THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :

Institut National Polytechnique de Toulouse (Toulouse INP)

Discipline ou spécialité :

Génie des Procédés et de l'Environnement

Présentée et soutenue par :

M. VICTOR CANTU MEDRANO

le vendredi 2 juillet 2021

Titre :

Métaheuristiques et matheuristiques pour des problèmes d'optimisation multi-objectifs en génie des procédés: application à la conception d'une chaîne logistique « hydrogène »

Ecole doctorale :

Mécanique, Energétique, Génie civil, Procédés (MEGeP)

Unité de recherche :

Laboratoire de Génie Chimique (LGC)

Directeur(s) de Thèse :

MME CATHERINE AZZARO-PANTEL

M. ANTONIN PONSICH

Rapporteurs :

M. ANTONIO ESPUNA, UNIV POLITECNICA DE CATALUNYA BARCELONA

M. JEAN-BAPTISTE CAILLAU, UNIVERSITE COTE D'AZUR

Membre(s) du jury :

M. BRUNO SARENI, TOULOUSE INP, Président

M. ANTONIN PONSICH, UNIVERSIDAD AUTONOMA METRPOLITANA MEXICO, Membre

M. JOSE MARIA PONCE ORTEGA, U. Michoacana Morelia, Membre

MME ANNABELLE BRISSE, EUROPEAN INSTITUTE FOR ENERGY RESEARCH, Membre

MME CATHERINE AZZARO-PANTEL, TOULOUSE INP, Membre

to my father, the doctor of the family

Acknowledgments

I gratefully thank the Mexican Council of Science and Technology (CONACyT) and the Energy Department of Mexico (SENER) for the scholarship support for pursuing PhD studies during the last three years; without their support, it would likely be impossible to achieve this work.

I express my sincere gratitude to my advisors Catherine Azzaro-Pantel and Antonin Ponsich. I thank you for giving me this great opportunity of doing a PhD thesis under your supervision. This has been the window to acquire the proper scientific understanding from your expertise and to realize the excellence of your human quality. The time we have worked together is counted in years, yet the tracks you have left in my career is counted in a lifetime.

I thank Prof. Antonio Espuña and Prof. Jean-Baptiste Caillau, my thesis reviewers, not only for your time and effort you dedicated for reading the document, but also for your valuable comments and insights as experts in the Mathematical Optimization area. It is an honor for me to have your critical opinion on my research.

I wish to thank also Prof. Bruno Sareni, Prof. José María Ponce Ortega and Scientific Advisor Annabelle Brisse, for your precious examination of my research work. I genuinely thank you for each question you asked and each comment you made; thanks to your expertise you allowed me to broader the perspectives of my work.

In a more general way, I want to express my gratefulness to each professor who devoted a moment of his/her life for sharing with me a bit of their great knowledge. In particular, I bring to mind professors in Mexico which initially contributed to develop my scientific background. Thank you.

Besides, my gratitude goes to my family which support has been tangible regardless of the distance all these years. Thank you very much for your understanding and unconditional love. I miss you.

I thank Liliana, my wife. Thank you for your love, your encouragement and effort each day since we met each other. Our lives have changed since we arrived in France for the first time. This story of us is like a dream come true. Thank you! Also, I thank Elias and Thiago, for making these moments much happier by being with us on this journey.

And lastly but most importantly, I thank God for the privilege of accomplishing this doctoral research. I know I am blessed. Thank you.

Abstract

Complex optimization problems are ubiquitous in Process Systems Engineering (PSE) and are generally solved by deterministic approaches. The treatment of real case studies usually involves mixed-integer variables, nonlinear functions, a large number of constraints, and several conflicting criteria to be optimized simultaneously, thus challenging the classical methods. The main motivation of this research is therefore to explore alternative solution methods for addressing these complex multiobjective optimization problems related to the PSE area, focusing on the recent advances in Evolutionary Computation. If multiobjective evolutionary algorithms (MOEAs) have proven to be robust for the solution of multiobjective problems, their performance yet strongly depends on the constraint-handling techniques for the solution of highly constrained problems. The core of innovation of this research is the adaptation of metaheuristic-based tools to this class of PSE problems. For this purpose, a two-stage strategy was developed.

First, an empirical study was performed in the perspective of comparing different algorithmic configurations and selecting the best to provide a high-quality approximation of the Pareto front. This study, comprising both academic test problems and several PSE applications, demonstrated that a method using the gradient-based mechanism to repair infeasible solutions consistently obtains the best results, in particular for handling equality constraints. Capitalizing on the experience from this preliminary numerical investigation, a novel matheuristic solution strategy was then developed and adapted to the problem of Hydrogen Supply Chain (HSC) design that encompasses the aforementioned numerical difficulties, considering both economic and environmental criteria.

A MOEA based on decomposition combined with the gradient-based repair was first explored as a solution technique. However, due to the important number of mass balances (equality constraints), this approach showed a poor convergence to the optimal Pareto front. Therefore, a novel matheuristic was developed and adapted to this problem, following a bilevel decomposition: the upper level (discrete) addresses the HSC structure design problem (facility sizing and location), whereas the lower level (Linear Programming problem) solves the corresponding operation subproblem (production and transportation). This strategy allows the development of an ad-hoc matheuristic solution technique, through the hybridization of a MOEA (upper level) with a LP solver (lower level) using a scalarizing function to deal with the two objectives considered. The numerical results obtained for the Occitanie region case study highlight that the hybrid approach produces an accurate approximation of the optimal Pareto front, more efficiently than exact solution methods.

Finally, the matheuristic allowed studying the HSC design problem with more realistic assumptions regarding the technologies used for hydrogen synthesis, the learning rates capturing the increasing maturity of these technologies over time and nonlinear relationships for the computation of Capital and Operational Expenditures (CAPEX and OPEX) for the hydrogen production facilities. The resulting novel model, with a non-convex, bi-objective mixed-integer nonlinear programming (MINLP) formulation, can be efficiently solved through minor modifications in the hybrid algorithm proposed earlier, which finds its mere justification in the determination of the time-wise deployment of sustainable hydrogen supply chains.

Résumé

Les approches systémiques du Génie des Procédés font très fréquemment intervenir des problèmes complexes d'optimisation, généralement résolus par des approches déterministes. L'étude de cas réels implique des variables mixtes, des fonctions non linéaires, un grand nombre de contraintes ainsi que plusieurs critères conflictuels à optimiser simultanément, ce qui met à l'épreuve ces méthodes classiques. La motivation principale de cette recherche est donc d'explorer des méthodes alternatives pour résoudre ces problèmes d'optimisation multi-objectif complexes avec une attention particulière sur les avancées récentes des méthodes évolutionnaires. Si les algorithmes évolutionnaires multi-objectifs (MOEA) se sont avérés robustes pour la résolution de problèmes multiobjectifs, leurs performances dépendent largement des techniques de gestion des contraintes pour les problèmes fortement contraints. Le cœur de l'innovation de cette étude consiste en l'adaptation d'outils basés sur les métaheuristiques à cette classe de problèmes en Génie des Procédés. Dans ce but, la stratégie de recherche a comporté deux volets.

Tout d'abord, une étude empirique a été réalisée afin de comparer différentes configurations algorithmiques et sélectionner la meilleure pour fournir des approximations de fronts de Pareto de haute qualité. Cette étude, comprenant à la fois des problèmes de test académiques et applications en Génie des Procédés, a montré qu'une méthode utilisant le gradient de contraintes pour réparer les solutions infaisables obtenait les meilleurs résultats, en particulier pour le traitement des contraintes d'égalité. En capitalisant sur l'expérience acquise lors de cette étude numérique préliminaire, la conception optimale de chaînes logistiques durables « hydrogène » (HSC), prenant en compte des critères économiques et environnementaux, est étudiée.

Une méthode MOEA basée sur la décomposition et combinée à la réparation basée sur le gradient, a d'abord été exploré pour résoudre le problème. Cependant, en raison du nombre important de bilans massiques (contraintes égalité), cette approche a montré une faible convergence vers le front de Pareto optimal. Une nouvelle stratégie a donc été développée et adaptée à ce problème, à travers une reformulation en deux niveaux : le niveau supérieur (discret) traite le problème de conception de la structure de la HSC (dimensionnement et emplacement des installations), tandis que le niveau inférieur (problème de programmation linéaire) résout le sous-problème opérationnel correspondant (production et transport). Cette stratégie permet le développement d'une technique de solution matheuristique ad-hoc, par l'hybridation d'un MOEA avec un solveur LP utilisant une fonction de scalarisation pour traiter les deux objectifs considérés. Les résultats numériques obtenus pour l'étude de cas de la région Occitanie soulignent que l'approche hybride produit une bonne approximation du front de Pareto, et ce plus efficacement que les méthodes exactes.

Enfin, la matheuristique a permis d'étudier le problème de conception de la HSC avec des hypothèses plus réalistes concernant les technologies utilisées pour la synthèse de l'hydrogène, les taux d'apprentissage reflétant la maturité croissante de ces technologies au fil du temps et les relations non linéaires pour le calcul des dépenses d'investissement et d'exploitation (CAPEX et OPEX) des installations de production d'hydrogène. Le nouveau modèle, qui fait intervenir une formulation bi-objectif mixte non linéaire (MINLP), peut être résolu efficacement par l'algorithme hybride proposé.

Resumen

Los enfoques sistémicos en Ingeniería de Procesos hacen intervenir frecuentemente problemas complejos de optimización, los cuales suelen resolverse mediante enfoques deterministas. En particular, dichos problemas implican variables mixtas, funciones no lineales, un considerable número de restricciones, además de múltiples criterios antagónicos que deben optimizarse simultáneamente, lo cual supone un reto para los métodos clásicos de optimización. La principal motivación de esta investigación es, por tanto, explorar métodos de resolución alternativos para abordar problemas de optimización multiobjetivo en Ingeniería de Procesos, teniendo en cuenta los avances recientes en Computación Evolutiva. A este respecto, los algoritmos evolutivos multiobjetivo (MOEAs, por sus siglas en inglés) han demostrado ser robustos para la solución de problemas multiobjetivo, sin embargo, su rendimiento depende en gran medida de las técnicas de manejo de restricciones. La innovación de este trabajo es la adaptación de metaheurísticas de estado del arte a problemas en Ingeniería de Procesos.

En primer lugar, un estudio experimental fue realizado afín de comparar diferentes configuraciones algorítmicas y seleccionar la mejor, capaz de proporcionar aproximaciones precisas del frente de Pareto. Este estudio, que comprende tanto problemas académicos como de aplicación en Ingeniería de Procesos, demostró que un método que utiliza la información del gradiente de las restricciones para reparar las soluciones no factibles obtiene los mejores resultados, en particular en problemas con restricciones de igualdad. A partir de estos experimentos preliminares, se investigó el diseño óptimo de cadenas de suministro de hidrógeno (HSC, por sus siglas en inglés), teniendo en cuenta criterios económicos y medioambientales.

En primer lugar, se exploró como técnica de resolución un MOEA basado en la descomposición combinado con la reparación basada en el cálculo del gradiente. Sin embargo, debido al importante número de balances de materia (restricciones de igualdad), este enfoque mostró una pobre convergencia hacia el frente de Pareto óptimo. Por lo tanto, se desarrolló una nueva estrategia de resolución adaptada a este problema, basada en una reformulación binivel: el nivel superior (problema discreto) aborda el problema de diseño de la estructura de la HSC (dimensionamiento y ubicación de las instalaciones), mientras que el nivel inferior (problema lineal) resuelve el subproblema correspondiente a la operación de la cadena de suministro (producción y transporte). De este modo, la técnica de resolución consiste en la hibridación de un MOEA con un solver de programación lineal, que utiliza una función de escalarización para tratar los dos objetivos considerados. Los resultados computacionales obtenidos para el caso de estudio de la región Occitanie indican que el enfoque híbrido produce una aproximación precisa del frente de Pareto óptimo, de forma más eficiente que los métodos deterministas.

Por último, la técnica híbrida permitió estudiar el problema de diseño de la HSC con supuestos más realistas en cuanto a las tecnologías utilizadas para la síntesis de hidrógeno, las tasas de aprendizaje que reflejan la creciente madurez de estas tecnologías en función del tiempo, así como las relaciones no lineales para el cálculo de los costos de capital y de operación. El modelo resultante, formulado como un problema bi-objetivo de programación entera mixta no lineal (MINLP), puede resolverse eficientemente mediante el algoritmo híbrido propuesto.

Contents

List of Figures

List of Tables

List of Acronyms

AASF	Augmented achievement scalarizing function
AEL	Alkaline electrolyzer
ATP	Adaptive threshold penalty
BEV	Battery electric vehicle
CAPEX	Capital expenditure
CCUS	Carbon capture, utilization and storage
CDP	Constraint dominance principle
CMOP	Constrained multiobjective optimization problem
DE	Differential evolution
EA	Evolutionary algorithm
EC	Evolutionary computation
FCEV	Fuel cell electric vehicle
GAMS	General algebraic modeling system
GHG	Greenhouse gas
GWP	Global warming potential
HSC	Hydrogen supply chain
HV	Hypervolume indicator
IGD	Inverted generational distance
LCA	Life-cycle assessment
LP	Linear programming
MILP	Mixed integer linear programming
MINLP	Mixed integer nonlinear programming
MOEA	Multiobjective evolutionary algorithm
MOEA/D	Multiobjective evolutionary algorithm based on decomposition
MOP	Multiobjective optimization problem
MP	Mathematical programming
NLP	Nonlinear programming
NPV	Net present value
NSGA-II	Non-dominated sorting genetic algorithm-II
OPEX	Operational expenditure
PEM	Proton exchange membrane
PSE	Process systems engineering
SMR	Steam methane reforming
SMS-EMOA	$\mathcal{S}$ metric selection evolutionary multiobjective algorithm
SR	Stochastic ranking
TDC	Total daily cost

Introduction

General context

The main research goals in process systems engineering (PSE) consist in the modeling, design, synthesis, simulation and optimization of processes for transforming products. At a macroscopic level, these tasks necessarily involve preliminary studies for the synthesis and design of production systems, in order to numerically generate different scenarios that allow identifying appropriate structures and operating mode policies or to optimize one or different performance criteria. Historically, these preliminary studies used to address the maximization of yield or efficiency of production, storage and distribution systems, or the minimization of both capital and operating costs, as well as costs related to energy and material feedstocks. These goals, from a mathematical optimization point of view, can be considered as objective functions and combined into net present value (NPV) functions.

However, the current vision of PSE is the integration of the above-mentioned processes within their surrounding territory, particularly including environmental and societal concerns. To this end, different methodologies or tools providing an evaluation of the impact of industrial activities over humankind and environment have been proposed. Some of them consist in assigning a straightforward cost to these social or environmental impacts, while others are more sophisticated, such as Life-Cycle Assessment (LCA) that involves a cradle-to-grave approach. It is important to note that, while considering these new approaches embedded in PSE, the optimization of environment-based or society-based objectives should not be at the expense of economic criteria such as cost minimization or NPV maximization. This is why the simultaneous optimization of different kinds of objectives (economic, environmental, etc.) must be considered, thus leading to the treatment of multiobjective optimization problems (MOPs).

In order to optimize the variety of performance criteria as those described above, the relevant features of the considered systems are captured within optimization models that involve the definition of decision variables and constraints, in addition to objective function(s). Typically, optimization problems related to PSE include constraints (both equality and inequality), continuous and discrete variables (hence discontinuous search space) and nonlinearities. The equality constraints usually represent mass and energy balances, whereas inequality constraints may refer to some bounds on any output variable, e.g., a production time that must be lower than a horizon time limit, the concentration of a contaminant in some plant effluent that should respect some environmental upper limit, some bounds related to production plant capacities, and so on. Further, the continuous decision variables may represent physical parameters, e.g., operating conditions, stream flow rates or compositions, while discrete variables may account for the existence of some structures, e.g., number of trays in a distillation column, number of transformation (reactor) units or number of production items in a processing stage. Depending on the problem, the number of variables and constraints can turn out to be relatively high.

As a summary, the optimization problems formulated in PSE applications generally present the following properties: (1) they are typically multiobjective, i.e., several conflicting criteria must be simultaneously optimized, (2) they might involve an

important number of variables of different natures (continuous/discrete) implying high computational complexities (many of such problems have been proven to be NP-hard), (3) they include a significant number of constraints and (4) the mathematical functions involved in the corresponding optimization models might be nonlinear and sometimes non-convex. According to the type of variables and constraints they present, their mathematical models can be classified into LP (Linear Programming), MILP (Mixed-Integer Linear Programming), NLP (Nonlinear Programming) and MINLP (Mixed-Integer Nonlinear Programming).

Throughout the years, mathematical programming (MP) techniques have been used to address the solution of these problems. Several algorithms have been proposed to solve to optimality these problems (for a review, see Floudas and Gounaris, 2009; Tawarmalani and Sahinidis, 2013). It should be noted, however, that these deterministic methods rely on the determination of a "good" initial solution (i.e., close to the optimum) or on convexity assumptions to guarantee global optimality. Besides, their performance may strongly depend on the problem formulation, meaning that a different algebraic form of the same equations might lead to quite different algorithmic performances (Liberti, 2008), in terms of both efficacy and efficiency. In other words, the problem needs to satisfy some specific mathematical characteristics (e.g., convexity, derivability) so that an automatic valid reformulation can be generated, otherwise, the reformulated convex problem might miss the original global optimum and converge to a local optimum. Furthermore, the application of these exact techniques can be computationally expensive to obtain the rigorous global solution of large-scale problems. A final issue is the multiobjective aspect of the problem, which is usually addressed through classical techniques such as weighted sum aggregation functions or ε-constraint. Nevertheless, the drawbacks of these classical scalarizing strategies are well known, these are:

1. Multiple executions are needed to construct an accurate approximation of the whole Pareto front (one run to obtain one solution).

2. The resulting scalar optimization problems are parametrized (for example by a weight vector) to explore a specific subregion of the search space. To produce an evenly distributed Pareto frontier approximation, the corresponding parameters should be appropriately tuned, which might turn into a harsh task because of the nonlinear relationship between decision variables and cost functions.

An alternative to MP techniques is to use metaheuristics, formally defined as top-level general strategies which guide other lower-level heuristics[1] to search for feasible solutions in difficult domains or search landscapes (Coello Coello et al., 2007, p. 63). These methods are typically characterized by their stochastic nature, their ability to provide good-quality solutions in tractable computational times (though guarantee of optimality cannot be provided) and by the fact that no particular mathematical properties of the problem are required, e.g., derivability or convexity. In the context of multiobjective optimization, multiobjective evolutionary algorithms (MOEAs) stand as powerful bio-inspired search techniques suitable to solve such types of problems. It is important to recall that MOEAs have been proposed as efficient solution methods for tackling highly multimodal (non-convex) multiobjective optimization problems. Besides, MOEAs can provide an approximation of the Pareto front in one single run

[1] *Heuristic*: a problem-solving technique in which the most appropriate local solution or partial solution is selected using comparative rules.

and are less susceptible to the shape or continuity of the Pareto front. Regarding the use of these population-based algorithms in PSE literature, the main and almost the only reference is the NSGA-II (Deb et al., 2002), which became really popular in a wide range of applications due to its claimed superiority over other MOEAs of the same generation (late 1990s and early 2000s), such as the Pareto Archived Evolutionary Strategy (PAES), introduced in (Knowles and Corne, 1999), or the Strength Pareto Evolutionary Algorithm 2 (SPEA2) (Zitzler et al., 2001). However, almost no PSE study reports the use of alternative and more recent metaheuristic multiobjective solution methods, for example, MOEA/D (Zhang and Li, 2007), which works under the decomposition-based paradigm; SMS-EMOA (Emmerich et al., 2005), an indicator-based algorithm aiming directly for the optimization of a performance metric; AMOSA (Bandyopadhyay et al., 2008), a Pareto dominance-based algorithm that incorporates a probability of selecting dominated solutions, similar to that employed in the simulating annealing algorithm; or MOPSO (Zapotecas-Martínez and Coello Coello, 2011), which extends the use of particle swarm optimization algorithms to multiobjective optimization under a decomposition-based approach.

At this point, it should be emphasized that the applicability of MOEAs to PSE problems is usually limited by a deficiency of these techniques to manage constraints, and in particular, equality constraints that might involve an important number of decision variables, such as those representing mass and energy balances in the PSE area. In this respect, on the one hand, Evolutionary Computation (EC) researchers have proposed a number of sophisticated strategies for constraint handling that differ on their working principle, e.g., some repair infeasible solutions at some extent, others perform stochastic comparisons among solutions, while the simplest ones penalize infeasible solutions (static penalty function) or prefer feasible solutions over infeasible ones (feasibility rules). The performance of these techniques, nevertheless, is usually measured only on some academic benchmarks that might include unrealistic properties (or neglect some important ones) which may lead to overestimate/underestimate them. In fact, almost no comparative study includes equality-constrained problems, similar to those related to PSE. On the other hand, in what regards the constraint-handling techniques employed in the PSE literature, the vast majority of related works use *old* strategies such as penalty functions or feasibility rules, or carry out a problem reformulation aiming to reduce, as many as possible, the number of constraints. These strategies, though efficient for solving small-size problems, are likely to be inefficient for solving large-size PSE case studies.

In addition, approaches that hybridize mathematical programming techniques and metaheuristics exist, and are becoming particularly popular in recent years. These approaches are known as hybrid methods or *matheuristics*. They seek to take advantage of each solution method, mitigating simultaneously their disadvantages (Jourdan et al., 2009). In many cases, better results than those obtained by each method separately are obtained using this kind of approaches, especially in real-life large-size problems. Besides, some recently published works related to PSE area present solution tools based on this kind of hybridization (see, for instance, Hernández-Pérez et al. (2020), López-Flores et al. (2021), and Teh et al. (2019)).

Summarizing, though significant advances are found in the literature dedicated to metaheuristics design, with a particular focus on strategies dealing with multiple objectives and constraint-handling, only a few have been explored in the PSE area, where classical deterministic optimization methods are usually employed although not being the most appropriate approaches, in particular for problems considering

multiple criteria or containing non-convex search spaces. Therefore, the scientific objective of this thesis is to investigate the relevance of state-of-the-art MOEAs as alternative methods for the solution of complex PSE optimization problems. This issue is treated, in a first part, through an empirical study devoted to the comparison of MOEAs operators applied to academic and PSE test problems, with an increasing complexity of problem features (number of objectives, type of constraints, continuous/mixed variables). In a second part, the optimal design of the hydrogen supply chain (HSC), which is being studied in the research team, will serve as an application support for the study.

An empirical study on constraint-handling with multiple objectives

As mentioned previously, evolutionary algorithms constitute potential alternatives to exact methods, in particular for tackling multiobjective optimization problems. Nevertheless, since MOEAs experience issues for working with highly constrained search spaces, a preliminary study on constraint-handling techniques in MOEAs is necessary for the correct evaluation of this kind of methods on PSE problems. In this vein, a numerical comparison on some state-of-the-art constraint-handling techniques in metaheuristics is carried out in a single-objective framework, for which these techniques were first proposed. Hence, in Chapter 2, 14 constrained problems related to the PSE area are investigated. The conclusions drawn in this chapter, suggesting a significant superiority of the method using the gradient information of constraints, are then explored in a multiobjective framework in the subsequent chapter.

In Chapter 3, the gradient-based repair method is embedded within six classical constraint-handling techniques and two different MOEAs: NSGA-II and MOEA/D. The test functions include classical academic benchmark problems with inequality constraints, as well as recent problems with equality constraints, so as to identify the most promising technique to be used later in an engineering problem.

Optimal design of hydrogen supply chains as a support of the methodology

The growing concern about the depletion of fossil energy sources, such as oil and gas, as well as the degradation of the environment by the combustion of these conventional fuels, has motivated the search for a more sustainable energy model based on renewable energy systems. In this context, hydrogen represents a potential alternative to fossil fuels by enabling meaningful reductions of CO_2 emissions in multiple sectors like industry, building and transportation (Brey, 2020). In addition, hydrogen can be produced from a variety of feedstocks, including fossil fuels (coupled with processes that capture, utilize and store CO_2, CCUS) and, more importantly, from renewable sources like biomass, wind or solar energy. It can consequently store surplus power from renewable energies when the electrical grid cannot absorb it (Carrera Guilarte and Azzaro-Pantel, 2020; IEA, 2019). In this research work, only the mobility sector is considered, yet the methodology developed shall be replicable to other sectors.

The optimal design of the sustainable hydrogen supply chain constitutes a current challenge to society as it gives the basis for the evaluation of a cost-efficient hydrogen-based economy. The variety of alternatives for hydrogen production, transportation, storage and distribution, makes the design and management of the hydrogen supply chain a complex task. This issue generally uses a set of approaches to efficiently

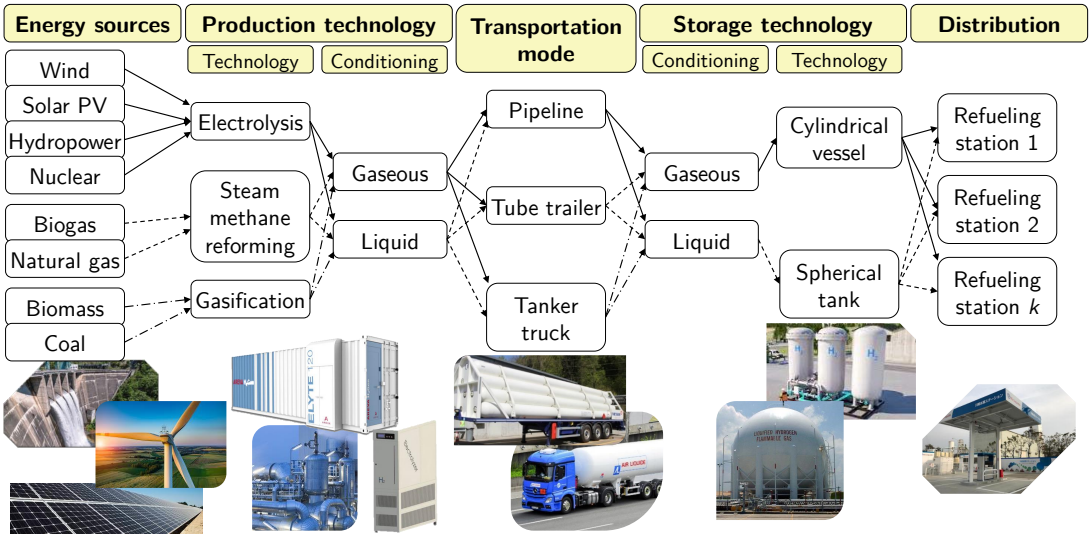


Figure 1: Hydrogen supply chain for mobility uses.

integrate each aspect of the supply chain, so that hydrogen is produced and distributed at the right quantities, to the right locations and at the right time, in order to minimize the system global cost and environmental impact while satisfying service level requirements. A hydrogen supply chain involves multiple echelons from energy source, production, storage, transportation and distribution that interact along the chain (see Figure 1). These multiple echelons are within the scope of this thesis.

Several works have been published that propose a superstructure representing all the possible paths in the supply chain, with their corresponding mathematical model. Then, the mathematical formulation of the optimization problem usually takes the form of a multiobjective mixed-integer linear programming problem, though only the bi-objective case (economic and environmental criteria) is usually considered. The solution strategy commonly adopted uses a commercial MILP solver (generally, CPLEX) combined with approaches based on scalarizing techniques, e.g., weighted sum or ε-constraint. This strategy, although effective for finding some solutions belonging to the true Pareto front, appear to be computationally prohibitive for treating large-size instances or for providing accurate Pareto front approximations, mainly because of the combinatorial aspects of the problem and the issue of appropriately tuning the scalarizing parameters (weight vectors or ε-levels) for each run. Some significant works using this strategy are Almansoori and Shah (2012), Almaraz et al. (2015), Câmara et al. (2019), Hugo et al. (2005), and Kim and Moon (2008).

