Innovative computational tools and models for the design, optimization and control of carbon capture

processes, in: Z. Kravanja, M. Bogataj (Eds.), Computer Aided Chemical Engineering, Elsevier2016, pp. 2391-2396.
[169] J.F. Green, Does carbon pricing reduce emissions? A review of ex-post analyses, Environ. Res. Lett. 16 (2021) 043004.
[170] J. Lilliestam, A. Patt, G. Bersalli, The effect of carbon pricing on technological change for full energy decarbonization: a review of empirical ex-post evidence, Wiley Interdiscip. Rev. Clim. Change 12 (2021) e681.
[171] H.A. Daggash, N. Mac Dowell, Higher carbon prices on emissions alone will not deliver the Paris Agreement, Joule 3 (2019) 2120-2133.
[172] M. Nicholson, T. Biegler, B.W. Brook, How carbon pricing changes the relative competitiveness of low-carbon baseload generating technologies, Energy 36 (2011) 305-313.
[173] C.A. Grande, Advances in pressure swing adsorption for gas separation, ISRN Chem. Eng. (2012) 982934.
[174] C.C.S. Reddy, Project engineering and management for process retrofitting and revamping, Chemical Process Retrofitting and Revamping 2016, pp. 19-56.
[175] G.H. Graaf, E.J. Stamhuis, A.A.C.M. Beenackers, Kinetics of low-pressure methanol synthesis, Chem. Eng. Sci. 43 (1988) 3185-3195.
[176] K.M.V. Bussche, G.F. Froment, A steady-state kinetic model for methanol synthesis and the water gas shift reaction on a commercial $Cu/ZnO/Al_2O_3$ catalyst, J. Catal. 161 (1996) 1-10.
[177] IECM, Integrated environmental control model (IECM) version 11.4 (Carnegie Mellon University), https://www.cmu.edu/epp/iecm/index.html, (accessed: 2020-08-01).
[178] AspenTech, Aspen Plus Methanol Synthesis Model, 2018. The methanol synthesis model can be downloaded from esupport.aspentech.com. Alternatively, for Aspen Plus V11 and higher versions, the model file can be accessed in C:\Program Files\AspenTech\Aspen Plus Vxx.x\GUI\Examples\Bulk Chemical\Methanol.
[179] S.A. Al-Sobhi, A. Elkamel, F.S. Erenay, M.A. Shaik, Simulation-optimization framework for synthesis and design of natural gas downstream utilization networks, Energies 11 (2018) 362.
[180] M. Khan, The EU's electricity market and why soaring gas prices are driving bills higher, https://www.ft.com/content/f37d2a36-4609-4b3e-9795-064b6d459676,
Financial Times, 2021 (accessed: 2021-10-15).
[181] M. Geissdoerfer, P. Savaget, N.M.P. Bocken, E.J. Hultink, The circular economy – a new sustainability paradigm? J. Clean. Prod. 143 (2017) 757-768.
[182] D. Dodd, Gas price crisis threatens to derail European recovery,
https://www.ft.com/content/223d5dd1-4212-4e47-8c7b-b54b53744ab2, Financial Times, 2021 (accessed: 2021-09-21).
[183] A.S. Bambal, V.S. Guggilla, E.L. Kugler, T.H. Gardner, D.B. Dadyburjor, Poisoning of a silica-supported Cobalt catalyst due to presence of sulfur impurities in syngas during Fischer–Tropsch synthesis: effects of chelating agent, Ind. Eng. Chem. Res. 53 (2014) 5846-5857.
[184] F. Galvanin, M. Barolo, F. Bezzo, S. Macchietto, A backoff strategy for model-based experiment design under parametric uncertainty, AIChE J. 56 (2010) 2088-2102.
[185] E.J. Sheu, E.M.A. Mokheimer, A.F. Ghoniem, A review of solar methane reforming systems, Int. J. Hydrog. Energy 40 (2015) 12929-12955.
[186] H. Von Storch, S. Becker-Hardt, C. Sattler, (Solar) Mixed reforming of methane: potential and limits in utilizing CO_2 as feedstock for syngas production - a thermodynamic analysis, Energies 11 (2018) 2537.

[187] D. Weisser, A guide to life-cycle greenhouse gas (GHG) emissions from electric supply technologies, Energy 32 (2007) 1543-1559.
[188] C. Zhang, K. W. Jun, R. Gao, Y.-J. Lee, S.C. Kang, Efficient utilization of carbon dioxide in gas-to-liquids process: process simulation and techno-economic analysis, Fuel 157 (2015) 285-291.
[189] I.C. Yates, C.N. Satterfield, Intrinsic kinetics of the Fischer-Tropsch synthesis on a Cobalt catalyst, Energy Fuel 5 (1991) 168-173.
[190] H. Jouhara, N. Khordehgah, S. Almahmoud, B. Delpech, A. Chauhan, S.A. Tassou, Waste heat recovery technologies and applications, Therm. Sci. Eng. Prog. 6 (2018) 268-289.
[191] P.F. Fleming, B.J. Church, 13 - Gas, in: D.A. Snow (Ed.) Plant Engineer's Reference Book (Second Edition), Butterworth-Heinemann, Oxford, 2002, pp. 13-11-13-35.
[192] R.M. Cuéllar-Franca, P. García-Gutiérrez, S.F.R. Taylor, C. Hardacre, A. Azapagic, A novel methodology for assessing the environmental sustainability of ionic liquids used for CO_2 capture, Faraday Discuss 192 (2016) 283-301.
[193] I. Muñoz, K. Flury, N. Jungbluth, G. Rigarlsford, L.M. i Canals, H. King, Life cycle assessment of bio-based ethanol produced from different agricultural feedstocks, Int. J. Life Cycle Assess. 19 (2014) 109-119.
[194] Eurostat, Natural gas price statistics, https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Natural_gas_price_statistics#Natural_gas_prices_for_non-household_consumers, 2021 (accessed: 2021-08-01).
[195] A. Boulamanti, J.A. Moya, Production costs of the chemical industry in the EU and other countries: ammonia, methanol and light olefins, Renew. Sust. Energy Rev. 68 (2017) 1205-1212.
[196] European-Commission, Weekly oil bulletin, https://ec.europa.eu/energy/data-analysis/weekly-oil-bulletin_en, 2021 (accessed: 2021-08-01).
[197] ICIS, ICIS Europe chemicals outlooks, https://www.icis.com/explore/resources/europe-chemical-outlook-2019/, 2019 (accessed: 2021-08-01).
[198] TLV, Calculator: steam unit cost, https://www.tlv.com/global/UK/calculator/steam-unit-cost.html (accessed: 2021-08-01).