Further, with the aim of mitigating the numerical difficulties associated with the solution of large-scale instances, some works proposed a strategy based on a bilevel (master-slave) decomposition (Guillén-Gosálbez et al., 2010; Sabio et al., 2010). Though these methods proved to be efficient for addressing the bi-objective case of the problem, they make use of ε-constraint method for addressing the multiobjective aspect of the problem, and thus might become impractical if more objectives are considered.

Combining deterministic and metaheuristic approaches

In the second part of this thesis, the scientific objective is to develop an adapted solution technique for the problem the design of hydrogen supply chains. First, the conclusions obtained from the first part on constraint-handling are exploited to propose a first strategy using MOEAs. Then, a novel strategy using a matheuristic is developed, that is, a hybrid optimization method that combines elements of both exact and stochastic methods in a cooperative way (Ball, 2011; Jourdan et al., 2009). More precisely, this solution strategy is based on bilevel decomposition, so that appropriate solution methods can be applied to the decisions levels involved. The upper level problem considers the installation of production and storage facilities (i.e., a multiobjective combinatorial problem), while the lower level focuses on the problem associated to transportation and production rates (linear programming problem of low complexity). The solution of the master problem is performed by a MOEA, taking thus full advantage of its potential to tackle the multiobjective and combinatorial features of the problem. Then, for each partial solution proposed by the evolutionary algorithm, the slave problem is addressed by a linear programming solver. In this way, the multiobjective bilevel problem is solved in an iterative manner, in one single run.

The interest of such a strategy is that it allows exploring a more realistic formulation of the problem, one that integrates algebraic sizing-cost relationships for production and storage facilities. As a consequence, the resulting model involves nonlinear terms since installation costs generally follow a classical "six-tenth" power rule for scaling equipment and plant costs. Thus, in spite of the fact that the resulting model takes the form of a MINLP problem, the proposed hybrid methodology still serves as an efficient approach for tackling this problem, because each nonlinear term is to be handled by the MOEA at the upper level, whereas the lower level will remain a LP problem. Finally, this methodology is able to provide details encompassing simultaneously spatial, temporal and technological aspects.

Scientific objective

The motivation of this thesis is to explore the applicability of recent advances on Evolutionary Computation in the Process Systems Engineering area. More precisely, these recent advances comprise (1) sophisticated constraint-handling techniques, different from penalty functions or feasibility rules, and (2) MOEAs that are constructed on paradigms different from Pareto dominance, i.e., decomposition and performance indicators. In this way, alternative efficient solution methods may be identified for tackling PSE optimization problems. Besides, a case study of current interest is investigated, namely, the multiobjective design of hydrogen supply chains (HSC), through state-of-the-art strategies that employ both MOEAs and MP techniques.

Therefore, taking into account the context previously presented, the four following research questions that arise are:

1. What is the best solution approach to tackle current PSE problems, typically formulated as multiobjective optimization problems?

2. Considering the recent advances in the metaheuristics field, in particular those related to constraint-handling, do they constitute efficient techniques for treating highly-constrained problems as those related to PSE?

3. Can a matheuristic approach be designed for the solution of the HSC design problem, so that it outperforms classical exact methods, thus providing solutions in tractable computational times?

4. And finally, can a matheuristic approach be designed as a general solution tool for a broad range of PSE problems?

The present work aims to provide answers to these research questions throughout each chapter, and they will be directly addressed in the last section (conclusions and perspectives).

Thesis structure

This thesis is composed of two main parts comprising seven chapters. The first part, which include the first three chapters, is mainly devoted to an empirical study on constraint-handling for multiobjective optimization algorithms. The second part (Chapters 4 to 7), is dedicated to the optimal design of hydrogen supply chains.

In particular, the Chapter 1 presents a general overview on multiobjective optimization; it covers some definitions and formal notation employed throughout this thesis. It also presents the classical scalarizing techniques generally used within mathematical programming approaches, and then introduces multiobjective evolutionary algorithms. The main components common to any MOEA are briefly described, as well as some popular genetic operators, and the three different selection paradigms in MOEAs. For each "family" of algorithms, a representative algorithm is presented in deeper details: NSGA-II, MOEA/D and SMS-EMOA.

The Chapter 2 presents a comparative study on constraint-handling techniques in metaheuristics, for the solution of PSE optimization problems, in a single-objective framework. The study takes into account 5 constraint-handling techniques and 14 PSE problems of small and medium sizes that, despite of their sizes, entail difficulties to classical techniques for converging to the global optimum. The obtained results showed the superiority of the method based on the repair of infeasible solutions using the constraint gradient (never used before in the PSE area), in particular when treating highly constrained problems.

The Chapter 3, based on the same philosophy, presents similar experiments in the context of multiobjective optimization. The purpose is to verify that the results obtained for the single-objective case are valid for problems with multiple objectives. To this end, the gradient-based repair is embedded within six state-of-the-art constraint-handling techniques, in their canonical version. The respective performances of these different methods, with and without the use of the repair technique, included within two classical MOEAs (NSGA-II and MOEA/D), are compared. Experiments were carried out on popular test suites drawn from the EC literature, as well as recent problems containing equality constraints.

The Chapter 4 introduces the context of optimal design of hydrogen supply chains. It provides the essential features of this important case study, describing in details every echelon of the supply chain. Thus, it constitutes the basis for the following chapters.

The Chapter 5 investigates the applicability of MOEAs for the optimal design of hydrogen supply chains. The results obtained using MOEA/D with gradient-based

repair method are presented and discussed. The need for a solution technique more adapted to the problem features is highlighted, leading to the use of a hybrid strategy.

The Chapter 6 is devoted to the presentation of a novel matheuristic based on bilevel optimization for the multiobjective design of hydrogen supply chains. This chapter also presents the numerical experiments made for the validation of this strategy. In a first step, three instances of small to medium sizes and a single-period were solved using MOEA/D at the upper level. Then, six instances of larger size were studied, using a more appropriate algorithm for two-objective real-world problems, SMS-EMOA.

The Chapter 7 is dedicated to the presentation of a formulation of the HSC that captures spatial, temporal and technological details. The modifications to the base MILP model are explained in a first part, with a particular focus on the impact on the solution approach. The second part of this chapter is dedicated to the presentation of the numerical results and their discussion.

Finally, general conclusions of this thesis are drawn, along with some perspectives for future work. Appendix A presents the mathematical formulation of the problems studied in the first chapter, as well as the information regarding local and global optimal solutions. Appendix B contains the nomenclature related to the HSC case study, as well as the data for the different instances.

The doctoral research presented in this manuscript was conducted at the Laboratory of Chemical Engineering, UMR 5503 CNRS/INP/UPS in the Process Systems Engineering department from March, 2018 to July, 2021. It is worth highlighting that the work presented in this thesis has been presented to the scientific community, in particular through two research articles in international journals, two conference papers and one book chapter:

- Victor H Cantú, Catherine Azzaro-Pantel, and Antonin Ponsich (2021). "Constraint-handling techniques within differential evolution for solving process engineering problems". In: *Applied Soft Computing* 108, p. 107442. DOI: https://doi.org/10.1016/j.asoc.2021.107442.

- Victor H Cantú, Catherine Azzaro-Pantel, and Antonin Ponsich (accepted). "A novel matheuristic based on bi-level optimization for the multi-objective design of hydrogen supply chains". In: *Computers & Chemical Engineering*, p. 107370. DOI: https://doi.org/10.1016/j.compchemeng.2021.107370.

- Victor H Cantú, Catherine Azzaro-Pantel, and Antonin Ponsich (2020a). "Multiobjective evolutionary algorithm based on decomposition (MOEA/D) for optimal design of hydrogen supply chains". In: *Computer Aided Chemical Engineering*. Ed. by Sauro Pierucci, Flavio Manenti, Giulia Luisa Bozzano, and Davide Manca. Vol. 48. Elsevier, pp. 883–888. DOI: https://doi.org/10.1016/B978-0-12-823377-1.50148-8.

- Victor H Cantú, Catherine Azzaro-Pantel, and Antonin Ponsich (2020b). "Optimal design of hydrogen supply chains by a multiobjective evolutionary algorithm based on decomposition (MOEA/D)". in: *EasyChair ROADEF2021*.

- Victor H Cantú, Antonin Ponsich, and Catherine Azzaro-Pantel (Apr. 2021). "Constraint Handling in Metaheuristics and Applications". In: ed. by Anand J Kulkarni, Efrén Mezura-Montes, Yong Wang, Amir H Gandomi, and Ganesh Krishnasamy. 1st ed. Springer Singapore. Chap. On the use of gradient-based

repair method for solving constrained multiobjective optimization problems – A comparative study, pp. 119–149. DOI: 10.1007/978-981-33-6710-4.

Introduction

Contexte général

Les principaux objectifs de recherche en Procédés et Systèmes Industriels (PSI) consistent à modéliser, synthétiser, simuler et optimiser les procédés de transformation des produits. À un niveau macroscopique, ces tâches impliquent nécessairement des études préliminaires pour la synthèse et la conception de systèmes de production, afin de générer numériquement différents scénarios permettant d'identifier les structures et les politiques de mode de fonctionnement appropriées, ou d'optimiser un ou plusieurs critères de performance. Historiquement, ces études préliminaires ont eu pour but de maximiser le rendement ou l'efficacité des systèmes de production, de stockage et de distribution, ou de minimiser les coûts d'investissement et d'exploitation, ainsi que les coûts liés à l'énergie et aux matières premières. Ces critères peuvent être considérés, du point de vue de l'optimisation mathématique, comme des fonctions objectif, éventuellement combinés en fonctions de valeur actuelle nette (VAN).

Cependant, la vision actuelle en PSI est l'intégration des processus susmentionnés dans leur territoire environnant, incluant notamment des préoccupations environnementales et sociétales. A cette fin, différentes méthodologies ou outils permettant d'évaluer l'impact des activités industrielles sur l'homme et l'environnement ont été proposés. Certains d'entre eux consistent à attribuer un coût direct à ces impacts sociaux ou environnementaux, tandis que d'autres sont plus sophistiquées, comme l'analyse du cycle de vie (ACV) qui implique une approche « du berceau à la tombe ». Il est important de noter que, tout en considérant ces nouvelles approches intégrées aux PSI, l'optimisation des objectifs environnementaux ou sociaux ne doit pas se faire au détriment des critères économiques, tels que la minimisation des coûts ou la maximisation de la VAN. C'est pourquoi l'optimisation simultanée de différents types d'objectifs (économiques, environnementaux, etc.) doit être envisagée, conduisant au traitement de problèmes d'optimisation multi-objectif.

Afin d'optimiser la variété de critères de performance tels que ceux décrits ci-dessus, les caractéristiques pertinentes des systèmes considérés sont représentées au moyen de modèles d'optimisation qui impliquent la définition de variables de décision et de contraintes, en plus de la (ou des) fonction(s) objectif. Généralement, les problèmes d'optimisation liés au domaine des PSI comprennent des contraintes (égalité et inégalité), des variables continues et discrètes (d'où un espace de recherche discontinu) et des non-linéarités. Les contraintes égalité représentent généralement des bilans de masse et d'énergie, tandis que les contraintes d'inégalité font en général référence à certaines limites sur une variable de sortie, par exemple, un temps de production qui doit être inférieur à un horizon de temps limité, la concentration d'un contaminant dans un effluent d'usine qui doit respecter une limite supérieure environnementale, certaines limites liées aux capacités de l'usine de production, etc. Par ailleurs, les variables de décision continues peuvent représenter des paramètres physiques, par exemple des conditions de fonctionnement, des débits ou des compositions de flux, tandis que les variables discrètes peuvent rendre compte de l'existence de certaines structures, par exemple le nombre de plateaux dans une colonne de distillation, le nombre d'unités de transformation (réacteurs) ou le nombre d'équipements

de production dans une étape de traitement. Selon le problème, le nombre de variables et de contraintes peut s'avérer relativement élevé.

En résumé, les problèmes d'optimisation formulés dans les applications de PSI présentent généralement les propriétés suivantes : (1) ils sont typiquement multi-objectifs, c'est-à-dire que plusieurs critères antagonistes doivent être optimisés simultanément, (2) ils peuvent impliquer un nombre important de variables de différentes natures (continues/discrètes), impliquant des complexités de calcul élevées (beaucoup de ces problèmes se sont avérés NP-difficiles), (3) ils incluent un nombre significatif de contraintes et (4) les fonctions mathématiques impliquées dans les modèles d'optimisation correspondants peuvent être non linéaires et parfois non convexes. En fonction du type de variables et de contraintes qu'ils présentent, les modèles mathématiques associés à ces problèmes peuvent être classés en PL (Programmation Linéaire), PLM (Programmation Linéaire en variables Mixtes), PNL (Programmation Non Linéaire) et PNLM (Programmation Non Linéaire en variables Mixtes).

Au fil des années, ce sont les techniques de programmation mathématique (PM) qui ont en général été utilisées pour résoudre ces problèmes. Plusieurs algorithmes ont été proposés pour résoudre ces problèmes de manière optimale (pour un panorama complet, voir Floudas and Gounaris, 2009; Tawarmalani and Sahinidis, 2013). Il convient toutefois de noter que ces méthodes déterministes reposent sur la détermination d'une "bonne" solution initiale (c'est-à-dire proche de l'optimum) ou sur des hypothèses de convexité pour garantir l'optimalité globale. En outre, leur performance peut dépendre fortement de la formulation du problème, ce qui signifie qu'une forme algébrique différente des mêmes équations peut conduire à des performances algorithmiques très différentes (Liberti, 2008), en termes d'efficacité et de qualité de solutions trouvées. En d'autres termes, le problème doit satisfaire certaines caractéristiques mathématiques spécifiques (par exemple, la convexité, la dérivabilité) pour qu'une reformulation automatique valide puisse être générée. Sinon, la reformulation convexe pourrait manquer exclure l'optimum global du problème original et amener à une convergence vers un optimum local. En outre, l'application de ces techniques exactes peut être coûteuse en termes de calcul pour obtenir la solution globale rigoureuse de problèmes à grande échelle. Enfin une dernière question est l'aspect multiobjectif du problème, qui est généralement traité par des techniques classiques telles que les fonctions d'agrégation (par exemple, somme pondérée) ou la technique de contraintes ε. Néanmoins, les inconvénients de ces stratégies classiques de scalarisation sont bien connus, en particulier :

1. Plusieurs exécutions sont nécessaires pour construire une approximation précise de l'ensemble du front de Pareto (une exécution nécessaire par solution).

2. Les problèmes d'optimisation scalaire qui en résultent sont paramétrés (par exemple par un vecteur de poids) pour explorer une sous-région spécifique de l'espace de recherche. Pour que l'approximation de la frontière de Pareto soit distribuée de manière uniforme, les paramètres correspondants doivent être ajustés de façon appropriée, ce qui peut s'avérer une tâche difficile en raison de la relation non linéaire entre les variables de décision et les fonctions de coût.

Une alternative aux techniques de PM consiste à utiliser des métaheuristiques, formellement définies comme des stratégies générales de niveau supérieur, guidant d'autres heuristiques de niveau inférieur pour rechercher des solutions réalisables dans des espaces de recherche complexes (Coello Coello et al., 2007, p. 63). Ces

méthodes sont généralement caractérisées par leur nature stochastique, leur capacité à fournir des solutions de bonne qualité en des temps de calcul contrôlables (bien qu'une garantie d'optimalité ne puisse être fournie) et par le fait qu'aucune propriété mathématique particulière du problème n'est requise, par exemple, la dérivabilité ou la convexité. Dans le contexte de l'optimisation multiobjectif, les algorithmes évolutionnaires multi-objectifs (MOEA) sont de puissantes techniques de recherche bio-inspirées, adaptées à la résolution de ce type de problèmes. Il est important de rappeler que les MOEA ont été proposés comme des méthodes efficaces pour résoudre les problèmes d'optimisation multi-objectifs hautement multimodaux (non convexes). En outre, les MOEA peuvent fournir une approximation du front de Pareto en une seule exécution et sont moins sensibles à la forme ou à la continuité du front de Pareto optimal. En ce qui concerne l'utilisation de ces algorithmes distribués (c'est-à-dire, basés sur l'évolution d'une population de solutions) dans la littérature en PSI, la principale et presque unique référence est l'algorithme NSGA-II (Deb et al., 2002), très populaire dans un large éventail d'applications en raison de sa supériorité par rapport à d'autres MOEA de la même génération (fin des années 1990 et début des années 2000), tels que les algorithmes PAES (Knowles and Corne, 1999), ou SPEA2 (Zitzler et al., 2001). Cependant, presque aucune étude en PSI ne fait état de l'utilisation de méthodes alternatives et plus récentes, par exemple, MOEA/D (Zhang and Li, 2007), qui fonctionne selon le paradigme de décomposition ; SMS-EMOA (Emmerich et al., 2005), un algorithme basé sur des indicateurs visant directement l'optimisation d'une métrique de performance ; AMOSA (Bandyopadhyay et al., 2008), un algorithme basé sur la dominance de Pareto qui incorpore une probabilité de sélection des solutions dominées, similaire à celle employée dans l'algorithme de recuit simulé ; ou MOPSO (Zapotecas-Martínez and Coello Coello, 2011), qui étend l'utilisation des algorithmes d'optimisation par essaims de particules à l'optimisation multiobjectif, dans le cadre d'une approche basée sur la décomposition.

À ce stade, il convient de souligner que l'applicabilité des MOEA aux problèmes de PSI est généralement limitée par l'incapacité de ces techniques à gérer des contraintes, en particulier les contraintes égalité qui peuvent impliquer un nombre important de variables de décision, comme celles qui représentent les bilans massiques et énergétiques dans le domaine des PSI. À cet égard, les chercheurs en Algorithmes Évolutionnaires (AE) ont proposé un certain nombre de stratégies sophistiquées pour la gestion des contraintes, qui diffèrent par leur mode de fonctionnement. Par exemple, certaines réparent les solutions infaisables dans une certaine mesure, d'autres effectuent des comparaisons stochastiques entre les solutions, tandis que les plus simples pénalisent les solutions infaisables (fonction de pénalité statique) ou préfèrent les solutions faisables aux solutions infaisables (règles de faisabilité). Néanmoins, la performance de ces techniques n'est généralement évaluée que sur des problèmes test académiques, qui peuvent inclure des propriétés irréalistes (ou négliger certaines propriétés importantes), ce qui peut conduire à les surestimer/sous-estimer. En fait, presque aucune étude comparative n'inclut les problèmes comportant des contraintes égalité, similaires à ceux liés aux PSI. D'autre part, en ce qui concerne les techniques de traitement des contraintes employées dans la littérature associée aux PSI, la grande majorité des travaux utilisent des stratégies plutôt anciennes, telles que les fonctions de pénalité ou les règles de faisabilité, ou effectuent une reformulation du problème visant à réduire, autant que possible, le nombre de contraintes. Ces stratégies, bien qu'efficaces pour résoudre des problèmes de petite taille, sont susceptibles d'être inefficaces pour résoudre des cas d'étude de grande

taille en PSI.

Par ailleurs, il existe des approches hybrides entre les techniques de programmation mathématique et les métaheuristiques, qui sont devenues particulièrement populaires dans les dernières années. Ces approches sont connues sous le nom de méthodes hybrides ou matheuristiques. Elles cherchent à tirer avantage de chaque méthode canonique de résolution, tout en atténuant simultanément leurs inconvénients (Jourdan et al., 2009). Dans de nombreux cas, de meilleurs résultats que ceux obtenus par chaque méthode séparément sont obtenus en utilisant ce type d'approches, en particulier dans les problèmes réels de grande taille. Certains travaux récemment publiés dans le domaine des PSI présentent des outils de résolution basés sur ce type d'hybridation (voir, par exemple, Hernández-Pérez et al. (2020), López-Flores et al. (2021), and Teh et al. (2019)).

En résumé, des avancées significatives aient été réalisées dans la littérature dédiée à la conception de métaheuristiques, avec un accent particulier sur les stratégies traitant des objectifs multiples et le traitement des contraintes. Cependant, seules quelques unes ont été exploitées dans le domaine des PSI, où les méthodes d'optimisation déterministes classiques sont habituellement employées bien que n'étant pas les approches les plus appropriées, en particulier pour les problèmes considérant des critères multiples ou contenant des espaces de recherche non convexes. Par conséquent, l'objectif scientifique de cette thèse est d'étudier la pertinence de MOEA récents comme méthodes alternatives pour la résolution de problèmes d'optimisation complexes en PSI. Cette question est traitée, dans une première partie, à travers une étude empirique consacrée à la comparaison d'opérateurs évolutionnaires multiobjectif appliqués à des problèmes académiques et des problèmes test tirés de la littérature en PSI, avec une complexité croissante des caractéristiques du problème (nombre d'objectifs, type de contraintes, variables continues/mixtes). Dans une deuxième partie, la conception optimale de la chaîne d'approvisionnement en hydrogène (HSC), qui est étudiée dans l'équipe de recherche, servira de support d'application à l'étude.

Une étude empirique sur le traitement des contraintes avec des objectifs multiples

Comme mentionné précédemment, les algorithmes évolutionnaires constituent des alternatives potentielles aux méthodes exactes, en particulier pour aborder les problèmes d'optimisation multi-objectif. Néanmoins, étant donné que les MOEA rencontrent des problèmes pour travailler avec des espaces de recherche sévèrement contraints, une étude préliminaire sur les techniques de gestion des contraintes dans les MOEA est nécessaire pour une évaluation correcte de ce type de méthodes, en vue d'une application à des problèmes de PSI. Dans cette optique, une comparaison numérique de certaines techniques récentes de traitement des contraintes dans les métaheuristiques est effectuée dans un cadre mono-objectif, pour lequel ces techniques ont été proposées pour la première fois. Ainsi, dans le Chapitre 2, 14 problèmes contraints liés au domaine de PSI sont étudiés. Les conclusions tirées dans ce chapitre, suggérant une supériorité significative d'une méthode utilisant l'information du gradient des contraintes, sont ensuite exploitées dans un cadre multiobjectif dans le chapitre suivant.

Dans le Chapitre 3, la méthode de réparation par gradient des contraintes est intégrée à six techniques classiques de gestion des contraintes et à deux MOEA différents : NSGA-II et MOEA/D. Les fonctions test comprennent des problèmes académiques de

référence, incluant des contraintes inégalité, ainsi que des problèmes plus récents avec des contraintes égalité, afin d'identifier la technique la plus prometteuse à utiliser ultérieurement dans un problème d'ingénierie.

Conception optimale des chaînes d'approvisionnement en hydrogène à l'appui de la méthodologie

L'inquiétude croissante concernant l'épuisement des sources d'énergie fossiles, telles que le pétrole et le gaz, ainsi que la dégradation de l'environnement par la combustion de ces combustibles conventionnels, a motivé la recherche d'un modèle énergétique plus durable basé sur des systèmes utilisant les énergies renouvelables. Dans ce contexte, l'hydrogène représente une alternative potentielle aux combustibles fossiles permettant des réductions significatives des émissions de CO_2 dans de multiples secteurs comme l'industrie, le bâtiment et le transport (Brey, 2020). En outre, l'hydrogène peut être produit à partir de diverses matières premières, y compris les combustibles fossiles (associés à des procédés de captage, d'utilisation et de stockage du CO_2, CCUS) et, surtout, à partir de sources renouvelables comme la biomasse, l'énergie éolienne ou solaire. L'hydrogène peut par conséquent stocker le surplus d'énergie provenant des énergies renouvelables lorsque le réseau électrique ne peut l'absorber (Carrera Guilarte and Azzaro-Pantel, 2020; IEA, 2019). Dans ce travail de recherche, seul le secteur de la mobilité est considéré, mais la méthodologie développée doit pouvoir être reproduite dans d'autres secteurs.

La conception optimale de la chaîne d'approvisionnement durable en hydrogène constitue un défi actuel pour la société, car elle sert de base à l'évaluation d'une économie rentable basée sur l'hydrogène. La diversité des solutions disponibles pour la production, le transport, le stockage et la distribution de l'hydrogène fait de la conception et de la gestion de la chaîne d'approvisionnement en hydrogène une tâche complexe. Cette question fait généralement appel à un ensemble d'approches visant à intégrer efficacement chaque aspect de la chaîne d'approvisionnement, de sorte que l'hydrogène soit produit et distribué dans les bonnes quantités, aux bons endroits et au bon moment, afin de minimiser le coût global du système et l'impact environnemental tout en satisfaisant aux exigences de niveau de service. Une chaîne d'approvisionnement en hydrogène implique de multiples échelons depuis la source d'énergie, la production, le stockage, le transport et la distribution qui interagissent tout au long de la chaîne (voir la Figure 2). Ces multiples échelons entrent dans le cadre de cette thèse.

Plusieurs travaux ont été publiés, proposant une superstructure représentant toutes les options possibles dans la chaîne d'approvisionnement, ainsi que le modèle mathématique correspondant. Par a suite, la formulation mathématique du problème d'optimisation prend généralement la forme d'un problème de programmation linéaire en variables mixtes multiobjectif, bien que seul le cas bi-objectif (critères économiques et environnementaux) soit généralement considéré. La stratégie de résolution généralement adoptée utilise un solveur PLM commercial (généralement, CPLEX) combiné à des approches de scalarisation, par exemple, la somme pondérée ou la technique des contraintes ε. Cette stratégie, bien qu'efficace pour trouver certaines solutions appartenant au véritable front de Pareto, semble être prohibitive sur le plan informatique pour traiter des instances de grande taille ou pour fournir des approximations précises du front de Pareto, principalement en raison des aspects combinatoires du problème et du réglage approprié des paramètres de scalarisation

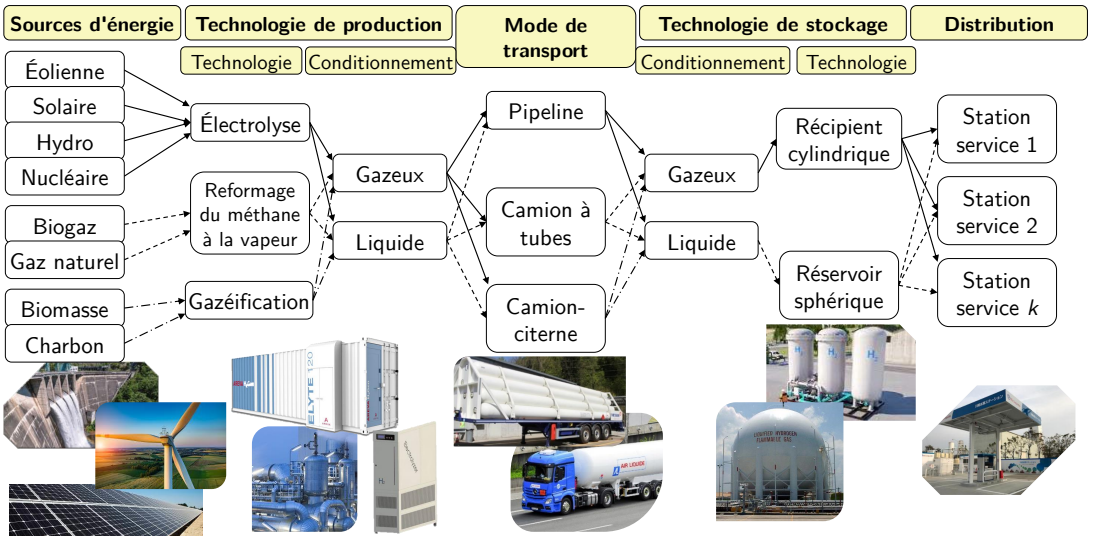


Figure 2: Chaîne d'approvisionnement en hydrogène pour les usages liés à la mobilité.

(vecteurs de poids ou niveaux ε) pour chaque exécution. Parmi les travaux importants utilisant cette stratégie, citons Almansoori and Shah (2012), Almaraz et al. (2015), Câmara et al. (2019), Hugo et al. (2005), and Kim and Moon (2008).

En outre, dans le but d'atténuer les difficultés numériques associées à la résolution d'instances de grande échelle, certains travaux ont proposé une stratégie basée sur une décomposition bi-niveaux (maître-esclave) (Guillén-Gosálbez et al., 2010; Sabio et al., 2010). Bien que ces méthodes se soient avérées efficaces dans le cas bi-objectif du problème, elles utilisent la méthode des contraintes ε pour traiter l'aspect multiobjectif du problème, et pourraient donc devenir peu pratiques si plus de deux objectifs sont considérés.

Combinaison d'approches déterministes et métaheuristiques

Dans la deuxième partie de cette thèse, l'objectif scientifique est de développer une technique de résolution adaptée au problème de la conception de chaînes d'approvisionnement en hydrogène. Tout d'abord, les conclusions obtenues dans la première partie sur le traitement des contraintes sont exploitées pour proposer une première stratégie utilisant les MOEA durable. Puis, une nouvelle stratégie utilisant une matheuristique est développée, c'est-à-dire une méthode d'optimisation hybride qui combine des éléments des méthodes exactes et stochastiques de manière coopérative parencitejourdan2009hybridizing, ball2011heuristics. Plus précisément, cette stratégie de résolution est basée sur une décomposition bi-niveaux, de sorte que des méthodes appropriées peuvent être appliquées aux niveaux de décision concernés. Le problème du niveau supérieur concerne l'installation d'équipements de production et de stockage (c'est-à-dire un problème combinatoire multiobjectif), tandis que le niveau inférieur se concentre sur le problème associé au programme de transport et taux de production (problème de programmation linéaire de faible complexité). La solution du problème principal est effectuée par un MOEA, tirant ainsi pleinement parti de son potentiel pour traiter les caractéristiques multi-objectifs et combinatoires du problème. Ensuite, pour chaque solution partielle proposée par l'algorithme évolutionnaire, le problème esclave est traité par un solveur de program-

mation linéaire. De cette façon, le problème multiobjectif bi-niveaux est résolu de manière itérative, en une seule exécution.

L'intérêt d'une telle stratégie est qu'elle permet d'explorer une formulation plus réaliste du problème, qui intègre alors des relations algébriques entre le dimensionnement et le coût des installations de production et de stockage. En conséquence, le modèle implique des termes non linéaires puisque les coûts d'installation obéissent généralement à la règle classique de puissance des « six dixième » pour la mise à l'échelle des coûts des équipements et des usines. Ainsi, bien que le modèle résultant prenne la forme d'un problème de type PNLM, la méthodologie hybride proposée peut toujours être utilisée pour résoudre efficacement ce problème, car tous les termes non linéaires doivent être traités par le MOEA au niveau supérieur, tandis que le niveau inférieur restera un problème PL. Enfin, cette méthodologie est capable de fournir des détails englobant simultanément les aspects spatiaux, temporels et technologiques relatif au déploiement de la chaîne d'approvisionnement de l'hydrogène.

Objectif scientifique

La motivation de cette thèse est d'explorer l'applicabilité des avancées récentes en Calcul Évolutionnaire dans le domaine de l'ingénierie des systèmes de procédés. Plus précisément, ces avancées récentes comprennent (1) des techniques sophistiquées de gestion des contraintes, différentes des fonctions de pénalité ou des règles de faisabilité, et (2) des MOEA qui sont construits sur des paradigmes différents du principe de dominance de Pareto, en particulier : algorithmes travaillant par décomposition ou basé sur des indicateurs de performance. De cette façon, des méthodes de solution efficaces alternatives peuvent être identifiées pour aborder les problèmes d'optimisation complexes en PSI. En outre, une étude de cas d'intérêt actuel est examinée, à savoir la conception multiobjectif des chaînes d'approvisionnement en hydrogène (HSC), par le biais de stratégies de pointe qui utilisent à la fois les MOEA et les techniques MP.

Par conséquent, compte tenu du contexte présenté précédemment, quatre questions de recherche se posent :

1. Quelle est l'approche la plus adaptée pour résoudre les problèmes actuels en PSI, généralement formulés comme des problèmes d'optimisation multi-objectif ?

2. Compte tenu des avancées récentes dans le domaine des métaheuristiques, en particulier celles liées au traitement des contraintes, ces techniques sont-elles efficaces pour traiter des problèmes fortement contraints comme ceux liés au domaine des PSI ?

3. Une approche matheuristique peut-elle être conçue pour la résolution du problème de conception du HSC, de sorte qu'elle surpasse les méthodes exactes classiques et fournisse des solutions dans des temps de calcul raisonnables ?

4. Enfin, une approche matheuristique peut-elle être conçue comme un outil de solution général pour un large éventail de problèmes en PSI ?

Le travail présenté dans cette thèse vise à aborder ces questions de recherche tout au long de chaque chapitre. Une réponse explicite sera apportée à chacune d'elles dans la dernière section (conclusions et perspectives).

Structure de la thèse

Cette thèse est composée de deux parties principales comprenant sept chapitres. La première partie, qui comprend les trois premiers chapitres, est principalement consacrée à une étude empirique sur le traitement des contraintes pour les algorithmes d'optimisation multi-objectif. La deuxième partie (Chapitres 4 à 7), est consacrée à la conception optimale des chaînes d'approvisionnement en hydrogène.

En particulier, le Chapitre 1 présente une vue d'ensemble de l'optimisation multiobjectif ; il couvre certaines définitions et la notation formelle employée tout au long de cette thèse. Il présente également les techniques classiques de scalarisation généralement utilisées dans les approches de programmation mathématique, puis introduit les algorithmes évolutionnaires multiobjectifs. Les principaux composants communs à tout MOEA sont brièvement décrits, ainsi que certains opérateurs génétiques populaires, et les trois différents paradigmes de sélection dans les MOEA. Pour chaque "famille" d'algorithmes, un algorithme représentatif est présenté de manière plus détaillée : NSGA-II, MOEA/D et SMS-EMOA.

Le Chapitre 2 présente une étude comparative des techniques de traitement des contraintes dans les métaheuristiques, pour la résolution de problèmes d'optimisation en PSI, dans un cadre mono-objectif. L'étude prend en compte 5 techniques de traitement des contraintes et 14 problèmes PSE de petite et moyenne taille qui, malgré leur taille, posent des difficultés aux techniques classiques pour converger vers l'optimum global. Les résultats obtenus montrent la supériorité d'une méthode basée sur la réparation des solutions infaisables à l'aide du gradient des contrainte (jamais utilisé auparavant dans le domaine des PSI), en particulier lors du traitement de problèmes fortement contraints.

Le Chapitre 3, basé sur la même philosophie, présente des expériences similaires dans le contexte de l'optimisation multiobjectif. L'objectif est de vérifier que les résultats obtenus pour le cas mono-objectif sont valables pour les problèmes considérant plusieurs objectifs. À cette fin, la réparation par gradient est intégrée à six techniques de gestion des contraintes de l'état-de-l'art. Les performances respectives de ces différentes méthodes, dans leur version canonique et avec l'utilisation de la technique de réparation, inclues dans deux MOEA classiques (NSGA-II et MOEA/D), sont comparées. Les expériences ont été menées sur des banques de problèmes populaires tirées de la littérature de Calcul Évolutionnaire, ainsi que sur des problèmes récents contenant des contraintes égalité.

Le Chapitre 4 présente le contexte de la conception optimale des chaînes d'approvisionnement en hydrogène. Il fournit les caractéristiques essentielles de ce cas d'étude important, en décrivant en détail chaque échelon de la chaîne d'approvisionnement. Il constitue ainsi la base des chapitres suivants.

Le Chapitre 5 étudie l'applicabilité des MOEA pour la conception optimale des chaînes d'approvisionnement en hydrogène. Les résultats obtenus en utilisant l'algorithme MOEA/D avec la méthode de réparation par gradient sont présentés et analysés. La nécessité d'une technique de résolution plus adaptée aux caractéristiques du problème est mise en évidence, ce qui conduit à l'utilisation d'une stratégie hybride.

Le Chapitre 6 est consacré à la présentation d'une nouvelle matheuristique basée sur l'optimisation bi-niveaux pour la conception multiobjectif des chaînes d'approvisionnement en hydrogène. Ce chapitre présente également les expériences numériques réalisées pour la validation de cette stratégie. Dans un premier temps,

trois instances de petite à moyenne taille et à période unique ont été résolues en utilisant MOEA/D au niveau supérieur. Puis, six instances de plus grande taille ont été étudiées, en utilisant un algorithme plus approprié pour les problèmes bi-objectifs, SMS-EMOA.

Le Chapitre 7 est consacré à la présentation d'une formulation de la HSC qui prend en compte les détails spatiaux, temporels et technologiques. Les modifications apportées au modèle PLM de base sont expliquées dans une première partie, avec un accent particulier sur l'impact sur l'approche de la résolution. La deuxième partie de ce chapitre est consacrée à la présentation des résultats numériques et à leur discussion.

Enfin, des conclusions générales de cette thèse sont tirées, ainsi que quelques perspectives pour des travaux futurs. L'annexe A présente la formulation mathématique des problèmes étudiés dans le premier chapitre, ainsi que les informations concernant les solutions optimales locales et globales. L'annexe B contient la nomenclature relative à l'étude de cas HSC, ainsi que les données des différentes instances.

La recherche doctorale présentée dans ce manuscrit a été menée au Laboratoire de Génie Chimique, UMR 5503 CNRS/INP/UPS dans le département Procédés et Systèmes Industriels de mars 2018 à juillet 2021. Aussi, il convient de souligner que les travaux présentés dans cette thèse ont été présentés à la communauté scientifique, notamment par le biais de deux articles de recherche dans des revues internationales, de deux articles de conférence et d'un chapitre de livre :

- Victor H Cantú, Catherine Azzaro-Pantel, and Antonin Ponsich (2021). "Constraint-handling techniques within differential evolution for solving process engineering problems". In: *Applied Soft Computing* 108, p. 107442. DOI: https://doi.org/10.1016/j.asoc.2021.107442.

- Victor H Cantú, Catherine Azzaro-Pantel, and Antonin Ponsich (accepted). "A novel matheuristic based on bi-level optimization for the multi-objective design of hydrogen supply chains". In: *Computers & Chemical Engineering*, p. 107370. DOI: https://doi.org/10.1016/j.compchemeng.2021.107370.

- Victor H Cantú, Catherine Azzaro-Pantel, and Antonin Ponsich (2020a). "Multi-objective evolutionary algorithm based on decomposition (MOEA/D) for optimal design of hydrogen supply chains". In: *Computer Aided Chemical Engineering*. Ed. by Sauro Pierucci, Flavio Manenti, Giulia Luisa Bozzano, and Davide Manca. Vol. 48. Elsevier, pp. 883–888. DOI: https://doi.org/10.1016/B978-0-12-823377-1.50148-8.

- Victor H Cantú, Catherine Azzaro-Pantel, and Antonin Ponsich (2020b). "Optimal design of hydrogen supply chains by a multiobjective evolutionary algorithm based on decomposition (MOEA/D)". in: *EasyChair ROADEF2021*.

- Victor H Cantú, Antonin Ponsich, and Catherine Azzaro-Pantel (Apr. 2021). "Constraint Handling in Metaheuristics and Applications". In: ed. by Anand J Kulkarni, Efrén Mezura-Montes, Yong Wang, Amir H Gandomi, and Ganesh Krishnasamy. 1st ed. Springer Singapore. Chap. On the use of gradient-based repair method for solving constrained multiobjective optimization problems – A comparative study, pp. 119–149. DOI: 10.1007/978-981-33-6710-4.

Part I

An empirical study on constraint-handling with multiple objectives

Chapter 1

General overview on multiobjective optimization

Contents

1.1 Introduction

This chapter provides the background related to multiobjective optimization and the formal notation that will be used in the manuscript. First, Section 1.2 presents general definitions covering multiobjective optimization problems, optimality definitions, reference points and performance indicators. Section 1.3 introduces some classical scalarizing techniques, mainly used with mathematical programming approaches, namely aggregation functions and ε-constraint method. Section 1.4 gives a general overview on multiobjective evolutionary algorithms (MOEAs), outlining the main components common to any MOEA as well as the three different paradigms they are constructed on: Pareto dominance, decomposition and performance indicators. In addition, the most representative algorithm to each of the three paradigms is presented in more details.

Note that this chapter focuses only on *a posteriori* solution methods, i.e., methods that do not require *a priori* information on the decision makers' preferences. Indeed, in real-world engineering problems, the decision-makers usually do not have *a priori* knowledge about efficient solutions and, therefore, methods that can provide a set of alternative choices are more interesting. For a review on *a priori* and *interactive* approaches, the reader is referred to Miettinen (2012).

1.2 Basic concepts

The formulation of many real-world engineering problems involves the simultaneous optimization of several performance criteria (objectives), frequently with a set of constraints that need to be fulfilled. Typically, there is conflict between objectives, which means that improving one criterion results in the deterioration of another, and hence it does not exist a single solution that is optimal with respect to every objective function. These problems can be stated mathematically as constrained multiobjective optimization problems.

1.2.1 Problem definition

We study a constrained multiobjective optimization problem (CMOP) of the form:

$$\begin{aligned} \text{minimize} \quad & \mathbf{f}(\mathbf{x}) \\ \text{subject to} \quad & g_i(\mathbf{x}) \leq 0, \quad i = 1, \ldots, p \\ & h_j(\mathbf{x}) = 0, \quad j = 1, \ldots, q \\ & \mathrm{l}_i \leq x_i \leq \mathrm{u}_i, \quad i = 1, \ldots, n. \end{aligned} \tag{1.1}$$

where $\mathbf{x} = [x_1, x_2, \ldots, x_n]^T$ is a n-dimensional vector of decision variables (either discrete or continuous), $\mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_k(\mathbf{x})]^T$ is a k-dimensional vector of conflicting objective functions to be minimized[1], p is the number of inequality constraints and q is the number of equality constraints. The functions g_i and h_j may be linear or nonlinear, continuous or not, real-valued functions. Each variable x_i has upper and lower bounds, u_i and l_i, respectively, which define the search space $\mathcal{S} \subseteq \mathbb{R}^n$. Inequality and equality constraints, as well as the decision variables' bounds, define the feasible region $\mathcal{F} \subseteq \mathcal{S}$.

The CMOP's evaluation function, $\mathbf{f} : \mathcal{S} \rightarrow \mathcal{Z}$, maps decision variables ($\mathbf{x} = x_1, x_2, \ldots, x_n$) to vectors ($\mathbf{f} = f_1, f_2, \ldots, f_k$), that is, each solution in the decision variable space is associated to a point in the objective space, according to the objective function. This is illustrated in Figure 1.1 for the case of two decision variables $n = 2$ and three objective functions $k = 3$ where a disconnected feasible region is represented in both decision and objective spaces.

In the context of constrained optimization, the *overall constraint violation* is useful to evaluate the degree of infeasibility of a solution $\mathbf{x}$. It is defined as:

$$\phi(\mathbf{x}) = \sum_{i=1}^{p} \max\{0, g_i(\mathbf{x})\}^{\alpha} + \sum_{j=1}^{q} \left|h_j(\mathbf{x})\right|^{\alpha}. \tag{1.2}$$

where α takes a positive value, in this work $\alpha = 1$. Note that equality constraints can be transformed into inequality constraints as: $\forall j \in \{1, \ldots, q\}, \left|h_j(\mathbf{x})\right| - \epsilon \leq 0$, where ϵ is a small tolerance, typically $\epsilon = 1\mathrm{e} - 4$.

1.2.2 Optimality concepts

In multiobjective optimization, in order to compare two feasible solutions $\mathbf{x}, \mathbf{y} \in \mathcal{F}$, the *Pareto dominance relation* must be defined: A solution $\mathbf{x}$ is said to *dominate* a

[1] For the case of maximization, a negative sign is just added to the objective function

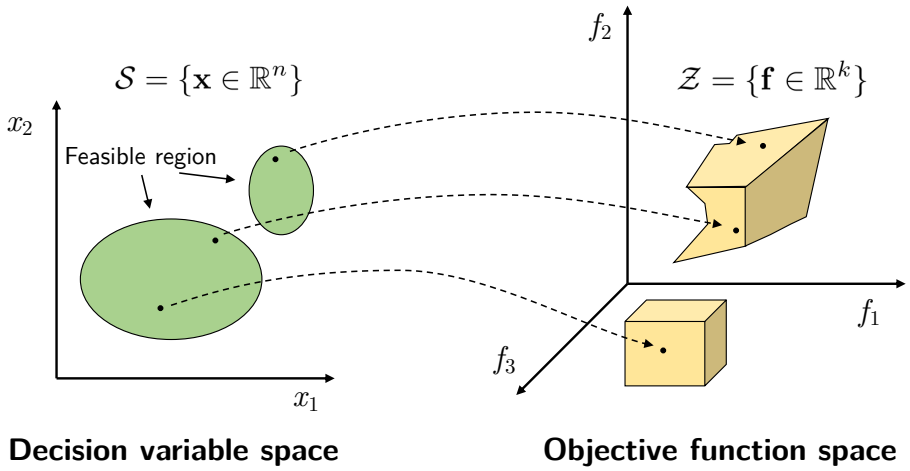


Figure 1.1: Multiobjective problem mapping.

solution $\mathbf{y}$, denoted by $\mathbf{x} \prec \mathbf{y}$, if and only if $\mathbf{x}$ is at least as good as $\mathbf{y}$ for all objectives $(\forall i \in \{1,\ldots,k\}, f_i(\mathbf{x}) \leq f_i(\mathbf{y}))$ and better for at least one objective $(\exists j \in \{1,\ldots,k\}, f_j(\mathbf{x}) < f_j(\mathbf{y}))$.

The set of all incomparable solutions, also known as efficient or non-dominated solutions, constitutes the *Pareto Optimal Set* $(\mathcal{PS})$, and is formally defined as:

$$\mathcal{PS} = \{\mathbf{x}^* \in \mathcal{F} : \nexists \mathbf{x} \in \mathcal{F}, \mathbf{x} \prec \mathbf{x}^*\}. \tag{1.3}$$

The mapping of these non-dominated vectors in the objective space is collectively known as the *Pareto Optimal Front* $(\mathcal{PF})$ or *True Pareto Front* (true PF), expressed as:

$$\mathcal{PF} = \{\mathbf{f}(\mathbf{x}^*) : \mathbf{x}^* \in \mathcal{PS}\}. \tag{1.4}$$

Throughout this work, we use the term true PF interchangeably with $\mathcal{PF}$. Further, since the $\mathcal{PS}$ may contain an infinite number of solutions, the multiobjective problem is limited to determine a finite number of Pareto optimal solutions that represents a good approximation of the $\mathcal{PF}$ in terms of both convergence and diversity, so that decision makers can select a solution according to their experience and preferences.

1.2.3 Reference points

The *ideal objective vector* $\mathbf{z}^* \in \mathbb{R}^k$, also known as utopian point, is obtained by minimizing each objective function individually subject to the constraints, in this manner, each component of the ideal vector can be represented as $z_i^* = \min\{f_i(\mathbf{x}) : \mathbf{x} \in \mathcal{F}\}$. The *nadir objective vector* $\mathbf{z}^{\text{nad}} \in \mathbb{R}^k$ constitutes an upper bound of the Pareto optimal front. It is defined in the objective space as $z_i^{\text{nad}} = \max\{f_i(\mathbf{x}) : \mathbf{x} \in \mathcal{PS}\}$. It is used, along with the ideal point, for performing an objective normalization in some scalarizing functions.

1.2.4 Performance indicators

With the purpose of evaluating the quality of an approximation to the Pareto optimal front, several performance indicators (or metrics) have been proposed in the literature.

These indicators aim to measure desirable aspects of non-dominated sets (Zitzler et al., 2000) and their corresponding front, namely: (1) the distance of the obtained approximation front to the Pareto optimal front, which is to be minimized, (2) the distribution of the solutions found in the objective space and (3) the extent of the non-dominated front in the objective space, which should contain a wide range of values for each objective.

Several performance metrics exist, for example, the hypervolume, the inverted generational distance, the R2 indicator, among many others (for a review see Coello Coello et al., 2007, p. 254-266). Later in this work, two popular metrics are to be employed for carrying out algorithm performance comparisons, these are the hypervolume and the inverted generational distance. They are described in the following.

Hypervolume

The hypervolume (HV), also known as $\mathcal{S}$ metric, is an indicator that reflects the volume of objective space enclosed by a Pareto front approximation A and a reference point $\mathbf{z}$ dominated by all elements of A (Zitzler and Thiele, 1998). Its mathematical representation is described in the following equation:

$$HV(A;\mathbf{z}) = \mathcal{L}\left(\bigcup_{a\in A}\{\mathbf{x} : \mathbf{a} \prec \mathbf{x} \prec \mathbf{z}\}\right) \tag{1.5}$$

where $\mathcal{L}$ is the Lebesgue measure (k-dimensional volume) and $\mathbf{z}$ should be dominated by all elements on A. This performance indicator has the remarkable property that its maximum value is yielded only by the optimal Pareto set. It is the only known indicator to have this "Pareto-compliance" property. A large HV value shows that a given solution set approximates the Pareto optimal front well in terms of both convergence and diversity in the objective space. A graphical representation of the hypervolume in two dimensions (area), for a classical benchmark problem, is provided in Figure 1.2. Set A in Figure 1.2 does not exhibit convergence nor diversity, and its hypervolume value is the lowest of the three sets. In set B, all solutions achieve good convergence, but a cluster of solutions is observed in the middle part of the front, evidencing a poor diversity. Finally in set C, solutions show a good convergence and are well distributed along the Pareto front, so that the highest hypervolume value is obtained among the three sets.

The main drawback of using the hypervolume is its high computational cost, which grows exponentially with the number of objectives (Knowles and Corne, 2002).

Inverted generational distance

The inverted generational distance (IGD) indicates how far the approximated Pareto front A is from the discretized Pareto optimal front R, i.e., it is the average distance from each true Pareto point to its nearest solution in the given approximated front (Van Veldhuizen and Lamont, 2000). It is mathematically expressed as:

$$IGD(A;R) = \frac{\left(\sum_{i=1}^{|R|} d_i^2\right)}{|R|} \tag{1.6}$$

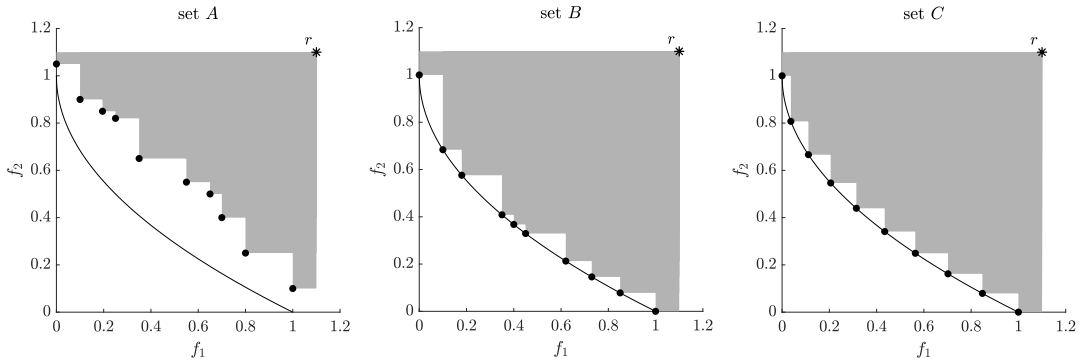


Figure 1.2: Hypervolume representation for the ZDT1 minimization problem. HV(A) = 0.5808; HV(B) = 0.8101; HV(C) = 0.8237.

where $|R|$ is the number of solutions in the Pareto optimal front R and d_i is the Euclidean distance in the objective space between a vector i in R to the nearest member of A. This indicator measures both convergence and diversity. A smaller value of IGD indicates a better performance of the algorithm.

1.3 Scalarizing techniques

Typically, in the PSE literature, optimization problems of the form in (1.1) are addressed by scalarizing techniques. This means that the problem is transformed into a set of scalar (single-objective) problems, for instance, by combining objective functions into a new objective function or by reformulating some objectives as constraints. In this way, classical algorithms for the solution of scalar optimization problems can be used. Once the transformation has been performed, the resulting problem is solved several times, using different parameters, with the aim of obtaining different efficient solutions at each execution. Depending on the scalarizing technique and the problem under study, solutions along the whole Pareto front can be found. However, this procedure can become computationally expensive for the accurate solution of large-size problems, or for problems involving more than two objectives. In addition, the parametrization chosen for each scalar optimization is not trivial for obtaining well-distributed approximated Pareto fronts. In the following, some classical scalarizing techniques found in the PSE literature are presented, namely utility functions and the ε-constraint approach.

1.3.1 Utility functions

A utility function u, also known as aggregation function or scalarizing function, transforms the original CMOP into a scalar optimization problem, through a weight vector $\mathbf{w}$ whose elements represent the preference or priority corresponding to each objective and that serves as a target direction in the objective space. Problem (1.1) then becomes:

$$\begin{aligned} &\text{minimize} && u(\mathbf{f}'(\mathbf{x}); \mathbf{w}) \\ &\text{subject to} && \mathbf{x} \in \mathcal{F} \end{aligned} \tag{1.7}$$

where $\mathbf{w}$ must satisfy $w_i \geq 0$ for all $i = 1, \ldots, k$ and, usually, $\sum_{i=1}^{k} w_i = 1$. In general, it is recommended to perform an objective normalization, so that all objective values have the same magnitude order. This normalization can take the form of:

$$\mathbf{f}'(\mathbf{x}) = \frac{\mathbf{f}(\mathbf{x}) - \mathbf{z}^*}{\mathbf{z}^{\text{nad}} - \mathbf{z}^*}. \tag{1.8}$$

where $\mathbf{z}^*, \mathbf{z}^{\text{nad}} \in \mathbb{R}^k$ are the ideal and nadir points, obtained from the single-objective optimization of each objective function. In the following, some characteristics of two popular utility functions are described. For a comprehensive study on utility functions, the reader is referred to Pescador-Rojas et al. (2017).

Weighted sum

The weighting sum is probably the simplest and most widely used method to transform a CMOP into a scalar problem. A weight is assigned to each objective function and minimizes the corresponding weighted sum of the objectives. The problem takes the form of:

$$u^{ws}(\mathbf{f}'; \mathbf{w}) = \sum_{i=1}^{k} w_i f_i'(\mathbf{x}) \tag{1.9}$$

In order to obtain Pareto optimal solutions, the weight vectors w_i must take only positive values. The main drawback of this method is that not all Pareto optimal points can be found if the true PF is concave (convex in the case of maximization). Besides, a uniformly distributed set of weighting vectors does not necessarily produce a uniformly distributed representation of the Pareto optimal set (Das and Dennis, 1998).

Weighted Tchebycheff

Also known as weighted min-max function, the weighted Tchebycheff function (TCH) seeks to minimize the weighted norm-1 distance between the objective vector and the ideal point. It can be written as:

$$u^{tch}(\mathbf{f}'(\mathbf{x}); \mathbf{w}) = \max_{i=1,\ldots,k} \{w_i |f_i'(\mathbf{x})|\} \tag{1.10}$$

This method can provide solutions along the Pareto optimal front with appropriate variations of the weight vector. However, in some cases, it can produce solutions that are only weakly Pareto optimal (i.e., dominated solutions). To address this drawback, the *augmented Tchebycheff* function (ATCH) was proposed (Kaliszewski, 1987), which always produces Pareto optimal solutions. It is expressed as follows:

$$u^{atch}(\mathbf{f}'(\mathbf{x}); \mathbf{w}) = \max_{i=1,\ldots,k} \{w_i |f_i'(\mathbf{x})|\} + \alpha \sum_{i=1}^{k} w_i |f_i'(\mathbf{x})| \tag{1.11}$$

where α must take small values, for instance 0.001 as proposed in (Pescador-Rojas et al., 2017).

Achievement scalarizing

The achievement scalarizing function (ASF) (ibid.) can produce weakly Pareto optimal solutions and is written as:

$$u^{asf}(\mathbf{f}'(\mathbf{x}); \mathbf{w}) = \max_{i=1,\dots,k} \left\{ \frac{f_i'(\mathbf{x})}{w_i} \right\} \tag{1.12}$$

Despite its similarity to the weighted Tchebycheff function, recent studies indicate that ASF outperforms TCH in problems involving more than three objectives, as ASF can find an objective vector parallel to $\mathbf{w}$ (Hernández Gómez and Coello Coello, 2015). Moreover, in order to remove weakly Pareto optimal solutions, the *augmented achievement scalarizing function* (AASF) is defined as:

$$u^{aasf}(\mathbf{f}'(\mathbf{x}); \mathbf{w}) = \max_{i=1,\dots,k} \left\{ \frac{f_i'(\mathbf{x})}{w_i} \right\} + \alpha \sum_{i=1}^{k} \frac{f_i'(\mathbf{x})}{w_i} \tag{1.13}$$

where α must take small values, for instance 0.001 as proposed in (Pescador-Rojas et al., 2017).

1.3.2 ε-constraint method

In the ε-constraint method, the multiobjective problem is transformed into several single-objective optimization problems with additional constraints, i.e., one of the objective functions is selected to be optimized at a time, while the others are treated as constraints by setting an upper bound to each of them. The problem to be solved takes the form of:

$$\begin{aligned} &\text{minimize} && f_\ell(\mathbf{x}), \\ &\text{subject to} && f_j(\mathbf{x}) \leq \varepsilon_j \quad \forall j = 1, \dots, k, \quad j \neq \ell, \\ & && \mathbf{x} \in \mathcal{F} \end{aligned} \tag{1.14}$$

where $\ell \in \{1, \dots, k\}$.

It is important to note that, in order to obtain appropriate values for ε_j, a preliminary analysis needs to be made, usually by carrying out a single-objective optimization for each objective function. Besides, by varying the ε_j values appropriately, non-inferior solutions of the problem can be obtained. However, this may yield also some weakly dominated solutions, because the ε constraints are not necessarily active at the solution found. Some improved versions of this method indeed tackle this problem by forcing the ε constraints to be saturated (Mavrotas, 2009; Mavrotas and Florios, 2013). Again, the main disadvantage of the method is that it can become computationally expensive for solving large-scale problems, or problems with more than two objectives.

1.4 Evolutionary multiobjective optimization

Multiobjective evolutionary algorithms (MOEAs) are bio-inspired search algorithms that mimic natural evolutionary processes, such as selection, crossover and mutation, to steer a set of solutions (the so-called population) towards optimal or near-optimal

solutions (Bäck, 1996; Emmerich and Deutz, 2018). Due to their population-based working mode, they have the ability to provide an approximation of the Pareto set in one single run and do not require particular mathematical properties of the problem, e.g., continuity or convexity. The common features to all MOEAs are:

1. A population of individuals (solutions), updated at each generation (i.e., iteration) t and denoted as $P_t = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(\mu)}\}$, where $\mathbf{x}^{(i)} \in \mathcal{S}$ represents and individual and μ is the population size (operating parameter).

2. The parent selection. It consists in choosing the individuals that will participate in the reproduction (combination) of the members of the current generation (parents), in order to produce new ones (offspring). Typically, this is done by a biased random sampling or tournament selection, but is sometimes dictated by the variation operator.

3. The variation operator. This operator stochastically generates new individuals from the given parents, through the crossover and mutation processes. In the evolutionary metaphor context, this operator corresponds to the sexual reproduction and serves to combine the genetic information of the parents. Depending on the solution representation (encoding), different variation operators exist. Examples of variation operators for binary encoding are $1/2/k$-point crossover and binary mutation; whereas for real encoding, some popular operators are simulated binary crossover (SBX) with polynomial mutation, or the so-called differential evolution mutation operator, among others.

4. The environmental selection (survival). It specifies which individuals will survive to the next generation and are likely to be used as parents. Generally, the *best* individuals among the parents and offspring are chosen (i.e., parents and offspring populations compete for their survival to the next generation), which is referred to as *elitism*. It is worth mentioning that in this step, different selection paradigms exist in the framework of multiobjective optimization: Pareto dominance, decomposition and performance metric (see more details in Section 1.4.2).

In addition, some MOEAs use an external archive that acts as a secondary population across generations. This archive stores the non-dominated solutions found so far and usually has a user-defined size limit.

In the following, two classical variation operators are described in more details. A representative algorithm of each selection paradigm is also presented along with its main characteristics and pseudocode, these are, NSGA-II (Deb et al., 2002), MOEA/D (Zhang and Li, 2007) and SMS-EMOA (Emmerich et al., 2005).

1.4.1 Variation operators

Simulated Binary Crossover (SBX)

This crossover technique was proposed in (Deb and Agrawal, 1995) to overcome some drawbacks observed in binary-coded genetic algorithms when used for continuous optimization problems, such as the necessity to define in advance the desired precision on the final solution, the computational burden associated to solution encoding, or the so-called Hamming drift problem, i.e., slight differences in the genotype (encoded

solution vector) may result in large differences in the phenotype (decoded decision variables), and vice versa. In SBX, two parents $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$ generate two offspring $\mathbf{y}^{(1)}$ and $\mathbf{y}^{(2)}$ according to:

$$\begin{aligned} y_i^{(1)} &= 0.5\left((1-\beta_i)x_i^{(1)} + (1+\beta_i)x_i^{(2)}\right) \\ y_i^{(2)} &= 0.5\left((1+\beta_i)x_i^{(1)} + (1-\beta_i)x_i^{(2)}\right) \end{aligned} \tag{1.15}$$

where $i \in \{1, \ldots, n\}$, that is, the procedure is done for each decision variable, with a probability p_c. The parameter β_i is calculated as:

$$\beta_i = \begin{cases} (2u)^{\frac{1}{\eta_c+1}}, & \text{if } u \leq 0.5 \\ \left(\frac{1}{2(1-u)}\right)^{\frac{1}{\eta_c+1}}, & \text{otherwise} \end{cases} \tag{1.16}$$

where u is a random number between 0 and 1, and η_c is the distribution index, which is a parameter set by users.

The particular probability distribution β was designed in order that SBX performs similarly to single-point crossover in binary-coded genetic algorithms. It takes the form of:

$$p(\beta) = \begin{cases} 0.5(\eta_c+1)\beta^{\eta_c}, & \text{if } 0 \leq \beta \leq 1 \\ 0.5(\eta_c+1)\beta^{-\eta_c-2}, & \text{if } \beta > 1 \end{cases} \tag{1.17}$$

where η_c has a direct effect in the spread of child solutions (in the search space): if η_c is large, there is a higher probability that the offspring are close to the parent solutions, on the contrary, if η_c takes small values, solutions distant from the parents are likely to be generated. In this work, SBX operator is employed with standard settings: $p_c = 1, \eta_c = 20$.

Regarding the encoding of discrete variables, the authors of (ibid.) recommended the use of an adapted operator for the representation of those variables only. However, in practice, SBX is usually extended for handling discrete variables in a simple manner: discrete variables are encoded as continuous ones and their value is rounded to the closest integer only in the evaluation module (and *not* in the solution encoding).

Besides, in order to maintain diversity in the population, a mutation operator can be used. This is usually achieved by adding a perturbation to the current value of a variable, following a predefined polynomial probability distribution.

Differential evolution

Differential evolution (DE), is a relatively recent evolutionary algorithm (EA) proposed by Storn and Price (1997). The key point of this EA is a variation operator that performs differences between individuals' decision vectors for generating new solutions. Although using at least three individuals to produce one offspring, this variation operator is generally referred to as a mutation operator. More precisely, a mutant individual is produced by adding a scaled difference vector to another individual (base vector). In the classical version ($rand/1/bin$), this operation can be written as:

$$\mathbf{v}^{(i)} = \mathbf{x}^{(r_1)} + F \cdot (\mathbf{x}^{(r_2)} - \mathbf{x}^{(r_3)}) \tag{1.18}$$

where $r_1 \neq r_2 \neq r_3 \neq i \in \{1, \ldots, \mu\}$ are indexes randomly chosen among the population, and F is a scaling parameter. The mutant vector $\mathbf{v}^{(i)}$ is then recombined with its parent with a crossover probability CR. This procedure produces an offspring $\mathbf{y}^{(i)}$ according to:

$$y_j^{(i)} = \begin{cases} v_j^{(i)} & \text{if } u \sim U(0,1) \leq CR \text{ or } j = j_r, \\ x_j^{(i)} & \text{otherwise} \end{cases} \tag{1.19}$$

where $j \in \{1, \ldots, n\}$ and j_r is a randomly chosen index $\in \{1, \ldots, n\}$ ensuring that $\mathbf{y}^{(i)}$ gets at least one variable from the mutant $\mathbf{v}^{(i)}$.

Depending on the mutation and crossover schemes considered, several variants of this mutation operator exist. They are usually denoted as $DE/a/b/c$, where a specifies the base vector which can be *rand* (randomly chosen) or *best* (the vector with better fitness from the current population), b denotes the number of difference vectors used, and c denotes the crossover scheme that can be *bin* (independent binomial experiments) or *exp* (exponential crossover, similar to single-point crossover in GA). In this work, the most popular version of DE is used, that is, $DE/rand/1/bin$, which considers one difference between individuals randomly chosen from the population and performs binomial crossover, as represented in (1.18) and (1.19).

Similar to the SBX operator, discrete variables in DE are encoded in a straightforward way: they are rounded to the next integer, only for evaluation purposes.

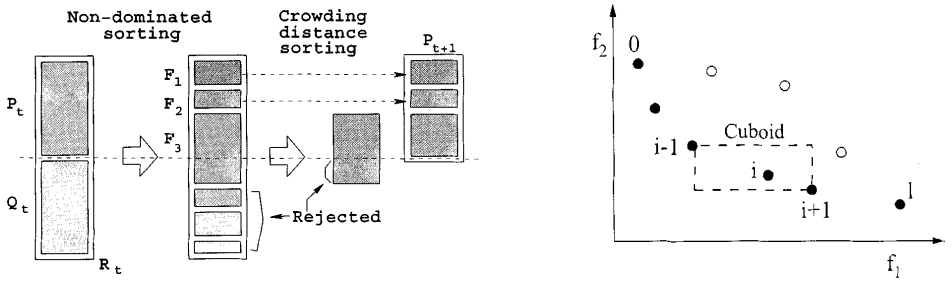
1.4.2 Selection paradigms

Pareto-based algorithms: NSGA-II

This family of algorithms uses the Pareto dominance relationship to differentiate individuals in the population. The population is ranked according to this criterion and, in general, use the contribution to diversity as the second selection criterion among solutions that are incomparable according to Pareto dominance. Examples of such algorithms are SPEA2 (Zitzler et al., 2001) and PAES (Knowles and Corne, 1999), while the most widely used is NSGA-II (Deb et al., 2002).

In NSGA-II, the fitness (rank) of an individual is assigned primarily according to the non-dominated front it belongs to, also known as the nondomination level, and is determined as follows: non-dominated individuals in some population P are assigned rank 1 and removed from P, then, non-dominated individuals in the new population (P without first-rank individuals) are assigned rank 2 and removed from it. This process is repeated until each solution has an assigned rank.

More generally, at each generation t, an offspring population Q_t of size μ is generated through classical variation operators, usually SBX with polynomial mutation. Then, the union of parents and offspring populations ($R_t = P_t \cup Q_t$), of size 2μ, is sorted in multiple fronts according to non-dominance. The μ best individuals, i.e., those belonging to the first ranks, are selected for surviving to the next generation and conform the next population P_{t+1} (see Figure 1.3a). At this respect, in order to exactly select μ individuals from $P_t \cup Q_t$, only a given number of solutions might be needed from a given rank. In this case, a measure of density of solutions in their neighborhood, the so-called crowding distance, is used as the selection criterion. It is given by the average distance in each dimension of the two nearest neighbors that surround a solution (as represented in Figure 1.3b). In this manner, those individuals



(a) Selection procedure.

(b) Crowding distance calculation.

Figure 1.3: Main aspects of NSGA-II. Taken from Deb et al., 2002.

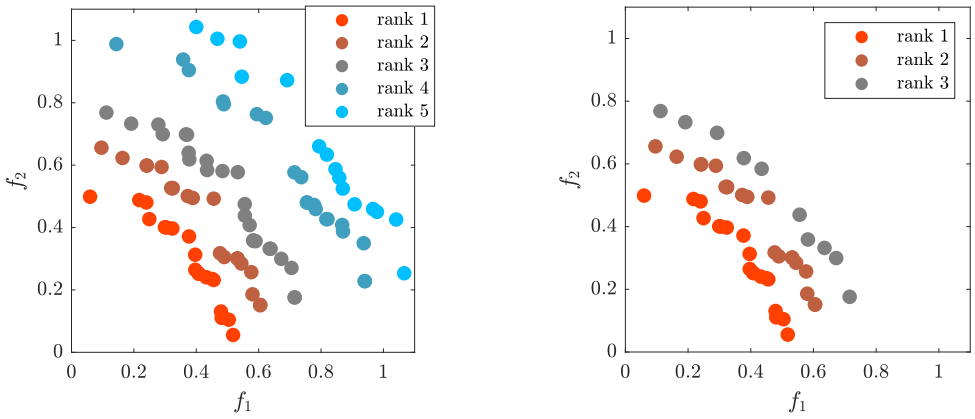
that most contribute to the uniform distribution of solutions in the objective space are selected. This process is depicted in Figure 1.3 and the corresponding pseudocode is given in Algorithm 1.

Algorithm 1 NSGA-II Algorithm

initialize population $P_0 \subset \mathcal{F}^\mu$
while not terminate **do**
  {Begin variate}
  $Q_t \leftarrow \emptyset$
  for all $i \in \{1, \ldots, \mu\}$ **do**
    $(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}) \leftarrow$ select_mates(P_t) {select parent individuals}
    $\mathbf{r}_t^{(i)} \leftarrow$ recombine$(\mathbf{x}^{(1)}, \mathbf{x}^{(2)})$
    $\mathbf{q}_t^{(i)} \leftarrow$ mutate$(\mathbf{r}_t^{(i)})$
    $Q_t \leftarrow Q_t \cup \{\mathbf{q}_t^{(i)}\}$
  end for
  {End variate}
  {Begin selection, select μ-"best" out of $(P_t \cup Q_t)$ by a two step procedure:}
  $(R_1, \ldots, R_\ell) \leftarrow$ non-dom_sort$(\mathbf{f}, P_t \cup Q_t)$ {Assign rank based on Pareto dominance}
  Find the element of the partition, R_{i_μ}, for which the sum of the cardinalities $|R_1| + \cdots + |R_{i_\mu}|$ is for the first time $\geq \mu$. If $|R_1| + \cdots + |R_{i_\mu}| = \mu, P_{t+1} \leftarrow \cup_{i=1}^{i_\mu} R_i$, otherwise determine set H containing $\mu - (|R_1| + \cdots + |R_{i_\mu - 1}|)$ elements from R_{i_μ} with the highest crowding distance and $P_{t+1} \leftarrow (\cup_{i=1}^{i_\mu - 1} R_i) \cup H$.
  {End selection}
  $t \leftarrow t + 1$
end while
return P_t

The typical selection carried out according to NSGA-II is shown in Figure 1.4, for a given generation, considering a minimization problem for both objectives and $\mu = 50$. In the left-hand side picture, the combined population R_t has 100 individuals, composed of parents and offspring. It contains 19 solutions in rank 1, which are all selected; 21 solutions in rank 2, also selected; and 25 solutions in rank 3 (the remaining solutions have higher ranks). Thus, only 10 ($= 50 - 19 - 21$) solutions are selected out of the 25 belonging to the third rank, according to the crowding distance criterion (see the right-hand side in Figure 1.4).

Pareto-based algorithms usually works well for problems containing up to three



(a) Ranked population conformed of parents and offspring.

(b) Best μ individuals selected to survive for the next generation.

Figure 1.4: Typical selection according to NSGA-II (minimization problem).

objectives and can effectively cope with discontinuities in the objective space. Their main concerns typically appears when treating many-objective problems (i.e., problems having four or more objectives) since a majority of individuals might be non-dominated and the search is then exclusively guided by the diversity criterion, without promoting convergence.

Decomposition-based algorithm: MOEA/D

MOEAs based on decomposition divide the problem into several scalar (single-objective) subproblems using scalarizing (or utility) functions. The emblematic algorithm MOEA/D is built on this paradigm (Zhang and Li, 2007). In MOEA/D, the original MOP is divided into a number of scalar optimization subproblems equal to the number of individuals in the population, that is, each individual searches in a specific region of the objective space, according to its assigned weight vector. Therefore, the weight vectors should be evenly distributed in the search space in order to produce an uniformly-distributed approximation of the Pareto front. Weight vector generation is typically done through the Simplex-Lattice method (Das and Dennis, 1998). Then, the optimization procedure is performed in a collaborative way: the population is divided into neighborhoods (according to the similarity of the weight vectors assigned to different individuals), and individuals in the same neighborhood may exchange information, i.e. solutions. In particular, when an offspring is generated for an individual, this new solution is shared with the neighboring individuals, which may accept it according to the corresponding fitness, computed as the utility function employed. Nevertheless, the number of neighbors that can pick a given solutions is limited, to avoid hampering the diversity of the population.

In its canonical form, MOEA/D uses the SBX crossover with polynomial mutation as variation operators, but recent versions employ the differential evolution operator, also combined with polynomial mutation, with significant improvements obtained with respect to SBX (Zhang et al., 2009). Besides, in order to avoid objective scaling issues, an objective normalization such as (1.8) is usually performed, considering the reference points found so far. Finally, the classical version of MOEA/D makes use of an external archive that stores the non-dominated solutions found throughout the

optimization process. The pseudocode is provided in Algorithm 2. It is important to highlight that MOEA/D offers advantages with respect to methods that independently solve the scalar subproblems (like those in mathematical programming literature), as the search is carried out in a cooperative manner (Emmerich and Deutz, 2018). Besides, the external archive may contain many more solutions than individuals in the initial population.

Algorithm 2 MOEA/D

1: input: n_r, δ {number of replacements, probability of choosing parents locally}
2: set: EP $= \emptyset$ {external archive}
3: input: $\Lambda = \{\lambda^{(1)}, \ldots, \lambda^{(\mu)}\}$ {weight vectors}
4: initialize neighborhoods $B^{(i)}$ by collecting k closest weight vectors in Λ for each $\lambda^{(i)}$
5: initialize population $P_0 \subset \mathcal{F}^\mu$
6: initialize $\mathbf{z}^*, \mathbf{z}^{\text{nad}}$ from P_0
7: **while** not terminate **do**
8: **for all** $i \in \{1, \ldots, \mu\}$ **do**
9: Set $c = 0$ {number of replacements effectuated}
10: {Begin variate}
11: $T \leftarrow$ mates_pool$(P_t, B^{(i)}, \delta)$ {choose of mating pool}
12: $(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \mathbf{x}^{(3)}) \leftarrow$ select_mates(T) {select parent individuals}
13: $\mathbf{y} \leftarrow$ recombine$(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \mathbf{x}^{(3)})$
14: $\mathbf{y}' \leftarrow$ mutate$(\mathbf{y})$
15: {End variate}
16: {Begin selection}
17: Randomly pick an index j from T.
18: **if** $u(\mathbf{f}'(\mathbf{y}'); \lambda^{(j)}) < u(\mathbf{f}'(\mathbf{x}^{(j)}); \lambda^{(j)})$ **and** $c < n_r$ **then**
19: $\mathbf{x}^{(j)} \leftarrow \mathbf{y}'$
20: $c \leftarrow c + 1$
21: $T \leftarrow T \setminus \{j\}$ and go to line 17 {delete j from T}
22: **end if**
23: {End selection}
24: Update $\mathbf{z}^*, \mathbf{z}^{\text{nad}}$.
25: **end for**
26: Update EP
27: $t \leftarrow t + 1$
28: **end while**
29: **return** P_t and EP

In Figure 1.5 some aspects of the decomposition-based approach are illustrated. To this end, a two-objective minimization problem is analyzed, whose Pareto optimal front involves both convex and concave shapes. In red and blue colors (and a mix of both), the weight vectors, which assign target directions for the scalar subproblems, are depicted. It can be noted that, even if the weight vectors are uniformly distributed in the objective space, the resulting search direction highly depends on the chosen scalarizing function. In this way, the weighted sum function cannot find (or search) solutions in concave regions of the PF, as shown in the left figure. On the contrary, the target directions defined by ATCH and AASF functions cover the whole PF and are well distributed. Therefore, efficient solutions (shown in circles) are found all along the PF.

Recently, decomposition-based MOEAs have become very popular mainly because they scale well to problems with many objective functions, see Pescador-Rojas et al.

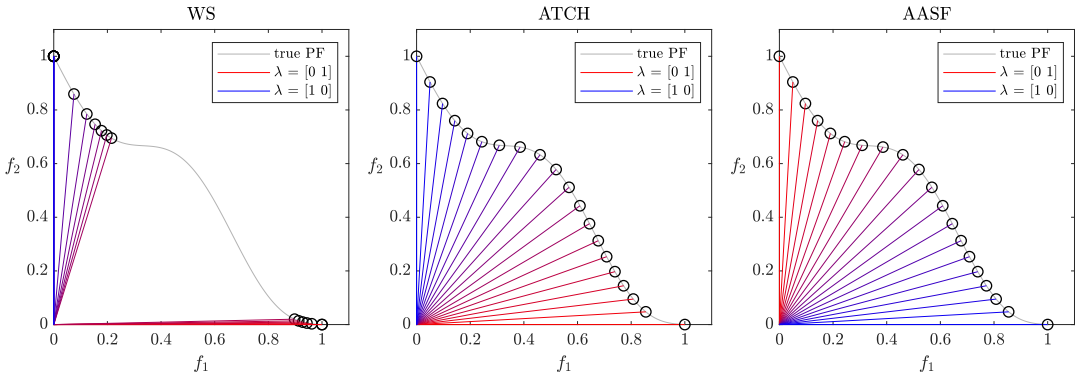


Figure 1.5: Target directions using different scalarizing functions for a given problem.

(2017). Another significant example of a decomposition-based algorithm is NSGA-III (Deb and Jain, 2013), which combines the Pareto-dominance scheme used in NSGA-II, but substitutes the crowding distance by a niching mechanism using weight vectors as search directions.

Indicator-based algorithms: SMS-EMOA

The third selection paradigm in MOEAs is based on optimizing performance indicators, which measure the quality of the Pareto front approximation (see Section 1.2.4 of this chapter), instead of optimizing directly the objectives. In particular, a relevant algorithm is SMS-EMOA (Emmerich et al., 2005) which seeks to maximize the only indicator known to be Pareto-compliant, the hypervolume indicator. SMS-EMOA is a steady-state greedy algorithm, meaning that only one individual is generated at each iteration and no decrease in the hypervolume covered by the current population is allowed. This implies that new offspring solutions can only integrate the current population if replacing a member increases the hypervolume covered by the population, or, more precisely, if the new offspring (1) dominates at least one individual in the current population, or (2) does not contribute the least to the hypervolume computation provided that it belongs to the last Pareto front. This is detailed in the following:

- Condition (1) is taken from the above-described NSGA-II algorithm, and is easy to grasp: among two solutions, the one that dominates the other is preferred at any point of the evolutionary process.

- Condition (2) introduces the contribution to the hypervolume Δ_S of each individual as the selection criterion for those individuals in the last front considered (according to the non-dominated sorting procedure in NSGA-II). That is, the individual that contributes the least to the hypervolume is discarded from the worst ranked front.

In a bi-objective case, the contribution to the hypervolume can be computed efficiently as the product of the difference of the objective values between two subsequent solutions, once the set has been sorted in ascending order according to the values of the first objective function f_1. Note that, since extreme points are to be maintained in the population, the contribution to the hypervolume is only computed for internal

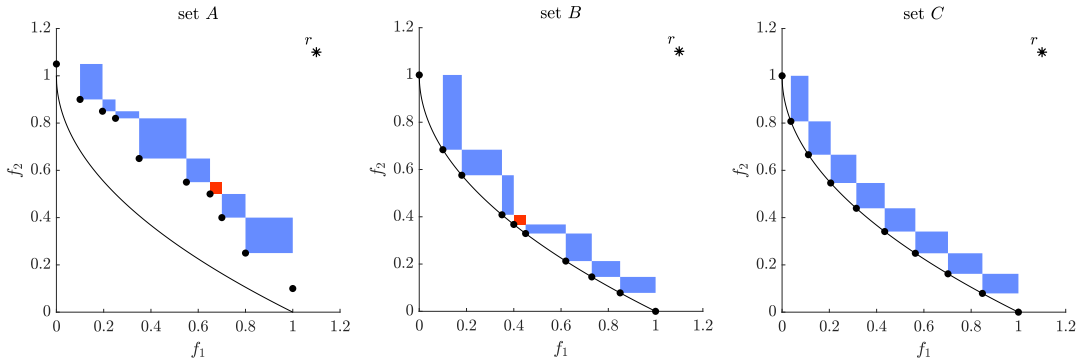


Figure 1.6: Representation of the contribution to the hypervolume for the ZDT1 problem.

solutions of the last non-dominated rank. Then, considering the last front $R_\ell = \{s_1, \ldots, s_{|R_\ell|}\}$ with $|R_\ell| > 2$, the $\Delta_{\mathcal{S}}$ of each internal solution s_i is computed as:

$$\Delta_{\mathcal{S}}(s_i, R_\ell) = (f_1(s_{i+1}) - f_1(s_i)) \cdot (f_2(s_{i-1}) - f_2(s_i)) \tag{1.20}$$

for all $n \in \{2, \ldots, |R_\ell| - 1\}$. In Figure 1.6 the contribution of the hypervolume of three approximation sets is displayed graphically for a classical test function, ZTD1 (Zitzler et al., 2000). The area of each rectangle corresponds to the $\Delta_{\mathcal{S}}$ of each candidate solution. The red rectangle in sets A and B represents the $\Delta_{\mathcal{S}}$ of the solution that contributes the least to the HV. Thus, the best possible approximation to the Pareto front for a given number of efficient solutions is a set in which each solution contributes in an equal extent to the hypervolume total value, such as set C in Figure 1.6.

The main drawback of SMS-EMOA is the high computational cost of $\Delta_{\mathcal{S}}$ for more than two objectives, which undermines its efficiency for many objectives or large sets (Knowles and Corne, 2002).

Algorithm 3 SMS-EMOA

initialize population $P_0 \subset \mathcal{F}^\mu$
while not terminate **do**
    {Begin variate}
    $(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}) \leftarrow$ select_mates(P_t) {select parent individuals}
    $\mathbf{c}_t \leftarrow$ recombine$(\mathbf{x}^{(1)}, \mathbf{x}^{(2)})$
    $\mathbf{q}_t \leftarrow$ mutate$(\mathbf{c}_t)$
    {End variate}
    {Begin selection, select μ-"best" out of $P_t \cup \{\mathbf{q}_t\}$ by a two step procedure:}
    $(R_1, \ldots, R_\ell) \leftarrow$ non-dom_sort$(\mathbf{f}, P_t \cup \mathbf{q}_t)$ {Assign rank based on Pareto dominance}
    $r \leftarrow \operatorname{argmin}_{s \in R_\ell} [\Delta_{\mathcal{S}}(s, R_\ell)]$ {Detect element of R_ℓ with lowest $\Delta_{\mathcal{S}}(s, R_\ell)$ }
    $P_{t+1} \leftarrow (P_t \cup \mathbf{q}_t) \setminus \{r\}$ {Eliminate detected element}
    {End selection}
    $t \leftarrow t + 1$
end while
return P_t

1.5 Conclusions

In this first chapter, the background related to multiobjective optimization has been presented. This general overview has covered methods typically employed in the PSE literature, i.e., scalarizing techniques that decompose the problem into a set of single-objective subproblems. Though efficient solutions can be obtained by these methods, in many cases, they can constitute computationally expensive techniques, for instance, if high accuracy of the Pareto front approximation is desired, or if many objectives are involved. Therefore, a particular attention has been paid to MOEAs, for which recent advances can entail potential benefits as solution techniques to PSE applications. The features common to all MOEAs have been described, namely, the set of individuals that conform the population at a given generation, the parent selection, the variation operators and the survival selection. As can be noted, a MOEA in its canonical form does not contain a constraint-handling mechanism, mainly because, originally, they have been designed to work in unconstrained search spaces. Actually, this feature (constraint-handling) is one of the most important for effectively addressing PSE optimization problems.

Therefore, taking the above observations into account, a preliminary study on constraint-handling needs to be carried out before implementing MOEAs to real-world PSE case studies. The next two chapters provide systematic studies on constraint-handling techniques, first in a single-objective framework, and then, for multiobjective optimization problems.

Chapter 2

Constraint-handling in a single-objective framework

Contents

2.1 Introduction

In this chapter, a comparison study on some state-of-the-art constraint-handling techniques in metaheuristics is carried out in a single-objective framework. As a reminder, the scientific objective of this PhD thesis is to explore alternative solution methods to complex optimization problems related to process systems engineering, which usually involve multiple conflicting objectives that need to be simultaneously optimized. Metaheuristics and, in particular, multiobjective evolutionary algorithms, are potential alternatives to exact methods, able to provide good-quality solutions in tractable computational times. Nevertheless, the main drawback they present is the inability to cope effectively with constraints, in part because they have been designed to work under unconstrained search spaces. Hence, multiple constraint-handling mechanisms have been proposed to be embedded within evolutionary algorithms, typically in a single-objective framework, and more recently some have been explored for multiobjective problems. Therefore, this chapter provides a comparison study on some state-of-the-art constraint-handling techniques, considering 14 single-objective

The content of this chapter has been published in the form of journal article. Victor H Cantú, Catherine Azzaro-Pantel, and Antonin Ponsich (2021). "Constraint-handling techniques within differential evolution for solving process engineering problems". In: *Applied Soft Computing* 108, p. 107442. DOI: https://doi.org/10.1016/j.asoc.2021.107442

problems related to PSE area. The techniques considered are: static penalty function, stochastic ranking, feasibility rules, ε-constraint method and gradient-based repair. The obtained results allow to get insights about the best constraint-handling techniques to be used later in multiobjective PSE problems.

The chapter is organized as follows. Section 2.2 presents a brief description of mathematical programming techniques and evolutionary algorithms, focusing on the main difficulties they present for the solution of complex PSE problems. Then, Section 2.3 discusses the constraint-handling techniques in metaheuristics; some state-of-the-art methods proposed in the dedicated literature are reviewed and are compared to classical techniques usually employed in PSE literature. Then, Section 2.4 includes a detailed description of some of the most representative constraint-handling techniques in EAs, which will be employed for the study. After that, the methodology developed for the computational experiments is presented along with some information regarding the test problems. The obtained results are presented and discussed in Section 2.6. The last section is devoted to the conclusions of this work.

2.2 Motivation

Complex optimization problems are ubiquitous in chemical engineering, and more generally, in process systems engineering (PSE). Examples of optimization problems related to this area encompass batch process design (Ponsich et al., 2007; Voudouris and Grossmann, 1992), phase equilibrium (Dowling and Biegler, 2015), distillation sequencing (Zhu et al., 2016), heat exchanger networks (Ayala et al., 2016; Yee and Grossmann, 1991), reactor network design (Kaiser et al., 2016), supply chain design (Almaraz et al., 2015; Woo et al., 2016), among others. All these real-world problems are typically represented by a mathematical model involving both discrete and continuous variables, and a set of linear and nonlinear constraints, i.e., leading to a mixed-integer nonlinear programming (MINLP) formulations.

Throughout the years, mathematical programming techniques have been used to address the solution of these problems. Several algorithms such as the Generalized Benders Decomposition (GBD), Outer-Approximation with the Equality-Relaxation strategy (OA/ER) and multiple Branch-and-Bound algorithms (BB) have been proposed to solve to global optimality these problems (in general with some user-defined tolerance over the optimality gap, i.e., the difference between the lower and upper bounds) (Floudas and Gounaris, 2009; Tawarmalani and Sahinidis, 2013). It should be noted, however, that all these Newton-based methods rely on the appropriate selection of an initial solution and on derivability and convexity assumptions regarding the nonlinear functions to guarantee the global optimum solution. In fact, their performance may depend strongly on the problem's formulation, meaning that a different algebraic form representing the same system might lead to quite different algorithmic performances (Liberti, 2008). That is, the problem needs to satisfy some specific mathematical characteristics (e.g., convexity, derivability) so that a valid reformulation can be generated automatically. Otherwise, the reformulated convex problem might miss the original global optimum and converge to a local optimum. For the convex relaxation to be performed, the problem has to be reduced to a *standard form*, in which nonlinear terms are linearized (by adding new variables and constraints), and then each non-convex term is replaced by a convex envelope obtained as the convex hull of the inequality constraints. Since one of the key factors

to obtain the global optimum consists in computing a tight convex lower bound, several methods have been proposed depending on the mathematical characteristics of the studied problems (e.g. McCormick's envelopes, piecewise-linear underestimators, etc.). Furthermore, the application of these deterministic techniques can be computationally expensive to obtain the rigorous global solution of large-scale problems.

In that context, metaheuristics, and especially evolutionary algorithms (EAs), have been proposed as alternative methods to address highly non-convex optimization problems. Some representative examples of EAs are Evolutionary Strategies (ES), Genetic Algorithms (GA), Particle Swarm Optimization (PSO) and Differential Evolution (DE), among others (Bozorg-Haddad et al., 2017). Even if these techniques lack of any theoretical convergence proof, it has been empirically demonstrated that they are able to identify good quality solutions and several advantages can be highlighted : (i) these population-based algorithms do not require any mathematical property for the treated problem as they only use the evaluation of the objective function; (ii) due to their population-based nature, EAs are particularly suited to tackle problems in which more than one objective is to be optimized (which will be more relevant in the following chapters). This property is important in PSE for which several kinds of objectives (for instance, environmental or social impacts) need to be minimized simultaneously with an economic criterion. However, as EAs have been designed as directed search engines to work over unconstrained spaces, a drawback appears when tackling constrained problems. Thus, in their canonical version, the applicability of EAs is limited by a deficiency of general techniques to manage constraints. Consequently, much effort has been made to efficiently incorporate constraint-handling techniques into EAs. The next section provides a literature review on this subject.

2.3 Literature review

Evolutionary computation researchers have proposed a number of approaches for handling constraints, beginning, as in mathematical programming, by penalty functions. Nevertheless, though penalty functions may work well in some problems, they involve the tuning of some parameters that may affect significantly the quality of the final solutions found. For this reason, in Michalewicz and Janikow (1996), the authors proposed a co-evolutionary algorithm for ensuring feasibility of individuals in problems containing only linear constraints. Equality constraints are eliminated by substituting an equal number of decision variables, the feasible space being then a convex set defined by linear inequalities. Thanks to this property, the genetic operators consist of linear combinations of individuals, which ensure the feasibility of the solutions thus created. Maintaining feasibility can also be obtained through "decoders", i.e. instructions contained in the chromosome that dictate a process to build feasible solutions. In Arnold and Porter (2015), an approach for handling inequality constraints in (1+1)-ES uses an augmented Lagrangian technique (which involves additional quadratic penalty terms) and the performance of the algorithm is evaluated on sphere and ellipsoid functions with a single linear constraint. This approach was then extended by Atamna et al. (2016) to work with an adaptive covariance matrix evolution strategy (CMA-ES); the performance of the algorithm was explored over a set of linearly constrained functions that include convex, quadratic and ill-conditioned objective functions, observing a linear convergence to the op-

timum. In another work, Deb (2000) proposed the well-known feasibility rules to direct the search towards feasible search spaces. As its name suggests, this technique consists on three simple rules that promote the preservation of feasible solutions in the population. Though this technique can be suitable for addressing problems containing inequality constraints, poor results are typically obtained for problems with equality constraints, because the exploration is hindered in early generations. Then, in order to overcome the parameter-setting drawback when using penalty methods, stochastic ranking was proposed in Runarsson and Yao (2000), as an attempt to balance the priority between the objective function and constraint violation without the need of tuning any penalty factor. This method showed promising results when solving benchmark problems, in particular when several constraints are active at the optimal solution. Later, the same authors introduced in Runarsson and Yao (2005) an extension of stochastic ranking for multiobjective optimization problems, conducting to a parameter bias-free search. Another classical approach consists in allowing constraint relaxation at some extent at early generations, and then to stress the search to feasible regions in later generations. The ε-constraint method (Takahama and Sakai, 2005) stands in this framework, in which a polynomial function is employed for decreasing the relaxation level throughout the evolutionary process. The same authors also proposed some extensions of this method (Takahama and Sakai, 2006, 2010), suggesting the preservation of elite solutions and the reparation of infeasible ones by using the constraint gradient information. In Zhang and Rangaiah (2012), another relaxation method is proposed, in which the relaxation level is computed at each generation considering the fraction of feasible solutions in the population and the median of the total violation of constraints. The authors considered in conjunction a self-adaptive differential evolution algorithm with tabu search, and employed a gradient-based local optimizer at the end of the search for ensuring convergence to the feasible region. Nevertheless, exploiting the constraint gradient information for solving constrained problems within EAs, was presented for the first time in Chootinan and Chen (2006); the obtained results indicated a slight superiority of the method over stochastic ranking. In Kheawhom (2010), the constraint gradient information is employed for repairing solutions that violate only equality constraints. The basic idea explored in this method is that, for a given infeasible solution, some variables can be set to a fixed value, while a number of variables equals to the number of equality constraints are repaired, by solving the system of equations (constraints).

Surprisingly enough, even if EAs have been widely used in literature for the solution of PSE optimization problems, the static penalty function and the feasibility rules are the two constraint-handling techniques almost always considered, whereas the performance of more recent methods has not been investigated in the PSE framework, at least not in a systematic manner. Further, in order to improve the performance of EAs in constrained problems, a reformulation of the problem is often carried out to reduce the number of decision variables and to remove, as many as possible, equality constraints. However, despite its efficiency, such an approach may not work properly for large-scale real-world problems that usually contain a important number of constraints. In addition, these constraints may be nonlinear, making it difficult the solution of the corresponding equation system. In Cardoso et al. (1997), authors tackled the solution of chemical engineering MINLP problems using an algorithm combining simulated annealing (for treating discrete variables) and the nonlinear simplex method (for treating continuous variables); for constraint handling, the authors employed a dynamic penalizing scheme that considers the

maximum extent of constraint violation in such a way that no penalty parameter needs to be tuned. Results showed that this method, though efficient for most of the treated problems, experiences difficulties for finding the global optimum in highly constrained problems. In Costa and Oliveira (2001), the authors studied the solution of seven chemical engineering optimization problems, comparing the performance of both genetic algorithms and evolution strategies. In this work, equality constraints were eliminated, therefore reducing the number of independent decision variables, whereas inequality constraints were treated using feasibility rules. It is noteworthy that only small problems (containing up to 10 variables) were studied. A similar study was carried out in Babu and Angira (2006), where authors employed differential evolution as a search engine, while constraints were handled using Deb's feasibility rules and, for problems presenting equality constraints, the model was reformulated in order to eliminate equality constraints and set some variables as dependent from the others. However, conclusions about the efficiency of this methodology for the solution of real-world problems are not clear, since only small instances were studied (the largest one involves only 7 continuous variables, 3 binary variables and no equality constraints). In Srinivas and Rangaiah (2007), the authors treated an important number of problems, implementing DE with tabu search to enhance the search. All the equality constraints were removed through model reformulation and the static penalty function was employed for handling inequality constraints. Actually, the authors used a high value of the penalty factor for all problems, which can be viewed as equivalent to using feasibility rules. In Yiqing et al. (2007), an enhanced version of the PSO algorithm was proposed, in which constraint handling was performed in two steps: (1) problem reformulation (reduction of independent decision variables) to remove all the equality constraints, and (2) Deb's feasibility rules for inequality constraints, pushing the search towards feasible regions. Also, as mentioned earlier, in Kheawhom (2010), a gradient-based reparation procedure was proposed to efficiently treat equality constraints without the need of reformulating the model, however, the results obtained seem to indicate that this method needs an important number of function evaluations for converging (approx. 10^6 for a problem containing 22 variables). Finally, the optimization of dynamic chemical processes was addressed in Chen et al. (2016) using DE with a modification in the mutation process, while constraints (both equality and inequality) are handled through Deb's feasibility rules.

Taking into account the above considerations, the scientific objective and the main contributions of this chapter are therefore to explore and compare the performance of some state-of-the art constraint-handling techniques, embedded in a metaheuristic search engine, for the solution of a selection of problems drawn from the process engineering framework. In particular, five constraint-handling techniques are investigated, namely penalty functions, stochastic ranking (Runarsson and Yao, 2000), feasibility rules (Deb, 2000), ε-constraint method (Takahama and Sakai, 2005) and gradient-based repair (Chootinan and Chen, 2006). It is worth mentioning that this diversity of methods is representative, by their respective working mode, of those that are typically used in most works found in the specialized literature. Since the aim is to provide a fair basis for constraint-handling methods comparison, the same search engine was used in all cases, i.e. a self-adaptive variant of DE (Brest et al., 2006), both because of its simplicity (it does not introduce any sophisticated operator nor any non-uniform probability distribution) and its claimed superiority over GAs and PSO in terms of its computational efficiency (Ponsich and Coello Coello, 2011; Vesterstrom and Thomsen, 2004). The aforementioned constraint-handling strategies have thus

been incorporated within DE for the solution of 14 well-known chemical engineering problems, highlighting the superiority of the gradient-based repair strategy for most of the treated examples.

2.4 Constraint-handling techniques

In this section, five popular techniques for handling constraints in evolutionary computation are presented, namely, penalty functions, Deb's feasibility rules, stochastic ranking, ε-constraint method and gradient-based repair. As indicated in the introduction section, many other techniques exist in the specialized literature (Coello Coello, 2016; Mezura-Montes and Coello Coello, 2011), but there is no evidence that they significantly outperform the techniques tackled here and, in addition, these more sophisticated techniques usually involve the setting of several control parameters (Mallipeddi and Suganthan, 2010; Mezura-Montes and Coello Coello, 2005; Padhye et al., 2015; Samanipour and Jelovica, 2020; Yang et al., 2020). Therefore, only these five constraints handling methods, which might be defined as state-of-the-art techniques, are tackled in this study.

2.4.1 Penalty functions

Historically, the most common approach to incorporate constraints (both in evolutionary algorithms and in mathematical programming) involves penalty functions, which were originally proposed in the 1940s and later expanded in many research studies, mainly for their simplicity and efficiency. With this method, the fitness landscape is modified as some penalty is added to the objective value of each infeasible individual. In their general form, penalty functions can be represented as:

$$\psi(\mathbf{x}) = f(\mathbf{x}) + \sum_{i=1}^{p} r_i \cdot \max\{0, g_i(\mathbf{x})\}^{\alpha} + \sum_{j=1}^{q} c_j \cdot \left|h_j(\mathbf{x})\right|^{\alpha} \tag{2.1}$$

where $\psi(\mathbf{x})$ is the new fitness function to be minimized, r_i and c_j are positive constants called *penalty factors*, and α normally takes values of 1 or 2.

As can be noted, this implementation, though quite simple, requires the use of a number of parameters to be tuned (equal to the number of constraints) which might be impractical in highly constrained problems. For this reason, the static penalty function, the simplest form of penalty function, has remained as the most popular one:

$$\psi(\mathbf{x}) = f(\mathbf{x}) + r \cdot \phi(\mathbf{x}) \tag{2.2}$$

where r is the penalty coefficient and $\phi(\mathbf{x})$ is the overall constraint violation:

$$\phi(\mathbf{x}) = \sum_{i=1}^{p} \max\{0, g_i(\mathbf{x})\}^{\alpha} + \sum_{j=1}^{q} \left|h_j(\mathbf{x})\right|^{\alpha} \tag{2.3}$$

Although the static penalty function only needs the tuning of one parameter, this task is not straightforward. On the one hand, if r is too small, the search will be directed towards regions where the objective function is minimized, but the final obtained solutions are likely to be infeasible. On the other hand, if r is too high,

the minimization of the overall constraint violation will be prioritized, obtaining a feasible solution in early generations with the disadvantage that, if the search space is disconnected or highly constrained, it will be very difficult to escape from the first feasible region found and the process thus may get stuck in a local optimum. Ideally, the penalty should be kept as low as possible, just above the limit where the solutions found are infeasible, this strategy is called the *minimum penalty rule*.

Furthermore, dynamic penalty functions, in which the coefficient r varies throughout the evolutionary process, have been proposed (Coello Coello, 2000; Nanakorn and Meesomklin, 2001; Tessema and Yen, 2006). The underlying idea in dynamic penalty functions is that allowing low values of r at early generations enables to explore the regions where the objective function is minimized, whereas a high value of r is desired at final generations in order to push the search towards the feasible region. Such an idea would work well for problems in which the unconstrained global optimum is close to the constrained one, but there is no guarantee that this strategy will be efficient in all cases. Besides, the additional parameters needed to define the penalty coefficient increasing schedule make this method less attractive than the simple static penalty function.

Finally, since a good choice of the penalty coefficient is necessary to enable a good balance between the objective function and the overall constraint violation minimization, adaptive strategies have been suggested where information gathered from the search process is used to control the amount of penalty added to infeasible individuals. Adaptive penalty functions are not difficult to implement and they usually do not require user-defined parameters. Nevertheless, the results available in the literature are not very encouraging as adaptive penalty methods usually need a lot of iterations to find the optimal solution as illustrated in Tessema and Yen (2006). For a more complete review of adaptive penalty techniques the reader is referred to Barbosa et al. (2015).

2.4.2 Stochastic ranking

Stochastic ranking (SR) has been proposed by Runarsson and Yao (2000) as an attempt to balance the relative weights of the objective function and constraint violation components that compete within penalty functions. In this method, the population is sorted following a probabilistic procedure: two individuals are compared according to their objective function with a probability P_f, otherwise, the overall constraint violation is used for the comparison as indicated in the pseudocode presented in Algorithm 4. Once the population has been sorted by SR, a part of the population assigned with highest rank is selected for recombination, thus sharing its characteristics to the next generation. In this way, the search is directed by the minimization of the objective function and by feasibility concepts at the same time.

Since stochastic ranking was originally designed to work with Evolution Strategies (ES), which indeed requires a ranking process in its replacement mechanism, its implementation within other search paradigms is not straightforward, even if some studies have extended its use to other EAs (Ali et al., 2012; Fan et al., 2009; Zhang et al., 2008a). Considering DE, SR could be used in two different ways: for selecting a part of the population that would participate in the mutation process, or for selecting the individuals that would survive to the next generation (after the mutation and crossover processes). In Fan et al. (2009), the authors proposed to rank the population according to the SR procedure before the mutation process: they divide the population

Algorithm 4 Stochastic ranking procedure

```
for all i ∈ {1, ..., μ} do
    for all j ∈ {1, ..., μ − 1} do
        sample u ∼ U(0, 1)
        if φ(x^(j)) = φ(x^(j+1)) = 0 or u < P_f then
            if f(x^(j)) > f(x^(j+1)) then
                swap(x^(j), x^(j+1))
            end if
        else
            if φ(x^(j)) > φ(x^(j+1)) then
                swap(x^(j), x^(j+1))
            end if
        end if
    end for
    if no swap done then
        break
    end if
end for
```

into two parts (that they call higher and lower parts), the upper part containing the *best* individuals, i.e., the individuals ranked highest after SR. Then, for every trial vector, $\mathbf{v}^{(i)}$, the upper part contributes with two *good* individuals, while the lower part provides only one *less good* individual. This procedure was initially considered in this study, but since the obtained results were not satisfactory, SR was implemented within the selection process as follows: the new population is generated normally by the DE operator, i.e., using the entire population, and then both populations (parents and offspring) are ranked according to SR. Finally, each new individual is compared with his parent, and that with a higher rank survives to the next generation.

2.4.3 Feasibility rules

This constraint handling technique establishes the superiority of feasible solutions over infeasible ones, that is, as opposite to the penalty functions, feasibility rules do not merge information from both constraint violation and objective function, but consider them separately. Proposed in Deb (2000), the feasibility rules (also called lexicographical order) consist in a binary tournament selection that uses the following criteria:

1. Any feasible solution is preferred to any infeasible solution.
2. Among two feasible solutions, that with a better objective function value is preferred.
3. Among two infeasible solutions, that with a smaller constraint violation is preferred.

Feasibility rules represent an easy-to-implement, parameter-free technique to handle constraints. Further, due to its simplicity and its overall good performance, feasibility rules are usually the first constraint-handling technique tested for treating a given problem with EAs. However, one of the main drawbacks of this method appears when dealing with problems with a reduced and/or disconnected feasible

region (e.g., problems with one or several equality constraints). Because any feasible solution is preferred over an infeasible one, once the algorithm has converged to some feasible region, it may be very difficult to escape from there in order to explore other regions, i.e., once the constraints are fulfilled, the algorithm is very likely to get trapped prematurely in some subregion of the search space. Moreover, considering that there are high probabilities that the optimum lies close to the feasibility boundary, slightly infeasible solutions might be more useful to the search process than solutions wide inside the feasible region. However, the feasibility rules would prefer the latter solution to the former one. In fact, feasibility rules can be seen as a limiting case of static penalty function when the penalty value takes a very high value, in this way, when comparing two solutions, that with a lesser amount of overall constraint violation will be always preferred.

2.4.4 ε-constraint method

In order to tackle the above-mentioned issues related to feasibility rules in severely constrained problems, the ε-constraint method for evolutionary algorithms has been proposed by Takahama and Sakai (2005), where a relaxation of constraints is permitted to explore constrained regions. A tolerance level over the constraint violation, called the ε level, indicates the limit under which solutions are considered as feasible, that is, a solution $\mathbf{x}$ such that $g_i(\mathbf{x}) \leq \varepsilon$ and $|h_j(\mathbf{x})| \leq \varepsilon \; \forall i, j$ is viewed as feasible. Once the feasibility of solutions has been identified by means of the ε level, the lexicographical order (i.e. feasibility rules) is used for selecting the surviving individuals for the next generation. This technique has proven to be especially efficient in highly constrained problems, such as those involving equality constraints, because the relaxation allowed at the early generations, i.e. with a relatively high ε level, promotes the exploration of regions that would be impossible to reach by simple feasibility rules.

The main drawback of this method is the difficulty for setting the ε parameter. It has been remarked that ε level enables a good exploration of the search space in early generations but also it is clear that ε must be zero at some point of the evolutionary process in order to obtain feasible solutions. In Takahama and Sakai (2006), the authors proposed a dynamic control of ε level, according to:

$$\varepsilon(0) = \phi(\mathbf{x}^\theta) \tag{2.4}$$

$$\varepsilon(t) = \begin{cases} \varepsilon(0)(1 - \frac{t}{T_c})^{cp}, & 0 < t < T_c, \\ 0, & t \geq T_c \end{cases} \tag{2.5}$$

where $\mathbf{x}^\theta$ is the best θ-th individual (in terms of constraint violation) in the first generation, cp is a parameter to control the decrease speed of the ε level and T_c represents the generation after which the ε level is set to 0 (after that generation, Deb's feasibility rules are considered). According to the authors, the following parameter setting works well in many problems: $\theta = 0.2\mu$, $cp = 5$, $T_c \in 0.2T_{max}$. Despite these general guidelines, the tuning of these three parameters still constitutes a drawback, as it might become a harsh task. Additionally, it is important to recall that the use of the ε level according to equation (2.5) is only recommended for highly constrained problems in which the feasibility rules do not work properly, otherwise ε-constraint method may get worse results than the feasibility rules in terms of efficiency and efficacy.

It should be underlined that the ε-constraint method obtained the first place in the competition on constrained optimization of the Congress of Evolutionary Computation (CEC 2006) and very competitive results in CEC 2010 (Takahama and Sakai, 2006, 2010). Due to this success, the ε-constraint method has also been embedded in a number of algorithms for multiobjective optimization (Fan et al., 2016, 2017; Yang et al., 2014). However, the excellent results obtained by this method in the two above-mentioned competitions did not only depend on the ε level relaxation strategy, but also on the use of an additional operator, based on a gradient-based repair process (presented in the next section). It is difficult to determine which of both methods contribute the most in obtaining the excellent results published at CEC'2006 and CEC'2010. In this study, the ε-constraint method has been implemented with and without the gradient-based repair, so that the relative performance of both algorithms can be evaluated and compared.

2.4.5 Gradient-based repair

The gradient-based repair method, proposed by Chootinan and Chen (2006) is a constraint-handling technique that uses the gradient information derived from the constraint set to systematically repair infeasible solutions. Basically, the gradient of constraint violation is used to direct infeasible solutions toward the feasible region. The vector of constraint violations $\Delta C(\mathbf{x})$ is defined as:

$$\Delta C(\mathbf{x}) = [\Delta g_1(\mathbf{x}), \ldots, \Delta g_m(\mathbf{x}), \Delta h_1(\mathbf{x}), \ldots, \Delta h_p(\mathbf{x})]^T \tag{2.6}$$

where $\Delta g_i(\mathbf{x}) = \max\{0, g_i(\mathbf{x})\}$ and $\Delta h_j(\mathbf{x}) = h_j(\mathbf{x})$. This information, in addition to the gradient of constraints $\nabla C(\mathbf{x})$, is used to determine the step $\Delta\mathbf{x}$ to be added to the solution $\mathbf{x}$, according to:

$$\nabla C(\mathbf{x})\Delta\mathbf{x} = -\Delta C(\mathbf{x}) \tag{2.7}$$

$$\Delta\mathbf{x} = -\nabla C(\mathbf{x})^{-1}\Delta C(\mathbf{x}) \tag{2.8}$$

Although the gradient matrix ∇C is not invertible in general, the Moore-Penrose inverse or pseudoinverse $\nabla C(\mathbf{x})^+$ (Campbell and Meyer, 2009), which gives an approximate or best (least square) solution to a system of linear equations, can be used instead in (2.8). Thus, once the step $\Delta\mathbf{x}$ has been computed, the infeasible point $\mathbf{x}$ is moved to a less infeasible point $\mathbf{x} + \Delta\mathbf{x}$. This repair operation is performed with a probability P_g and repeated R_g times while the point is infeasible.

In this work, the computation of the gradient $\nabla C(\mathbf{x})$ is done numerically using forward finite differences for all problems. Also, it is worth noting that in the original article (Chootinan and Chen, 2006) only non-zero elements of $\Delta C(\mathbf{x})$ are repaired, i.e., the gradient is only computed for constraints that are indeed violated. On the contrary, in Takahama and Sakai (2006) all constraints are considered in the repair process, even those that are already satisfied. The former approach has the disadvantage that a given constraint may be fulfilled at one iteration but violated in the next on. Nevertheless, it was empirically observed in this work that the former approach is usually more efficient than the latter one, in terms of number of iterations needed to get to the feasible region. So, in this study, only non-zero elements of $\Delta C(\mathbf{x})$ are taken into account within the repair process. Note that this procedure can also produce situations where some variables lie outside their allowed variation range,

so that two inequality constraints may be added for each variable, accounting for their bounds. Due to the associated computational burden in real-world optimization problems, where the number of variables may be high, these additional constraints are not considered here. Instead, an additional repair process, performed at each iteration, sets the variable value to the violated bound if necessary. The pseudocode of the gradient-based repair procedure used in this study is presented in Algorithm 5.

Algorithm 5 Gradient-based repair procedure

for all $i \in \{1, \ldots, \mu\}$ **do**
  $t = 0$
  sample $u \sim U(0, 1)$
  while $t < R_g$ **and** $\phi(\mathbf{x}^{(i)}) > 0$ **and** $u < P_g$ **do**
    compute $\nabla C(\mathbf{x})$ of violating constraints
    compute $\nabla C(\mathbf{x})^+$
    compute $\Delta\mathbf{x}$
    $\mathbf{x}^{(i)} \leftarrow \mathbf{x}^{(i)} + \Delta\mathbf{x}$
    repair $\mathbf{x}^{(i)}$ to its bounds
    compute $\Delta C(\mathbf{x})$
    $t \leftarrow t + 1$
  end while
end for

Even if the gradient-based repair can be considered as a constraint-handling technique by itself, using it alone would be computationally expensive, since, in highly constrained spaces, this procedure might require many iterations to reach the feasible region, and in extreme cases, it might be impossible to reach a feasible solution. Therefore, this technique is usually coupled with any other constraint-handling technique. In this work, gradient-based repair is coupled with the ε-constraint method because, by its working mode, it presents advantages over feasibility rules when dealing with equality constraints and, in addition, methods like penalty functions or stochastic ranking might spoil the efforts made to repair infeasible solutions, due to a bad-tuned penalty factor or to the stochastic ranking sorting procedure itself. Nevertheless, for illustrative purposes, a preliminary test experiment is carried out for one instance (defined in the next section), by applying the gradient-based repair method to each of the four constraint-handling techniques. The results are shown in Table 2.1, in terms of the number of function evaluations (NFEs). It can be observed that every constraint-handling technique, when coupled with gradient-based repair, performed equally good. Since no significant differences are observed, it seems reasonable to use the constraint gradient-based repair procedure in combination with only one of the remaining techniques. Note that the identical results observed for feasibility rules and ε-constraint methods are due to the ε level, which, in this case, is zero at the first generation.

Table 2.1: Gradient-based repair coupled with each constraint-handling method. Experimental results for problem 1 in terms of NFEs needed to achieve convergence.

Problem	Constr-handling	Best	Median	Worst	Mean	Std	Feas. rate	Succ. rate	CPU time(s)
1	St. penalty fcn.	211	1884	4665	1943	1009	100	100	2.6
	SR	211	2063	6151	2295	1419	100	100	3.3
	Feas. rules	211	2142	5490	2176	1226	100	100	2.9
	ε-constraint	211	2142	5490	2176	1226	100	100	2.9

2.5 Computational experiments

To illustrate the benefits of the above-mentioned constraint-handling techniques in process engineering applications, 14 PSE problems have been selected as representative in the specialized literature. These problems present distinct mathematical characteristics typically found in process engineering, e.g., nonlinearities, equality and inequality constraints, binary and continuous variables. Some relevant features of these examples are provided in Table 2.2. In Appendix A, the complete formulation of these problems is presented in details and additional information concerning local and global optimal solutions is also given.

Table 2.2: Brief description of example problems

Problem	Decision variables		Constraints (active at optimal solution)	Description
	Binary	Continuous		
1	0	6	5(5)	Reactor network design
2	0	10	6(6)	Flowsheeting
3	1	1	2(1)	Process synthesis
4	1	2	2(2)	Process synthesis
5	1	2	3(2)	Process synthesis
6	3	2	5(3)	Process synthesis
7	2	6	8(6)	Reactor network design
8	4	3	9(5)	Process synthesis
9	3	8	9(7)	Planning problem
10	5	7	13(7)	Batch plant design
11	5	7	13(8)	Batch plant design
12	12	16	61(15)	Batch plant design
13	14	19	85(18)	Batch plant design
14	16	20	97(16)	Batch plant design

The algorithms previously presented were implemented with MATLAB R2017b and all the following computational experiments were carried out with a processor Intel Xeon E3-1505M v6 at 3.00 GHz and 32 Go RAM.

Parameters settings

In order to perform a fair comparison of the different constraint-handling methods, the parameters tuning has been set constant for each method over all the test problems, so that the overall performance without specific adjustments is assessed for each technique. Obviously, the only exception is the static penalty function, for which the

tuning of the penalization parameter r for each problem is intrinsic to the method. The actual parameters used are:

- Static penalty function. Parameter r is tuned following the *minimum penalty rule*, that is, r takes the lowest possible values so as to obtain feasible solutions. The precision of the parameter is set according to $r = x \times 10^y$, where x and y are non-negative integer numbers.
- Stochastic ranking. $P_f = 0.45$.
- ε-constraint method. $\theta = 0.2\mu$, $cp = 5$, $T_c = 0.2T_{max}$.
- Gradient-based repair. Identical parameters as for the ε-constraint and additionally, $P_g = 1$, $R_g = 3$. Besides, each computation of the gradient of constraints counts as 1 function evaluation.

Regarding the DE algorithm, the only parameter to be tuned is the population size (μ), as the scaling factor (F) and crossover rate (CR) are adjusted by the auto-adaptive algorithm. The population size is calculated as $\mu = \min(100, 10n)$ where n is the number of decision variables. The algorithm stops if the current best solution is as close as 0.0001% or 0.0001 to the reported global optimal solution or if the number of function evaluations (NFEs) exceeds 200 000. Due to the stochastic nature of evolutionary algorithms, 50 independent executions are carried out for each problem and each method.

2.6 Results and discussion

The results obtained for the 14 optimization test problems are summarized in Tables 2.3 to 2.6. (please note that all problems are minimization problems). The results are analyzed through the best, median and worst objective function values, "$-$" means that no feasible solution was found. Feasibility and success rates represent the rates of executions respectively finding at least one feasible solution and the optimal solution over the 50 independent runs. Please note that the computational times in Tables 2.4 and 2.6 represent the overall elapsed time for the 50 runs. In Tables 2.3 and 2.4 the results for problems 1 to 9 are shown. In Table 2.3 better results according to the best, median and worst solution found are represented in boldface. Similarly, better results according to the NFEs, and feasibility and success rates are presented in boldface in Table 2.4. Since problems 10 to 14 are different size instances of the same problem (the optimal design of a multi-product batch plant), their results are shown together in Tables 2.5 and 2.6.

Problem 1 addresses the optimal design of a sequence of two CSTR reactors. It can be considered as a small size problem, however, from Table 2.4 it can be appreciated that only the gradient-based repair technique achieved to find the optimum in each single run. Further, the low NFEs needed to converge is to be highlighted. Note that the numerical differences observed in Table 2.3, corresponding to the gradient-based repair method are not significant, considering that the optimal stopping criterion is reached for every solution, i.e., each solution is as close as 0.0001 from the reported optimum. This problem was studied in Babu and Angira (2006) and Srinivas and Rangaiah (2007), through the reformulation consisting in removing all equality constraints and eliminating dependent variables, and the use of a static penalty function.

Table 2.3: Experimental results in terms of objective function values.

Problem (optimum)	Constr-handling	Best	Median	Worst	Mean	Std
1 (-0.38881)	St. penalty fcn.	−0.38881	−0.38871	−0.38802	−0.38870	1.42e-04
	SR	−0.38881	−0.38871	−0.37867	−0.38813	1.87e-03
	Feas. rules	−0.38881	−0.38871	−0.38377	−0.38854	7.77e-04
	ε-constraint	−0.38881	−0.38871	−0.38720	−0.38848	4.24e-04
	Grad-based repair	−0.38881	−0.38876	−0.38872	−0.38878	3.92e-05
2 (9490593)	St. penalty fcn.	10 041 455	12 562 687	16 932 972	12 626 104	1.32e+06
	SR	—	—	—	—	—
	Feas. rules	10 148 136	12 421 765	18 043 019	12 667 483	1.40e+06
	ε-constraint	10 603 444	—	—	13 457 709	1.88e+06
	Grad-based repair	**9 490 594**	**9 490 600**	**9 490 603**	**9 490 600**	**2.51e+00**
3 (2.000)	St. penalty fcn.	2.000	2.000	2.236	2.005	3.34e-02
	SR	**2.000**	**2.000**	**2.000**	**2.000**	**3.00e-05**
	Feas. rules	2.000	2.000	2.236	2.019	6.47e-02
	ε-constraint	2.000	2.000	2.236	2.014	5.66e-02
	Grad-based repair	**2.000**	**2.000**	**2.000**	**2.000**	**3.03e-05**
4 (2.124)	St. penalty fcn.	2.124	2.124	2.558	2.168	1.31e-01
	SR	2.124	2.124	2.558	2.142	8.58e-02
	Feas. rules	2.124	2.558	2.558	2.549	6.13e-02
	ε-constraint	2.124	2.124	2.558	2.150	1.04e-01
	Grad-based repair	**2.124**	**2.124**	**2.124**	**2.124**	**1.71e-05**
5 (1.0765)	St. penalty fcn.	1.0766	1.0766	1.2500	1.0801	2.45e-02
	SR	1.0766	1.0766	1.2500	1.0835	3.43e-02
	Feas. rules	1.0766	1.0766	1.2500	1.0880	4.19e-02
	ε-constraint	**1.0766**	**1.0766**	**1.0766**	**1.0766**	**1.80e-05**
	Grad-based repair	**1.0765**	**1.0766**	**1.0766**	**1.0766**	**3.14e-05**
6 (7.667)	St. penalty fcn.	7.667	7.667	7.931	7.688	7.22e-02
	SR	7.667	7.667	7.931	7.693	7.99e-02
	Feas. rules	7.667	7.931	8.240	7.928	1.07e-01
	ε-constraint	7.667	7.931	7.931	7.846	1.24e-01
	Grad-based repair	**7.667**	**7.667**	**7.667**	**7.667**	**1.49e-05**
7 (99.238)	St. penalty fcn.	99.238	99.240	107.374	101.355	3.60e+00
	SR	99.238	99.239	107.374	100.703	3.16e+00
	Feas. rules	99.238	107.374	—	111.974	2.30e+01
	ε-constraint	99.238	107.374	107.374	103.795	4.08e+00
	Grad-based repair	**99.238**	**99.240**	**99.240**	**99.239**	**2.69e-04**
8 (4.57958)	St. penalty fcn.	4.57958	4.57962	4.57968	4.57962	3.18e-05
	SR	4.57958	4.57967	4.57968	4.57966	1.69e-05
	Feas. rules	4.57958	4.57966	4.57968	4.57966	1.86e-05
	ε-constraint	4.57958	4.57966	4.57968	4.57966	2.14e-05
	Grad-based repair	4.57958	4.57958	4.57964	4.57958	8.95e-06
9 (-1.9231)	St. penalty fcn.	−1.9231	−1.7236	−1.4125	−1.6925	1.95e-01
	SR	−1.9231	−1.7235	−0.2202	−1.5924	4.51e-01
	Feas. rules	−1.4125	−1.2138	0.7607	−0.8621	6.70e-01
	ε-constraint	−1.4099	−0.0011	0.7431	−0.0370	3.71e-01
	Grad-based repair	**−1.9231**	**−1.9231**	**−1.9230**	**−1.9230**	**1.31e-04**

Table 2.4: Experimental results in terms of NFEs needed to achieve convergence.

Problem	Constr-handling	Best	Median	Worst	Mean	Std	Feas. rate	Succ. rate	CPU time(s)
1	St. penalty fcn.	14 340	64 230	200 000	88 734	73 334	100	90	10.1
	SR	45 480	156 840	200 000	153 434	44 795	100	70	91.6
	Feas. rules	27 660	149 490	200 000	136 008	53 187	100	86	15.6
	ε-constraint	32 700	193 320	200 000	156 421	61 826	100	54	19.1
	Grad-based repair	**211**	**2142**	**5490**	**2176**	**1226**	**100**	**100**	**2.9**
2	St. penalty fcn.	200 000	200 000	200 000	200 000	0	100	0	19.8
	SR	200 000	200 000	200 000	200 000	0	0	0	142.2
	Feas. rules	200 000	200 000	200 000	200 000	0	100	0	19.3
	ε-constraint	200 000	200 000	200 000	200 000	0	20	0	21.0
	Grad-based repair	**15 300**	**19 100**	**22 800**	**19 094**	**1366**	**100**	**100**	**90.3**
3	St. penalty fcn.	500	870	200 000	4851	28 162	100	98	1.2
	SR	640	1350	2640	1423	444	100	100	1.2
	Feas. rules	520	1110	200 000	17 006	54 511	100	92	4.3
	ε-constraint	420	1550	200 000	13 583	47 582	100	94	3.8
	Grad-based repair	**33**	**203**	**387**	**226**	**89**	**100**	**100**	**0.3**
4	St. penalty fcn.	1680	2775	200 000	22 441	59 792	100	90	3.6
	SR	3540	6825	200 000	14 887	38 229	100	96	6.6
	Feas. rules	10 200	200 000	200 000	196 214	26 843	100	2	31.2
	ε-constraint	24 090	28 245	200 000	38 545	41 240	100	94	7.4
	Grad-based repair	**100**	**108**	**676**	**197**	**151**	**100**	**100**	**0.2**
5	St. penalty fcn.	1830	2895	200 000	6804	27 885	100	98	1.2
	SR	4620	6465	200 000	14 212	38 332	100	96	6.0
	Feas. rules	2610	4065	200 000	19 623	53 738	100	92	3.2
	ε-constraint	22 800	28 575	31 200	27 988	2182	100	100	6.0
	Grad-based repair	**93**	**175**	**1046**	**244**	**173**	**100**	**100**	**0.3**
6	St. penalty fcn.	2400	4125	200 000	19 780	53 686	100	92	2.5
	SR	10 150	14 500	200 000	33 204	56 229	100	90	15.8
	Feas. rules	3900	200 000	200 000	184 422	53 365	100	8	22.2
	ε-constraint	14 550	200 000	200 000	143 018	83 935	100	32	18.2
	Grad-based repair	**195**	**198**	**360**	**207**	**37**	**100**	**100**	**0.3**
7	St. penalty fcn.	16 880	22 280	200 000	68 891	78 739	100	74	6.1
	SR	39 120	69 600	200 000	92 301	52 791	100	82	60.9
	Feas. rules	92 720	200 000	200 000	190 869	23 914	86	22	16.4
	ε-constraint	35 280	200 000	200 000	128 550	81 429	100	44	11.8
	Grad-based repair	**3761**	**9591**	**20 639**	**10 082**	**3574**	**100**	**100**	**10.3**
8	St. penalty fcn.	5810	6720	8540	6891	714	100	100	0.9
	SR	8330	11 480	14 770	11 739	1485	100	100	8.2
	Feas. rules	7840	9415	11 270	9362	785	100	100	1.3
	ε-constraint	8120	25 760	30 730	23 408	6563	100	100	3.3
	Grad-based repair	**407**	**2133**	**9879**	**3691**	**3276**	**100**	**100**	**3.3**
9	St. penalty fcn.	47 800	200 000	200 000	159 822	65 129	100	28	16.0
	SR	149 200	200 000	200 000	197 052	9930	100	18	149.5
	Feas. rules	200 000	200 000	200 000	200 000	0	100	0	20.8
	ε-constraint	200 000	200 000	200 000	200 000	0	100	0	20.9
	Grad-based repair	**16 197**	**41 450**	**77 610**	**41 982**	**10 260**	**100**	**100**	**66.3**

In contrast, the results obtained by the gradient-based repair method suggest that this reformulation is not necessary since this method finds the global optimum in short CPU times (lower than 0.1 second per run).

Problem 2 constitutes a difficult case, containing 6 nonlinear equality constraints that involve all decision variables. Not any one of the tested constraint-handling techniques, excepting the gradient-based repair, was able to found the optimum in any run. It is observed, however, that the computational time required for computing the gradient is considerable higher that those of simple techniques, e.g., static penalty function, even if less functions evaluations are carried out. In Rangaiah (2009), this problem was also addressed by DE considering the constraints as a system of nonlinear equations which is solved by an exact algorithm, so that the original problem is transformed into an unconstrained one. However, such strategy is computationally expensive, as it is performed for every individual at each generation. In this study, the same approach has been carried out for comparison purposes. The system of nonlinear equations has been solved using the Levenberg-Marquardt algorithm embedded in the MATLAB software. This approach took approximately 40 seconds per run, i.e., about 20 times more than the gradient-based repair procedure.

Problems 3 and 5 consist in small and rather simple MINLP examples. All constraint-handling methods obtained an overall good performance in terms of success rate. Notwithstanding, the gradient-based repair is significantly more efficient both in terms of NFEs and CPU time.

Problem 4 is modeled as a MINLP involving one nonlinear equality constraint and one binary variable, which together yield a rather high difficulty for the solution by feasibility rules, since this technique gets trapped in an "easy-to-access" local optimum, explaining its low success rate. Reversely, all the other techniques obtain a very good performance, in particular the gradient-based repair, which yielded the optimum in every run employing an insignificant computational time. It is noteworthy that in Costa and Oliveira (2001) and Yiqing et al. (2007) the problem is reformulated by reducing one continuous variable and thus eliminating the equality constraint. This approach, although efficient, is problem-devoted and may not be practical in highly constrained real-world problems.

Problem 6 takes into account a MINLP problem with 3 binary variables and 2 equality constraints. Although this problem can be considered as a small one (in terms of the number of variables), its characteristics are not easy to overcome by feasibility rules, meaning that the first feasible solution found is likely to be far from the global minimum region. Further, the relaxation done by ε-constraint method does not manage to obtain acceptable success rates, at least not with the parameters used here. Regarding stochastic ranking, static penalty function and gradient-based repair, they solve the problem efficiently, with a better success rate and much lower CPU times reported for the gradient-based repair technique. In Cardoso et al. (1997) and Srinivas and Rangaiah (2007), the same problem was tackled, but the model was simplified by eliminating the continuous variables by means of the equality constraints.

For problem 7, feasibility rules and ε-constraint method present a poor performance due to the existence of 2 binary variables and 4 equality constraints. Stochastic ranking and static penalty function present a fairly good performance. On the contrary, gradient-based repair method enables the algorithm to search in the whole search space before converging to an optimum. Again, this example was addressed in previous works (Cardoso et al., 1997; Costa and Oliveira, 2001; Srinivas and Ran-

gaiah, 2007; Yiqing et al., 2007) by reformulating the problem in order to eliminate equality constraints, thus reducing the number of decision variables. The remaining constraints was handled by the static penalty function.

For problem 8 all constraint-handling techniques performed excellently, finding the global optimum in all runs. For this problem the first feasible region found coincides with the region where the global optimal solution lies, even though the problem presents some difficulties regarding its mathematical properties (4 binary variables with 9 constraints). It is observed that the gradient-based repair is the most efficient in terms of NFEs, whereas the penalty function is the most efficient in terms of computational time.

Problem 9 represents a planning problem in which several alternatives are proposed for obtaining one desired product. Since it contains 3 binary variables and 5 equality constraints involving all the continuous variables, determining the global optimum is a difficult task. For SR, the balance between feasible and infeasible solutions is not enough to reach the global optimum region in most cases. With respect to Deb's feasibility rules, convergence to the global optimum is highly unlikely, since this technique always prefers feasible solutions over the infeasible ones, whatever the quality of the objective function value. For the ε-constraint method, the relaxation conducted on the equality constraints seems not to be sufficient to reach the global optimum. On the other hand, when this relaxation is combined with the gradient-based repair, the algorithm is able to search over the entire space so that the global optimum is found for each execution, with a low NFEs.

Problems 10–14 consider the optimal design of a multi-product batch plant consisting of a given number of processing stages M through which N different products have to be manufactured. The objective is to minimize the investment cost and then, for each processing stage j, the number of parallel units N_j and their size V_j need to be determined, as well as the batch sizes B_i and cycle times $T_{\mathrm{L}i}$ for each product i. Thus, increasing the number of stages M, the number of products N and the possible number of parallel units N_j^{u}, results in a large non-convex MINLP. Note that the mathematical formulation of the problem implemented in this study presents multiple non-convexities in the objective function and in several inequality constraints (see Appendix A). Hence, problems 10–12 are drawn from Kocis and Grossmann (1988) and only differ by their size (in terms of the number of products and processing stage). Also, in order to explore the scalability of the studied constraint-handling techniques, problems 13 and 14 have been artificially created increasing the size of problem 12, both in number of variables and in number of constraints. Thus, since no global optimum is reported in the literature for these problems, a convexified formulation of the problem was solved using the BARON solver within the GAMS environment, the corresponding solutions are reported in Table 2.5.

Problem 10 and 11 are equivalent in size, their only difference is the quantity Q_i of product i that needs to be manufactured. Nevertheless, performance of constraints-handling techniques like ε-constraint, feasibility rules and SR are significantly different for both problems. It seems that this slight modification makes problem 11 much more difficult for these techniques, the feasible region has been modified in such a way that the global optimum lies now in a region that is difficult to reach. Besides, the consistency of the static penalty function and gradient-based repair methods is observed in both problems, as their performance remained unchanged in terms of success rate.

Table 2.5: Experimental results for problems 10–14 in terms of objective function values.

Problem (`optimum`)	Constr-handling	Best	Median	Worst	Mean	Std
10 (`38499.5`)	St. penalty fcn.	38 499.5	38 500.1	38 500.2	38 500.1	5.33e-02
	SR	38 499.5	38 499.8	38 499.8	38 499.8	2.39e-02
	Feas. rules	38 499.5	38 500.1	38 500.2	38 500.1	5.84e-02
	ε-constraint	38 499.5	38 500.2	40 977.5	38 747.9	7.51e+02
	Grad-based repair	38 499.5	38 499.7	38 499.8	38 499.7	8.77e-02
11 (`106755.8`)	St. penalty fcn.	**106 755.8**	**106 756.8**	**106 756.9**	**106 756.8**	**8.69e-02**
	SR	106 755.8	106 755.9	112 947.6	107 009.4	1.23e+03
	Feas. rules	106 755.8	112 947.2	122 607.8	110 739.1	4.28e+03
	ε-constraint	106 755.8	122 607.8	136 009.7	126 123.0	1.02e+04
	Grad-based repair	**106 755.8**	**106 755.8**	**106 755.9**	**106 755.8**	**1.67e-02**
12 (`285506.5`)	St. penalty fcn.	304 660.5	310 155.0	311 349.9	308 282.6	2.66e+03
	SR	286 826.0	308 092.0	—	313 469.9	1.54e+04
	Feas. rules	310 350.1	322 711.5	332 793.1	322 466.2	7.04e+03
	ε-constraint	305 311.9	330 042.1	370 131.6	330 407.5	1.48e+04
	Grad-based repair	**285 550.6**	**285 868.6**	**286 497.8**	**285 911.8**	**2.41e+02**
13 (`430324.5`)	St. penalty fcn.	431 403.9	455 438.4	—	452 197.8	9.27e+03
	SR	450 347.2	—	—	476 650.0	3.40e+04
	Feas. rules	457 559.5	461 764.5	479 289.1	463 844.8	5.35e+03
	ε-constraint	444 914.7	467 524.0	576 092.0	470 245.2	1.99e+04
	Grad-based repair	**430 418.2**	**431 809.4**	**444 650.7**	**432 786.3**	**2.99e+03**
14 (`546998.6`)	St. penalty fcn.	550 965.5	564 520.4	—	564 746.7	3.72e+03
	SR	—	—	—	—	—
	Feas. rules	563 508.1	574 875.2	592 240.3	574 459.7	7.68e+03
	ε-constraint	562 878.8	584 552.9	663 862.1	585 503.9	1.64e+04
	Grad-based repair	**549 496.6**	**553 317.3**	**562 335.7**	**555 301.5**	**4.53e+03**

Table 2.6: Experimental results for problems 10–14 in terms of NFEs needed to achieve convergence.

Problem	Constr-handling	Best	Median	Worst	Mean	Std	Feas. rate	Succ. rate	CPU time(s)
10	St. penalty fcn.	**41 600**	**47 300**	**59 100**	**48 120**	**3810**	**100**	**100**	**5.8**
	SR	66 000	76 900	105 000	79 904	9492	100	100	66.1
	Feas. rules	54 900	62 650	84 600	63 884	5679	100	100	7.9
	ε-constraint	68 400	85 850	200 000	95 892	35 719	100	90	12.3
	Grad-based repair	31 684	48 641	63 401	47 392	7172	100	100	61.9
11	St. penalty fcn.	43 100	48 150	58 500	48 542	3222	100	100	6.2
	SR	136 000	183 700	200 000	178 034	21 151	100	68	153.1
	Feas. rules	69 000	200 000	200 000	144 760	60 730	100	46	17.6
	ε-constraint	87 900	200 000	200 000	191 724	28 443	100	8	24.2
	Grad-based repair	**2744**	**18 337**	**32 626**	**18 142**	**9130**	**100**	**100**	**27.3**
12	St. penalty fcn.	200 000	200 000	200 000	200 000	0	100	0	84.4
	SR	200 000	200 000	200 000	200 000	0	62	0	177.2
	Feas. rules	200 000	200 000	200 000	200 000	0	100	0	72.5
	ε-constraint	200 000	200 000	200 000	200 000	0	100	0	88.1
	Grad-based repair	200 000	200 000	200 000	200 000	0	100	0	1136.7
13	St. penalty fcn.	200 000	200 000	200 000	200 000	0	96	0	72.4
	SR	200 000	200 000	200 000	200 000	0	10	0	268.5
	Feas. rules	200 000	200 000	200 000	200 000	0	100	0	67.0
	ε-constraint	200 000	200 000	200 000	200 000	0	100	0	71.1
	Grad-based repair	200 000	200 000	200 000	200 000	0	100	0	944.3
14	St. penalty fcn.	200 000	200 000	200 000	200 000	0	88	0	70.6
	SR	200 000	200 000	200 000	200 000	0	0	0	269.3
	Feas. rules	200 000	200 000	200 000	200 000	0	100	0	76.3
	ε-constraint	200 000	200 000	200 000	200 000	0	100	0	74.4
	Grad-based repair	200 000	200 000	200 000	200 000	0	100	0	994.2

Problem 12 can be viewed as a medium size instance, it contains 6 integer variables (processing stages) with 4 possible values each (the equivalent to 12 binary variables), so the problem size is equivalent to solving 4 096 NLP subproblems. Additionally, it has 16 continuous variables and 61 inequality constraints. The global optimum corresponds to an ill-conditioned point, since variations as small as 0.01% in any of the 16 continuous variables produce infeasibility. For this problem, no constraint-handling technique could obtain the reported optimal solution. However, stochastic ranking and gradient-based repair were able to locate the global optimum region: SR in 20% of the runs and 100% of the runs for gradient-based repair. It seems that once the global optimal region has been identified, new solutions generated by DE operator are very likely to be infeasible, and even if the repair process acts upon them, the direction in which constraint violation is minimized is not necessarily the same as the direction in which the objective function decreases, so that the optimization process gets very slow.

For problem 13, an additional processing stage and an additional product are considered with respect to problem 12. The performance pattern observed in the previous problem is highlighted here since, this time, 26 additional constraints need to be fulfilled and one integer variable has been added. Table 2.6 shows that no constraint-handling technique achieves the optimum in any run. However, in can be appreciated in Table 2.5 that the static penalty function and gradient-based repair found near-optimal solutions. Besides, according to the median solution obtained for each constraint-handling technique, at least half of those obtained with the gradient-based repair can be considered as good quality solutions, the objective function lying at most at 3.3 % of the optimum value.

Finally, problem 14 is the largest instance considered here. It considers 6 different products to be manufactured in 8 processing stages, the resulting problem involves 97 inequality constraints and is equivalent to solving 65 536 NLP subproblems (2^{16}). Considering the results from Table 2.5, it can be observed that SR was unable to find any single feasible solution in any run; considering both the best and median solutions obtained using the feasibility rules, it can be concluded that this technique does not allow to obtain near-optimal solutions. Besides, the performance of the ε-constraint method is quite similar to that of feasibility rules, and even more, if their mean solutions are compared, it is observed that ε-constraint worsens the performance of feasibility rules. The static penalty function presents an overall good performance, even if the global optimum was not found in any run. Nevertheless, since, according to the minimum penalty rule, the best quality solutions are obtained using the lowest possible value for the penalty factor, this also implies that infeasible solutions are more likely to be obtained, as it was the case for this problem in 12% of the executions. Again, the gradient-based repair yields the best results according to the feasibility and quality of solutions obtained (comparing for example the median solution in Table 2.5). However, due to the high number of constraints in the problem, the computation of the gradient pseudoinverse matrix is very expensive, requiring approximately twice more computational time than a classical constraint-handling technique.

As it has been pointed out above, problems 12, 13 and 14 constitute difficult problems, in particular because of the high number of variables and constraints. For these problems, not one constraint-handling technique was able to find the global optimum in any run. However, some techniques obtained good-quality solutions, which are presumably located in the global optimum region. Thus, in order to

Table 2.7: Experimental results for problems 10–14 in terms of objective function values using a local search (SQP).

Problem (`optimum`)	Constr-handling	Best	Median	Worst	Mean	Std
12 (`285506.5`)	St. penalty fcn.	285 506.5	300 301.8	310 130.8	295 823.4	9.30e+03
	SR	285 506.5	285 506.5	—	288 383.8	1.41e+04
	Feas. rules	285 506.5	304 660.0	329 222.0	308 645.8	1.03e+04
	ε-constraint	285 506.5	315 532.8	—	314 852.7	1.23e+04
	Grad-based repair	**285 506.5**	**285 506.5**	**300 804.9**	**285 812.5**	**2.16e+03**
13 (`430324.5`)	St. penalty fcn.	430 324.5	430 324.5	—	433 012.9	5.18e+03
	SR	430 324.5	430 324.5	575 159.2	443 431.2	2.64e+04
	Feas. rules	430 324.5	454 395.7	458 842.3	452 870.8	7.67e+03
	ε-constraint	430 324.5	454 395.7	466 127.0	449 351.8	1.02e+04
	Grad-based repair	**430 324.5**	**430 324.5**	**441 482.2**	**430 547.7**	**1.58e+03**
14 (`546998.6`)	St. penalty fcn.	546 998.6	547 468.5	561 584.7	551 817.9	5.74e+03
	SR	546 998.6	—	—	617 325.4	7.46e+04
	Feas. rules	558 256.9	563 403.0	576 658.5	563 573.9	4.04e+03
	ε-constraint	547 468.5	563 403.0	611 740.9	564 695.0	9.35e+03
	Grad-based repair	546 998.6	547 468.5	558 256.9	549 355.9	3.83e+03

Table 2.8: Experimental results for problems 10–14 in terms of NFEs needed to achieve convergence using a local search (SQP).

Problem	Constr-hand.	Best	Median	Worst	Mean	Std	Feas. rate	Succ. rate	SQP calls	CPU time(s)
12	St. penalty fcn.	31 165	200 000	200 000	153 508	60 053	100	42	128.5	148.0
	SR	4660	96 226	200 000	100 980	45 765	98	94	84.9	278.9
	Feas. rules	158 731	200 000	200 000	198 708	6665	100	6	162.8	198.4
	ε-constraint	135 830	200 000	200 000	198 782	9085	98	2	133.8	162.5
	Grad-based rep.	**42 565**	**60 063**	**119 588**	**63 144**	**19 097**	**100**	**98**	**6.1**	**217.0**
13	St. penalty fcn.	2140	84 187	200 123	98 076	64 780	100	78	76.0	198.8
	SR	130 502	200 000	200 000	195 854	14 052	36	10	167.0	1201.2
	Feas. rules	130 979	200 000	200 000	197 779	10 963	100	6	152.6	332.3
	ε-constraint	101 029	200 000	200 000	193 700	19 877	100	12	151.7	576.8
	Grad-based rep.	**43 794**	**63 815**	**200 000**	**75 436**	**31 327**	**100**	**98**	**8.9**	**429.3**
14	St. penalty fcn.	54 784	200 000	200 000	178 757	43 890	100	28	122.7	452.0
	SR	162 276	200 000	200 000	199 317	5346	14	2	171.1	1422.8
	Feas. rules	200 000	200 000	200 000	200 000	0	100	0	140.2	368.8
	ε-constraint	200 000	200 000	200 000	200 000	0	100	0	146.4	524.3
	Grad-based rep.	91 052	200 000	200 000	188 084	30 139	100	20	35.9	979.1

further investigate the performance of these techniques and to evaluate their ability to identify sub-optimal solutions lying in the region of the global optimum, the use of an additional local search is explored. The local optimizer Successive Quadratic Programming (SQP) is applied with a probability $0.1/\mu$ to each individual, where μ is the population size, in this way, one individual in the population is improved, on average, every 10 generations. For the ε-constraint and gradient-based repair, this local search procedure is carried out only after the ε level is equal to zero, i.e., once the algorithm has presumably identified a promising feasible region. The obtained results are displayed in Tables 2.7 and 2.8. The NFEs reported in Table 2.8 take into account the evaluations of the objective function performed by both DE and SQP. The column "SQP calls" represents the average number of individuals on which the local search is carried out before the algorithm stops. For example in problem 12, using the gradient-based repair, the SQP solver was applied on average to 6.1 individuals before finding the global optimum, whereas for feasibility rules, SQP was called on average 162.8 times before the algorithm attains the maximum NFEs (more likely than finding the optimum, from Table 2.8).

Outstanding results for problem 12 are obtained by the stochastic ranking and gradient-based repair strategies, obtaining success rates of 94% and 100%, respectively. Conversely, a poor performance considering the success rate is still observed for static penalty, feasibility rules and ε-constraint methods. These results are in agreement with those of Table 2.5 where no local search is employed, that is, the local search is beneficial only if the global region has been identified. This feature is also observed for problems 13 and 14. For problem 13, those constraint-handling techniques that showed a poor performance before the use of SQP (SR, feasibility rules and ε-constraint) present a similar trend, with a significant additional CPU time, due to the use of the local optimizer. In addition, the static penalty function obtains an acceptable success rate (78%) and gradient-based repair solved to optimality this problem in all runs except one. On the contrary, problem 14 still constitutes a difficult problem, even if local search is applied. The high number of binary variables suggests that using a higher number of function evaluations might be necessary to solve this problem to optimality.

With respect to the use of the local search, it can be concluded that it shows indisputable advantages when the population-based algorithm succeeds to find promising regions. Otherwise, it is probable that no significant improvements could be observed. Further, the computational costs associated to the use of a local search must also be taken into account.

Summarizing, the above empirical study highlighted the importance of using an efficient constraint-handling technique when solving single-objective PSE optimization problems, which are in general highly constrained problems. The numerical results obtained point out that gradient-based repair method is the most robust and thus promising method among the techniques studied here.

Additional example: Biobjective case study

This problem, presented in Rangaiah (2009), considers the maximization of two objectives: the profit before taxes (PBT) and the net present value (NPW) for the Williams & Otto process problem. Here, the NSGA-II algorithm coupled with gradient-based repair as constraint-handling technique is used as a solution technique. Also, for comparison purposes, the constraint-handling strategy presented in Rangaiah

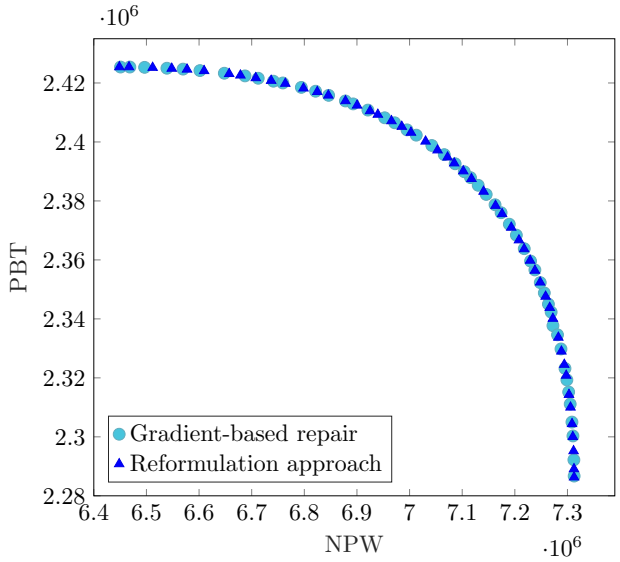


Figure 2.1: Final Pareto front approximations of the biobjective Williams-Otto problem using NSGA-II with gradient-based repair method and the reformulation approach.

(ibid.), in which all the equality constraints are eliminated by means of solving a system of nonlinear equations, is implemented here. The obtained approximations to the Pareto front of this problem are presented in Figure 2.1.

The non-dominated solutions are obtained in one single run, unlike mathematical programming techniques in which multiple runs are needed to produce an approximation of the Pareto front. The importance of an efficient constraint-handling technique is also to be highlighted: no other constraint-handling technique studied in this work was able to find a non-dominated solution in the real Pareto front, actually no feasible solutions could be found except with the gradient-based repair procedure. Besides, the reformulation approach proposed in Rangaiah (ibid.) was time-consuming, taking approximately 45 seconds per run (since the solution of the system of nonlinear equations has to be performed for every evaluation of the objective function) while, in contrast, the gradient-based repair approach takes approximately 1.5 seconds per run. Also, it is worth mentioning that the Pareto front approximations obtained by both approaches are comparable, i.e., no approach visually outperforms the other.

2.7 Conclusions

In this chapter, the performance of several constraint-handling techniques for EAs has been compared for the solution of a set of 14 benchmark problems from the PSE area. The empirical analysis conducted showed that the results' quality greatly depends on the constraint-handling technique used for the solution of problems with high number of constraints or binary variables.

The analysis of the dedicated literature has shown that the most widely used approach within EAs considers the reformulation of the model and the use of static penalty functions or feasibility rules as constraint-handling techniques. However, the results obtained here highlighted that the performance of this strategy, though acceptable in some cases, proved to be poor in others. Besides, among the constraint-handling techniques considered in this study, the gradient-based repair method

deserves a special attention, as this constraint-handling technique was the only one that can find the global optimum region in almost all the test problems. Coupled with the ε-constraint method, the search algorithm promotes the exploration of promising regions over the entire search space instead of getting trapped into a local optimum. It is worth emphasizing that, even if this method needs additional information (computation of constraints' gradient), its excellent results both in terms of computational time and solution quality encourage its use. In addition, the use of gradient-based repair method in highly constrained mixed-integer problems seems to be not only adequate, but necessary in order to obtain satisfactory results. Finally, this work highlighted the unquestionable benefits obtained using this constraint-handling method, usually under-estimated in the devoted literature. Therefore, these conclusions allow reconsidering evolutionary algorithms as a serious approach for solving highly-constrained real-world optimization problems.

Besides, the good performance exhibited in the solution of the biobjective case study allows to consider the solution of bigger instances of PSE multiobjective problems. As the gradient-based repair method can be coupled with any multiobjective evolutionary algorithm (MOEA), the solution of multiobjective MINLP problems related to PSE using more sophisticated MOEAs, is the subject of the next chapters.

Chapter 3

Constraint-handling in a multiobjective framework

Contents

3.1 Introduction

In this chapter, we study the effect of repairing infeasible solutions using the gradient information, that is, the most promising constraint-handling technique investigated in the previous chapter, in a multiobjective framework. As mentioned before, the need for studying diverse constraint-handling mechanisms in MOEAs arises when considering real-world process engineering problems, which usually contain a significant number of active constraints, involving an important quantity of variables. In

The content of this chapter has been published in the form of chapter in a book. Victor H Cantú, Antonin Ponsich, and Catherine Azzaro-Pantel (Apr. 2021). "Constraint Handling in Metaheuristics and Applications". In: ed. by Anand J Kulkarni, Efrén Mezura-Montes, Yong Wang, Amir H Gandomi, and Ganesh Krishnasamy. 1st ed. Springer Singapore. Chap. On the use of gradient-based repair method for solving constrained multiobjective optimization problems – A comparative study, pp. 119–149. DOI: 10.1007/978-981-33-6710-4

their canonical form, MOEAs lack of sophisticated mechanisms for evolving properly a population over constrained search spaces. As a consequence, unless appropriate constraint-handling mechanisms are used, the applicability of novel MOEAs into PSE area might be lessen to only a narrow range of problems, those containing reduced constrained search spaces.

In this perspective, in this chapter the gradient-based repair method is embedded in six classical constraint-handling techniques: constraint dominance principle, adaptive threshold penalty, C-MOEA/D, stochastic ranking, ε-constraint and improved ε-constraint. The performance of these techniques is explored on NSGA-II and MOEA/D algorithms, and the test functions include classical benchmark problems with inequality constraints, as well as recent problems with equality constraints.

This chapter is organized as follows. Section 3.2 presents a literature review on constrained multiobjective optimization. Section 3.3 develops and briefly discusses the six constraint-handling techniques studied in this work. The experimental methodology and the computational results are described in Sections 3.4 and 3.5, respectively. Finally, conclusions and perspectives for future work are drawn in Section 3.6.

3.2 Related work

In the last decade, the development of constraint-handling strategies for the solution of CMOPs has drawn considerable interest from the evolutionary multiobjective optimization (EMO) community. It must be emphasized that, commonly, most MOEAs are adapted for treating CMOPs through the constraint-dominance principle (CDP, which is an extension to multiobjective optimization of Deb's feasibility rules, based on the dominance operator when comparing two feasible solutions; see next section for a detailed definition and explanations) (Deb et al., 2002). However, the drawbacks of this strategy, well-known for single-objective optimization, are also evident in multiobjective optimization, in particular because diversity preservation among solutions is a critical issue in order to achieve a good approximation to the true Pareto front.

Consequently, alternative constraint-handling strategies adapted to the multiobjective framework have been developed in the two last decades. For instance, with the purpose of tackling the drawbacks of CDP for NSGA-II, the Infeasibility Driven Evolutionary Algorithm (IDEA) (Singh et al., 2009) proposes the use of a parameter α standing for the ratio of infeasible solutions to survive in the population. A constraint violation function is used as an additional objective and then, non-dominated ranking is applied to infeasible and feasible individuals separately. In this way, the new population (containing μ individuals) has, at most, $\alpha \cdot \mu$ best-ranked infeasible solutions. Besides, in a dominance-based framework too, a parameterless adaptive penalty function has been proposed in Woldesenbet et al. (2009), in which a modified objective function is computed to be used in the nondomination sorting so that the search is guided towards feasible regions first, and then, infeasible individuals information is used to explore promising infeasible regions.

In addition, in Jan and Khanum (2013), the authors propose adaptations of the CDP and Stochastic Ranking (SR) methods for working in MOEA/D. The proposed SR-based multiobjective version selects a solution according to the utility function value with probability p_f, otherwise the constraint violation is used as a comparison criterion. Results show that MOEA/D-CDP consistenly outperforms MOEA/D-SR for

several test functions. Another decomposition-based algorithm adapted for solving CMOPs is proposed in Asafuddoula et al. (2012), by means of a function that combines the overall constraint violation and the number of active constraints. The mean value of this function, weighted by the ratio of feasible individuals in the population, allows computing a threshold on the allowed constraint violation. Solutions below this threshold are considered as feasible and compared in terms of their utility function.

In Zapotecas-Martinez and Coello Coello (2014), the ε-constraint method (Takahama and Sakai, 2005) within MOEA/D is explored, in which the ε level is defined using a normalized constraint violation and an additional ε-comparison rule is proposed accounting for slightly-infeasible promising solutions. This approach was extended in Zapotecas-Martínez and Ponsich (2020) by using an additional scalarizing function to determine the balance between the constraint violation and the objective values, represented within the MOEA/D scalarizing function. Another adaptation of the ε-constraint method is introduced in Fan et al. (2019b), where the original function defining the ε level is modified, so that this latter increases or decreases depending on the ratio of feasible solutions in the population, in order to strengthen the search in both feasible and infeasible regions throughout the optimization.

More recently, a two-stage procedure (called, Push and Pull) was proposed in (Fan et al., 2019c). In the first stage (push stage), the algorithm explores the unconstrained search space, which allows the population to get across infeasible regions, and then, in the pull stage, original constraints are considered along with the improved ε-constraint to gradually pull the population towards feasible regions. The stagnation of the identified ideal and nadir points determines the switching mechanism between the "push" and "pull" phases. Nevertheless, this constraint-handling method has been tested for a problem test suite (LIRCMOP, Fan et al., 2019b) that contains only inequality constraints and some true Pareto fronts are identical to their counterparts in the unconstrained problems.

In Fan et al. (2019a), diversity preservation in the population is explicitly handled through a modification of MOEA/D-CDP. When two solutions are compared, if at least one of them is infeasible, a similarity measure is computed based on the angle of their corresponding objective vectors with respect to the ideal point. Similar solutions are compared according to their overall constraint violation, otherwise the scalarizing function value is used as the comparison criterion. In Peng et al. (2017), the original CMOP is modified considering the overall constraint violation as an additional objective. Two weight vector sets are generated, accounting for infeasible and feasible solutions, respectively, in order that infeasible individuals are well distributed along the Pareto front and may lead the search to promising regions. In Yang et al. (2020), authors use a modified ε-constraint strategy in a decomposition-based framework. The feasibility ratio and the minimum constraint violation value of the current population are employed in order to compute the ε level at each generation. Also, the scaling factor F within DE is adjusted dynamically in order to promote local search in late generations.

Finally, MOEA/D is modified in Ishibuchi et al. (2018) so that two solutions (one feasible and one infeasible) are assigned to each weight vector. This strategy is used to consider an individual on each side of the feasibility boundary. During the selection phase, an offspring solution is compared with two solutions of each neighboring weight vector, considering the scalarizing function and the overall constraint violation. For the bi-objective case, dominance is used to select two surviving individuals among the three (i.e., one offspring and two existing solutions for each weight vector). When

all three solutions are non-dominated with respect to their scalarizing function values, the solution with high constraint value is discarded.

In this study, six representative constraint-handling techniques are to be used. These techniques are chosen because they are easy-to-implement and their operating mode is relatively straightforward, which explains that they have been widely used in the devoted specialized literature and that they may be qualified as state-of-the-art methods.

3.3 A portfolio of constraint-handling strategies

The selected constraint-handling techniques are presented and discussed in detail in the following. Note that even if some of these methods have been originally proposed to tackle only problems with inequality constraints, they can be easily adjusted for considering equality constraints as well (please note that each equality constraint can be converted into two inequalities by using a small tolerance on equality satisfaction).

3.3.1 Constraint dominance principle

This constraint handling technique establishes the superiority of feasible solutions over infeasible ones. Proposed by Deb (2000), the feasibility rules consist in a binary tournament selection according to the three criteria presented in the previous chapter. Their extension to CMOPs, called the constraint dominance principle (CDP), reformulates condition 2 by a dominance-based comparison, and thus is stated as "among two feasible solutions, that which dominates the other is preferred", and a diversity operator is used if no solution dominates the other. Note that, for decomposition-based MOEAs, the utility function is used for condition 2.

Due to its simplicity and its overall good performance, the CDP is usually the first constraint-handling technique tested for treating a given problem with MOEAs. However, the main drawback of this method appears when dealing with problems with a reduced and disconnected feasible region: since any feasible solution is preferred over an infeasible one, once the algorithm has converged to a feasible region, it might be difficult to escape from there to explore the rest of the search space, i.e., once the constraints are fulfilled, the algorithm is likely to get trapped prematurely in some subregion of the search space. Besides, in the multiobjective context, this aspect may result in a bad distribution of the obtained approximation set if highly-constrained problems are tackled, that is, solutions populate only some parts of the true PF and thus the optimization goals respecting diversity cannot be achieved.

3.3.2 Adaptive threshold penalty

The adaptive threshold penalty function (ATP) (Jan and Zhang, 2010) is particularly adapted to be used within MOEA/D. This penalty function uses a threshold value, τ, for dynamically controlling the amount of penalty. The threshold value τ is defined as:

$$\tau = V_{min} + 0.3(V_{max} - V_{min}) \tag{3.1}$$

with $V_{min} = \min\{\phi(\mathbf{x}^{(i)})\}, \forall i \in T$ and $V_{max} = \max\{\phi(\mathbf{x}^{(i)})\}, \forall i \in T$, where T represents a given neighborhood in MOEA/D. Then, according to ATP method, the fitness function F of a solution $\mathbf{x}$ is defined as:

$$F(\mathbf{f}'(\mathbf{x}); \mathbf{w}) = \begin{cases} u(\mathbf{f}'(\mathbf{x}); \mathbf{w}) + s_1\phi^2(\mathbf{x}), & \text{if } \phi(\mathbf{x}) < \tau, \\ u(\mathbf{f}'(\mathbf{x}); \mathbf{w}) + s_1\tau^2 + s_2(\phi(\mathbf{x}) - \tau), & \text{otherwise} \end{cases} \tag{3.2}$$

where s_1 and s_2 are two scaling parameters. The authors obtained good results by setting $s_1 = 0.01$ and $s_2 = 20$.

In this work, when the ATP technique is implemented within a dominance-based MOEA (for instance, NSGA-II), every objective function is augmented by the above-described penalty terms, before the ranking procedure. In this case, P represents the whole population.

3.3.3 C-MOEA/D

This constraint-handling method separates the objective function and the violation of constraints, and proposes a violation threshold allowing for a relaxation of constraints under which solutions are considered as feasible (Asafuddoula et al., 2012). Once the relaxation is carried out and feasible and infeasible solutions have been identified accordingly, the constraint dominance principle is employed for the selection step. The violation threshold φ proposed by the authors is calculated as follows:

$$\varphi = \frac{\sum_{i=1}^{|P_t|} \phi(\mathbf{x}^{(i)})}{|P_t|} \cdot I_t(P_t). \tag{3.3}$$

where the first term denotes the average overall constraint violation over the current population P_t and I_t is the feasibility ratio (i.e. the proportion of feasible solutions in the population) at generation t of the evolutionary algorithm. Although it was introduced within the MOEA/D framework, this constraint-handling technique can be easily adapted within MOEAs using different paradigms different from the decomposition one (in this work, its implementation within NSGA-II is simply labeled as C-NSGA-II).

3.3.4 Stochastic ranking

Stochastic ranking (SR) (Runarsson and Yao, 2000) was proposed as an attempt to balance the relative weights of the objective and the constraint violation that occurs in penalty functions. In this method, the population is sorted following a probabilistic procedure (see Subsection 2.4.2 for further details). Since it was designed for a single-objective framework, its generalization to a multiobjective problem is not straightforward, even when aggregation functions are considered as in decomposition-based algorithms. One attempt of using SR for multiobjective optimization is introduced in Jan and Khanum (2013) for a decomposition-based algorithm (MOEA/D). In the present study, the same implementation is employed. Nevertheless, it must be emphasized that no actual *ranking* of the population is performed, but instead a stochastic comparison between the parent and the offspring: the comparison is performed, first, according to their respective scalarizing function

with a probability p_f, otherwise, the constraint dominance principle is used. Besides, please note that, in this work, SR is only explored with MOEA/D, since it cannot be implemented within NSGA-II because of its working mode (individual ranking is based on a dominance sorting procedure, which cannot be biased by constraint violation).

3.3.5 ε-constraint method

This method integrates a relaxation of constraints up to a so-called ε level, under which solutions are regarded as feasible (Takahama and Sakai, 2005). Its implementation in multiobjective optimization differs from that presented in the previous chapter only by the fact that the CDP is used instead of the feasibility rules. In order that the reader may easily compare this method with the one presented next, the dynamic control of ε level is presented here again, it is computed as:

$$\begin{aligned} \varepsilon(0) &= \phi(\mathbf{x}^\theta) \\ \varepsilon(t) &= \begin{cases} \varepsilon(0)(1-\frac{t}{T_c})^{cp}, & 0 < t < T_c, \\ 0, & t \geq T_c \end{cases} \end{aligned} \tag{3.4}$$

where $\mathbf{x}^\theta$ is the best θ-th individual in terms of constraint violation in the first generation, cp is a parameter to control the decreasing speed of the ε level and T_c represents the generation after which the ε level is set to 0, after which CDP is considered.

3.3.6 Improved ε-constraint method

In Fan et al. (2019b), another function for controlling the ε level is proposed, aiming to fix some drawbacks observed in the canonical ε-constraint method. Namely, the fact that the dynamic control of the ε level is a monotonically decreasing function, which prohibits the exploration of promising disconnected regions once the ε takes small values. Besides, in problems with large feasible regions, all population might be feasible in early generations and thus $\varepsilon(0)$ becomes zero prematurely. This would be equivalent to using CDP (with its drawbacks) all along the evolutionary process. Therefore, the proposed function for ε level permits increasing its value if the feasible ratio is above a given threshold (α parameter), according to:

$$\begin{aligned} \varepsilon(0) &= \phi(\mathbf{x}^\theta) \\ \varepsilon(t) &= \begin{cases} \varepsilon(t-1)(1-\rho), & \text{if } I_t(P_t) < \alpha \text{ and } t < T_c, \\ \phi_{max}(1+\rho), & \text{if } I_t(P_t) \geq \alpha \text{ and } t < T_c, \\ 0, & t \geq T_c \end{cases} \end{aligned} \tag{3.5}$$

where $\mathbf{x}^\theta$ has the same meaning as in equation (3.4), $I_t(P_t)$ is the ratio of feasible individuals of the population at generation t, parameter ρ is to control the speed of reducing the ε level (it ranges between 0 and 1), parameter α controls the searching preference between the feasible and infeasible regions and ϕ_{max} is the maximum overall constraint violation found so far.

3.4 Experimental methodology

3.4.1 Test Problems

We investigate the performance of the six above-mentioned constraint-handling techniques, and particularly, the effect of embedding the gradient-based repair method within each of these techniques. For this purpose, we carried out a comprehensive study over 35 test problems which include inequality constraints: ten constrained function (CF) test problems (Zhang et al., 2008b), fourteen CMOPs with large infeasible regions (LIRCMOPs) (Fan et al., 2019b) and eleven problems with equality constraints (Cuate et al., 2020a,b; Das and Dennis, 1998; Rangaiah, 2009).

The reason for the choice of CF and LIRCMOP test suites is related to the interesting features they present (numerous local optima, disconnected Pareto fronts and inequality constraints that are difficult to satisfy). Regarding the problems containing equality constraints, they are rarely found in the evolutionary computation literature: first, because the corresponding problems have been proposed quite recently and, on the other hand, because they entail solution difficulties for classical constraint-handling techniques, according to the preliminary experiments carried out by the authors.

3.4.2 Performance indicators

To assess the performance of the different algorithms, we use the inverted generational distance indicator (IGD) (Bosman and Thierens, 2003) as well as the hypervolume indicator (HV) (Zitzler and Thiele, 1998). As mentioned in Chapter 1, the IGD indicates how far the discretized Pareto optimal front is from the approximation set. This non-Pareto compliant indicator measures both convergence and diversity. A smaller value of IGD indicates a better performance of the algorithm. To generate the reference set for CF, LIRCMOP, Eq-DTLZ and Eq-IDTLZ test suites, 1 000 points are sampled uniformly from the true PF for two-objective problems; whereas 10 000 points are sampled uniformly from the true PF for three-objective problems. With respect to the EQC test suite, the number of points sampled from the true PF varies for each problem, depending on the data available.

Concerning the HV, it is the only performance indicator known to be Pareto-compliant. A large HV value shows that a given solution set approximates the Pareto optimal front well in terms of both convergence and diversity in the objective space. The k-dimensional reference vector for the HV computation is set to $[1.1, 1.1, \ldots, 1.1]^T$ in the normalized objective space $[0, 1]^k$, for all problems. To obtain the ideal and nadir points for problem EQC6, five independent runs using jDE with ε-constraint method coupled with gradient-based repair were performed for each single-objective problem, setting the number of function evaluations to 500 000, in order to obtain the extreme points.

3.4.3 Experimental settings

The detailed parameter settings for the experiments carried out here are summarized below.

1. MOEA/D parameters. Scalarizing function: augmented achievement (AASF), probability of choosing parents locally: $\delta = 0.9$, neighborhood size: $T = 0.1\mu$,

maximum number of replacements $n_r = 2$. The crowding distance (from NSGA-II) is used if the external archive exceeds a predefined size limit (the population size).

2. Parameters of the evolutionary variation operators:

   - DE parameters for MOEA/D. $CR = 1, F = 0.5$. Except for Eq-DTLZ and Eq-IDTLZ, where DE parameters are that used on the original reference, i.e., $CR = 0.2, F = 0.2$.
   - SBX parameters for NSGA-II. $p_c = 1, \eta_c = 20$.

   Both algorithms also use a polynomial mutation with parameters: $p_m = 1/n, \eta_m = 20$.

3. Population size (μ). For all problems with three objectives, $\mu = 300$. For two-objective problems: for the CF test suite, $\mu = 200$; for the LIRCMOP test suite, $\mu = 200$; for the EQC test suite, $\mu = 100$.

4. Number of function evaluations (NFE). For the CF test suite, $NFE =$ 1e5 for two-objective problems, $NFE =$ 1.5e5 for three-objective problems. For the LIRCMOP test suite, $NFE =$ 1.5e5. For the EQC problems suite, $NFE =$ 1e5 for two-objective functions, $NFE =$ 1.5e5 for EQC4 and $NFE =$ 0.5e5 for EQC5. For Eq-DTLZ and Eq-IDTLZ, $NFE =$ 1.5e5. Concerning the gradient-based repair method, each computation of the gradient of constraints counts as 1 function evaluation.

5. Parameter settings of constraint handling techniques: they were not tuned, but rather set to the values proposed in the original articles:

   – ATP: $s_1 = 0.01, s_2 = 20$.

   – Stochastic ranking: $p_f = 0.05$.

   – ε-constraint: $\theta = 0.2\mu, cp = 5, T_c = 0.2T_{max}$.

   – Improved ε-constraint: $\theta = 0.05\mu, \rho = 0.1, \alpha = 0.95, T_c = 0.8T_{max}$.

   – Gradient-based repair: $P_g = 1, R_g = 3$, step size for finite differences: 10^{-6}.

For each problem, 31 independent runs were performed (an odd number is chosen in order to be able to present median results). The algorithms previously presented were implemented in MATLAB R2019a and the computational experiments were performed with a processor Intel Xeon E3-1505M v6 at 3.00 GHz and 32 Go RAM.

3.5 Results and discussion

The results obtained with the six above-mentioned constraint handling techniques (CDP-MOEA/D, ATP-MOEA/D, C-MOEA/D, SR-MOEA/D, ε-MOEA/D and Improved ε-MOEA/D), without and with the gradient-based repair, are presented and analyzed in Sections 3.5.1, 3.5.2 and 3.5.3 for the CF functions, the LIRCMOP test suite and the equality-constrained problems, respectively. In particular, in Tables 3.1 to 3.12, the mean and standard deviation values of each considered performance indicator are provided for both the MOEAs employed here (MOEA/D and NSGA-II). Also, the

overall computational time (in seconds) for performing the 31 runs is displayed in each table presenting the IGD values.

For each instance and constraint handling technique, binary comparisons are carried out in order to highlight the effect of repairing infeasible solutions using the gradient information. In this perspective, a Wilcoxon rank sum test (with a significance level of 95%) is performed for each comparison, using the results for the considered indicators over the 31 executions. When the null hypothesis is rejected (i.e., there is no statistical difference between the indicator values obtained with and without the gradient-based repair), significantly better results are represented in boldface. At the bottom of each table, a summary of these statistical tests is displayed, where I, D and S respectively represent the number of instances for which the gradient-based repair process achieves statistically inferior (I), equivalent (no significant difference, D) or statistically superior (S) results when compared with the original technique without repair.

Finally, the results regarding the feasibility ratio indicator of the current population at the last generation (I_F) (i.e., different from the external archive) are presented in Tables 3.11 and 3.12 for problems containing equality constraints. I_F is presented only for these problems because some canonical algorithms experience difficulties for obtaining feasible solutions when tackling problems with equality constraints, which is not the case for the other test functions. Note that no statistical tests are computed for this indicator, as a high value of this indicator does not necessarily mean a good approximation to $\mathcal{PF}$. In all figures, the Pareto front approximations shown correspond to the median run with respect to the hypervolume indicator.

3.5.1 CF test problems

Tables 3.1 to 3.4 show that the overall performance of the gradient-based repair is satisfying for the CF test problems, i.e., globally, combining it with another constraint-handling technique produces at least as good solutions as the canonical method. More precisely, considering the statistical results for MOEA/D presented in Tables 3.1 and 3.3 as a whole (combining the IGD and HV results), it can be stated that the use of the gradient-based repair significantly improves the canonical algorithm in 37.5% of the instances, and significantly deteriorates the canonical algorithm only in 5.8% of the cases when using MOEA/D. Regarding the use of the repair method within NSGA-II, (Tables 3.2 and 3.4), it improves significantly the original techniques in 43% of the instances, while a significant deterioration is observed for only one test problem (CF10), which may suggest that this is an unusual case.

Besides, with respect to the computational times, it can be observed from Tables 3.1 and 3.2 that no additional cost is to be paid for using this gradient-based repair. Actually, for many instances, using the gradient-based repair leads to shorter CPU times. So, the computation of the gradient pseudoinverse, which is *a priori* time-consuming, is offset by the fact that less generations are needed (since the algorithm termination condition is the number of function evaluations). Consequently, it is noteworthy that when repairing infeasible solutions, the evolutionary process is somewhat accelerated, in regards to the number of generations needed to converge to $\mathcal{PF}$.

However, it should be emphasized that most CF functions constitute difficult problems for all the studied techniques, even when the gradient-based repair is included. The reason of these difficulties arise not only from the constraints but also

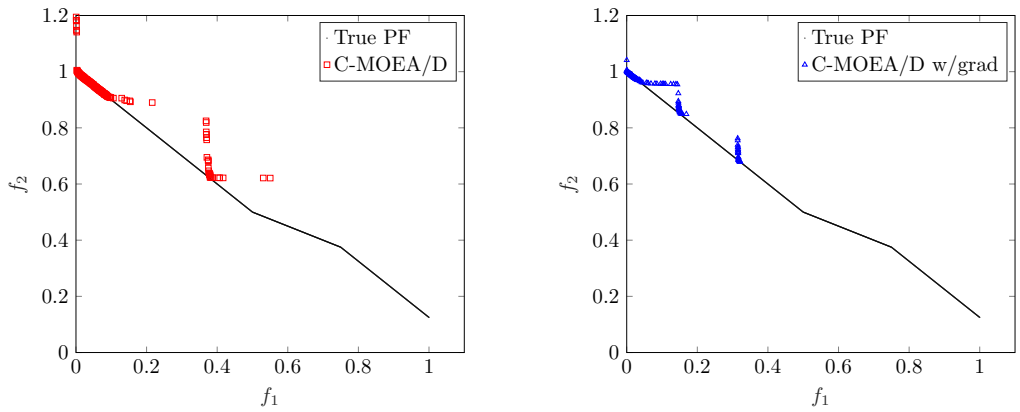


Figure 3.1: Final Pareto front approximations for CF5 function. C-MOEA/D without (left) and with (right) gradient-based repair.

Table 3.1: IGD values and CPU time (in seconds) obtained on CF test problems with MOEA/D.

		CDP-MOEA/D		ATP-MOEA/D		C-MOEA/D		SR-MOEA/D		ε-MOEA/D		Imp ε-MOEA/D	
		—	w/grad	—	w/grad	—	w/grad	—	w/grad	—	w/grad	—	w/grad
CF1	mean	5.68e-03	**2.34e-05**	5.68e-03	**1.89e-05**	4.99e-04	**1.99e-05**	6.94e-03	**2.66e-05**	5.68e-03	**2.34e-05**	5.68e-03	**2.61e-04**
	std	1.39e-03	3.12e-05	1.89e-03	3.39e-05	3.68e-04	3.35e-05	1.52e-03	2.77e-05	1.39e-03	3.12e-05	1.39e-03	8.31e-05
	CPU time	4.78e+01	4.25e+01	5.17e+01	3.73e+01	4.67e+01	4.44e+01	4.96e+01	3.50e+01	4.85e+01	4.23e+01	4.60e+01	3.98e+01
CF2	mean	4.10e-03	4.53e-03	4.24e-03	4.14e-03	5.71e-03	**3.88e-03**	7.17e-03	8.93e-03	4.10e-03	4.53e-03	3.76e-03	4.37e-03
	std	2.86e-03	3.29e-03	2.64e-03	2.54e-03	3.63e-03	2.07e-03	1.19e-02	1.78e-02	2.86e-03	3.29e-03	2.68e-03	2.22e-03
	CPU time	5.22e+01	6.36e+01	5.18e+01	5.13e+01	5.02e+01	6.77e+01	4.87e+01	4.57e+01	4.68e+01	6.22e+01	4.81e+01	6.00e+01
CF3	mean	2.22e-01	2.07e-01	2.14e-01	2.09e-01	2.07e-01	2.05e-01	2.32e-01	2.46e-01	2.22e-01	2.07e-01	2.03e-01	2.06e-01
	std	1.05e-01	9.29e-02	7.88e-02	8.79e-02	8.77e-02	9.46e-02	1.20e-01	1.18e-01	1.05e-01	9.29e-02	8.73e-02	8.91e-02
	CPU time	4.45e+01	5.97e+01	5.11e+01	4.45e+01	4.84e+01	5.96e+01	4.88e+01	4.31e+01	4.84e+01	6.46e+01	4.54e+01	6.44e+01
CF4	mean	5.31e-02	4.58e-02	5.27e-02	4.60e-02	6.73e-02	4.51e-02	4.69e-02	4.78e-02	5.31e-02	4.58e-02	6.01e-02	4.72e-02
	std	3.22e-02	3.18e-02	2.83e-02	1.60e-02	5.39e-02	1.96e-02	2.78e-02	1.99e-02	3.22e-02	3.18e-02	4.27e-02	2.79e-02
	CPU time	4.99e+01	5.40e+01	4.97e+01	4.32e+01	4.70e+01	5.94e+01	4.69e+01	3.99e+01	4.63e+01	5.49e+01	4.81e+01	5.28e+01
CF5	mean	2.64e-01	2.59e-01	2.27e-01	2.14e-01	2.50e-01	2.70e-01	2.59e-01	2.10e-01	2.64e-01	2.59e-01	2.56e-01	2.42e-01
	std	9.77e-02	1.23e-01	8.54e-02	9.86e-02	1.12e-01	1.17e-01	1.11e-01	9.16e-02	9.77e-02	1.23e-01	1.02e-01	1.23e-01
	CPU time	4.62e+01	6.15e+01	5.29e+01	4.81e+01	4.56e+01	5.86e+01	4.68e+01	3.61e+01	4.54e+01	5.64e+01	4.92e+01	5.29e+01
CF6	mean	1.07e-01	**7.42e-02**	6.26e-02	**3.53e-02**	1.13e-01	**5.02e-02**	1.16e-01	**7.71e-02**	1.07e-01	**7.42e-02**	9.92e-02	**5.95e-02**
	std	3.39e-02	4.20e-02	3.99e-02	2.97e-02	3.68e-02	3.21e-02	2.64e-02	4.60e-02	3.39e-02	4.20e-02	3.78e-02	4.85e-02
	CPU time	4.80e+01	6.24e+01	5.24e+01	4.43e+01	5.53e+01	6.37e+01	5.27e+01	4.17e+01	5.29e+01	6.09e+01	5.38e+01	5.84e+01
CF7	mean	2.49e-01	2.38e-01	2.24e-01	1.99e-01	2.51e-01	**2.00e-01**	2.80e-01	**2.08e-01**	2.49e-01	2.38e-01	2.68e-01	**2.28e-01**
	std	1.11e-01	1.26e-01	9.42e-02	8.72e-02	9.89e-02	9.55e-02	1.29e-01	8.49e-02	1.11e-01	1.26e-01	1.09e-01	1.13e-01
	CPU time	4.87e+01	5.90e+01	4.71e+01	7.28e+01	4.49e+01	7.01e+01	4.45e+01	4.64e+01	4.55e+01	6.33e+01	5.00e+01	6.29e+01
CF8	mean	5.59e+08	**3.77e-01**	**1.06e-01**	1.13e-01	5.59e+08	**1.35e-01**	4.27e-01	**3.81e-01**	6.74e-01	**3.65e-01**	4.64e-01	**2.23e-01**
	std	3.11e+09	7.33e-02	2.50e-02	1.39e-02	3.11e+09	2.73e-02	4.97e-02	5.96e-02	1.06e+00	6.68e-02	9.15e-02	1.20e-01
	CPU time	8.38e+01	5.56e+01	1.17e+02	7.28e+01	7.66e+01	6.23e+01	7.40e+01	4.98e+01	8.44e+01	5.42e+01	8.54e+01	6.32e+01
CF9	mean	5.41e-02	5.33e-02	**4.66e-02**	5.33e-02	6.05e-02	**5.50e-02**	6.38e-02	**5.71e-02**	5.35e-02	5.33e-02	5.41e-02	5.33e-02
	std	6.26e-03	4.85e-03	3.78e-03	6.13e-03	6.20e-03	6.38e-03	8.89e-03	4.58e-03	6.29e-03	4.85e-03	6.26e-03	4.85e-03
	CPU time	1.47e+02	9.31e+01	1.43e+02	9.15e+01	1.35e+02	8.90e+01	1.37e+02	8.76e+01	1.45e+02	9.84e+01	1.44e+02	1.02e+02
CF10	mean	3.71e+09	**1.33e+09**	3.04e-01	5.00e-01	3.71e+09	**1.33e+09**	3.85e+09	1.99e+09	2.12e+09	**2.94e+00**	3.18e+09	1.19e+09
	std	1.24e+09	1.95e+09	1.48e-01	8.67e-01	1.24e+09	1.95e+09	1.03e+09	2.09e+09	2.09e+09	1.11e+01	1.75e+09	1.90e+09
	CPU time	7.31e+01	4.38e+01	9.41e+01	5.64e+01	7.07e+01	3.93e+01	7.78e+01	4.15e+01	8.18e+01	4.62e+01	7.18e+01	4.32e+01
Wilc. test (I-D-S)		0–6–4		2–6–2		0–3–7		0–5–5		0–6–4		0–6–4	

Table 3.2: IGD values and CPU time (in seconds) obtained on CF test problems with NSGA-II.

		CDP-NSGA-II		ATP-NSGA-II		C-NSGA-II		ε-NSGA-II		Imp ε-NSGA-II	
		—	w/grad	—	w/grad	—	w/grad	—	w/grad	—	w/grad
CF1	mean	2.20e-02	**6.65e-04**	1.97e-01	**2.77e-03**	6.24e-02	**1.01e-03**	2.20e-02	**6.65e-04**	2.20e-02	**2.64e-03**
	std	2.91e-03	1.32e-04	3.97e-02	1.39e-03	3.00e-02	1.02e-03	2.91e-03	1.32e-04	2.91e-03	1.91e-03
	CPU time	1.21e+01	2.03e+01	9.30e+00	1.97e+01	9.81e+00	1.84e+01	8.74e+00	1.87e+01	8.48e+00	1.91e+01
CF2	mean	4.74e-02	3.98e-02	4.66e-02	4.26e-02	4.31e-02	4.25e-02	4.74e-02	3.98e-02	4.14e-02	4.21e-02
	std	2.00e-02	1.76e-02	1.55e-02	1.62e-02	1.64e-02	1.81e-02	2.00e-02	1.76e-02	1.43e-02	1.31e-02
	CPU time	9.24e+00	1.57e+01	8.56e+00	1.52e+01	9.02e+00	1.49e+01	9.80e+00	1.46e+01	1.00e+01	1.52e+01
CF3	mean	2.53e-01	2.50e-01	2.62e-01	2.62e-01	2.44e-01	2.61e-01	2.53e-01	2.50e-01	2.71e-01	2.66e-01
	std	9.82e-02	7.84e-02	9.60e-02	1.04e-01	8.48e-02	9.14e-02	9.82e-02	7.84e-02	9.16e-02	1.07e-01
	CPU time	8.68e+00	1.41e+01	1.12e+01	1.42e+01	8.95e+00	1.32e+01	8.55e+00	1.34e+01	9.39e+00	1.34e+01
CF4	mean	9.30e-02	7.49e-02	1.21e-01	**7.19e-02**	1.07e-01	**8.01e-02**	9.30e-02	7.49e-02	1.07e-01	**7.70e-02**
	std	4.02e-02	1.33e-02	3.76e-02	1.36e-02	3.02e-02	1.15e-02	4.02e-02	1.33e-02	3.52e-02	1.31e-02
	CPU time	9.97e+00	1.30e+01	9.34e+00	1.22e+01	9.66e+00	1.22e+01	9.81e+00	1.22e+01	9.00e+00	1.22e+01
CF5	mean	2.90e-01	3.01e-01	3.10e-01	3.27e-01	3.08e-01	3.08e-01	2.90e-01	3.01e-01	3.38e-01	**2.41e-01**
	std	1.20e-01	1.40e-01	1.21e-01	1.55e-01	1.16e-01	1.50e-01	1.20e-01	1.40e-01	1.27e-01	8.65e-02
	CPU time	1.02e+01	1.20e+01	1.00e+01	1.17e+01	1.03e+01	1.17e+01	1.03e+01	1.15e+01	9.92e+00	1.18e+01
CF6	mean	8.80e-02	1.33e+00	1.34e-01	**3.57e-02**	1.09e-01	**5.73e-01**	8.80e-02	1.33e+00	1.11e-01	**4.14e-02**
	std	3.03e-02	1.59e+00	3.14e-02	9.35e-03	2.70e-02	1.13e+00	3.03e-02	1.59e+00	4.08e-02	4.80e-03
	CPU time	1.11e+01	3.81e+01	1.08e+01	1.71e+01	1.08e+01	2.51e+01	1.12e+01	3.50e+01	1.07e+01	1.71e+01
CF7	mean	2.53e-01	2.58e-01	6.11e-01	**1.58e-01**	3.52e-01	**2.10e-01**	2.53e-01	2.58e-01	3.08e-01	**1.79e-01**
	std	1.22e-01	1.25e-01	2.08e+00	5.48e-02	1.53e-01	9.94e-02	1.22e-01	1.25e-01	1.35e-01	5.74e-02
	CPU time	1.10e+01	1.74e+01	1.16e+01	1.84e+01	1.11e+01	1.61e+01	1.04e+01	1.52e+01	9.43e+00	1.69e+01
CF8	mean	2.23e+09	**2.37e-01**	1.66e-01	**1.43e-01**	2.23e+09	**1.76e-01**	1.68e+09	**1.91e-01**	2.23e+09	**2.16e-01**
	std	5.90e+09	9.06e-02	2.54e-02	2.63e-02	5.90e+09	7.77e-02	5.21e+09	8.36e-02	5.90e+09	9.60e-02
	CPU time	5.46e+01	5.13e+01	3.47e+01	4.31e+01	5.90e+01	5.24e+01	5.21e+01	4.70e+01	7.82e+01	5.18e+01
CF9	mean	7.11e-02	7.12e-02	7.51e-02	7.76e-02	7.17e-02	7.67e-02	7.42e-02	7.12e-02	7.11e-02	7.12e-02
	std	7.07e-03	4.78e-03	6.84e-03	8.22e-03	5.36e-03	9.97e-03	1.03e-02	4.78e-03	7.07e-03	4.78e-03
	CPU time	5.77e+01	5.49e+01	6.12e+01	6.11e+01	6.15e+01	6.07e+01	5.33e+01	5.17e+01	5.41e+01	5.32e+01
CF10	mean	**2.42e-01**	6.63e+08	**2.42e-01**	3.19e-01	**2.42e-01**	6.63e+08	**2.42e-01**	2.95e-01	**2.42e-01**	1.33e+08
	std	5.52e-02	1.54e+09	5.52e-02	6.53e-02	5.52e-02	1.54e+09	5.52e-02	6.79e-02	5.52e-02	7.38e+08
	CPU time	6.37e+01	9.00e+01	6.32e+01	6.53e+01	6.33e+01	9.68e+01	6.39e+01	6.15e+01	6.18e+01	7.94e+01
Wilc. test (I-D-S)		1–7–2		1–4–5		1–4–5		1–7–2		1–3–6	

Table 3.3: HV values obtained on CF test problems with MOEA/D.

		CDP-MOEA/D		ATP-MOEA/D		C-MOEA/D		SR-MOEA/D		ε-MOEA/D		Imp ε-MOEA/D	
		—	w/grad	—	w/grad	—	w/grad	—	w/grad	—	w/grad	—	w/grad
CF1	mean	6.76e-01	**6.85e-01**	6.76e-01	**6.85e-01**	6.84e-01	**6.85e-01**	6.74e-01	**6.85e-01**	6.76e-01	**6.85e-01**	6.76e-01	**6.85e-01**
	std	2.24e-03	4.69e-05	2.83e-03	7.33e-05	6.13e-04	7.02e-05	2.38e-03	4.11e-05	2.24e-03	4.69e-05	2.24e-03	1.45e-04
CF2	mean	8.14e-01	8.16e-01	8.13e-01	**8.17e-01**	8.13e-01	**8.17e-01**	8.13e-01	8.14e-01	8.14e-01	8.16e-01	8.16e-01	8.15e-01
	std	8.31e-03	6.09e-03	7.43e-03	7.43e-03	6.28e-03	4.70e-03	9.56e-03	1.14e-02	8.31e-03	6.09e-03	3.52e-03	5.33e-03
CF3	mean	2.25e-01	2.33e-01	2.25e-01	2.32e-01	2.31e-01	2.33e-01	2.24e-01	2.13e-01	2.25e-01	2.33e-01	2.33e-01	2.33e-01
	std	5.56e-02	4.75e-02	4.22e-02	4.94e-02	4.64e-02	4.98e-02	5.51e-02	5.43e-02	5.56e-02	4.75e-02	4.42e-02	4.93e-02
CF4	mean	6.35e-01	6.42e-01	6.35e-01	6.38e-01	6.21e-01	6.42e-01	6.43e-01	6.37e-01	6.35e-01	6.42e-01	6.26e-01	6.37e-01
	std	3.61e-02	2.87e-02	3.54e-02	2.51e-02	5.24e-02	1.80e-02	3.60e-02	2.64e-02	3.61e-02	2.87e-02	4.04e-02	2.96e-02
CF5	mean	4.14e-01	4.13e-01	4.46e-01	4.56e-01	4.26e-01	4.14e-01	4.17e-01	4.41e-01	4.14e-01	4.13e-01	4.20e-01	4.25e-01
	std	7.54e-02	9.93e-02	6.25e-02	7.70e-02	1.00e-01	9.43e-02	9.83e-02	8.22e-02	7.54e-02	9.93e-02	8.72e-02	1.06e-01
CF6	mean	7.66e-01	**7.86e-01**	7.98e-01	**8.10e-01**	7.71e-01	**8.01e-01**	7.64e-01	**7.86e-01**	7.66e-01	**7.86e-01**	7.75e-01	**7.98e-01**
	std	1.62e-02	2.45e-02	2.23e-02	1.97e-02	1.92e-02	2.01e-02	9.18e-03	2.72e-02	1.62e-02	2.45e-02	1.81e-02	2.80e-02
CF7	mean	5.59e-01	5.59e-01	5.77e-01	5.91e-01	5.74e-01	5.91e-01	5.56e-01	5.98e-01	5.59e-01	5.59e-01	5.67e-01	5.72e-01
	std	1.05e-01	1.19e-01	1.02e-01	7.64e-02	7.80e-02	8.33e-02	1.03e-01	6.97e-02	1.05e-01	1.19e-01	8.26e-02	1.03e-01
CF8	mean	3.64e-01	**4.34e-01**	5.94e-01	5.97e-01	4.83e-01	**5.73e-01**	3.56e-01	**4.27e-01**	3.57e-01	**4.34e-01**	3.81e-01	**5.29e-01**
	std	6.90e-02	3.56e-02	3.46e-02	1.83e-02	9.62e-02	2.78e-02	2.00e-02	2.31e-02	9.65e-02	2.34e-02	2.21e-02	7.64e-02
CF9	mean	**6.72e-01**	6.58e-01	**6.83e-01**	6.59e-01	6.46e-01	6.50e-01	**6.55e-01**	6.48e-01	**6.75e-01**	6.58e-01	**6.72e-01**	6.58e-01
	std	2.07e-02	2.29e-02	1.87e-02	2.03e-02	1.82e-02	2.32e-02	2.78e-02	1.63e-02	1.87e-02	2.29e-02	2.07e-02	2.29e-02
CF10	mean	2.03e-02	3.24e-02	2.47e-01	2.87e-01	8.94e-03	0.00e+00	1.67e-02	1.63e-02	9.49e-02	8.70e-02	3.35e-02	3.94e-02
	std	7.45e-02	6.03e-02	9.64e-02	1.90e-01	3.52e-02	0.00e+00	7.26e-02	5.26e-02	1.29e-01	5.82e-02	6.49e-02	6.24e-02
Wilc. test (I-D-S)		1–6–3		1–6–3		0–6–4		1–6–3		1–6–3		1–6–3	

Table 3.4: HV values obtained on CF test problems with NSGA-II.

		CDP-NSGA-II		ATP-NSGA-II		C-NSGA-II		ε-NSGA-II		Imp ε-NSGA-II	
		—	w/grad	—	w/grad	—	w/grad	—	w/grad	—	w/grad
CF1	mean	6.52e-01	**6.84e-01**	3.95e-01	**6.81e-01**	6.01e-01	**6.84e-01**	6.52e-01	**6.84e-01**	6.52e-01	**6.81e-01**
	std	4.52e-03	2.07e-04	6.07e-02	2.08e-03	4.20e-02	1.49e-03	4.52e-03	2.07e-04	4.52e-03	2.79e-03
CF2	mean	7.38e-01	**7.56e-01**	7.37e-01	7.51e-01	7.37e-01	**7.53e-01**	7.38e-01	**7.56e-01**	7.47e-01	7.51e-01
	std	3.27e-02	3.04e-02	3.31e-02	2.42e-02	3.40e-02	3.24e-02	3.27e-02	3.04e-02	3.06e-02	2.73e-02
CF3	mean	2.26e-01	2.08e-01	2.08e-01	2.19e-01	2.20e-01	2.06e-01	2.26e-01	2.08e-01	2.11e-01	2.14e-01
	std	4.44e-02	4.93e-02	5.56e-02	5.37e-02	5.14e-02	4.97e-02	4.44e-02	4.93e-02	6.01e-02	5.54e-02
CF4	mean	5.62e-01	5.94e-01	5.26e-01	**5.99e-01**	5.48e-01	**5.86e-01**	5.62e-01	5.94e-01	5.49e-01	**5.92e-01**
	std	6.77e-02	2.13e-02	5.42e-02	2.27e-02	4.24e-02	1.65e-02	6.77e-02	2.13e-02	4.61e-02	1.78e-02
CF5	mean	3.75e-01	3.85e-01	3.69e-01	3.39e-01	3.69e-01	3.76e-01	3.75e-01	3.85e-01	3.43e-01	**4.10e-01**
	std	9.38e-02	1.13e-01	1.03e-01	1.22e-01	9.92e-02	1.16e-01	9.38e-02	1.13e-01	1.12e-01	8.82e-02
CF6	mean	7.62e-01	4.59e-01	7.46e-01	**7.98e-01**	7.61e-01	**6.39e-01**	7.62e-01	4.59e-01	7.47e-01	**7.95e-01**
	std	2.33e-02	3.96e-01	2.43e-02	8.30e-03	1.66e-02	3.18e-01	2.33e-02	3.96e-01	2.80e-02	6.65e-03
CF7	mean	5.76e-01	5.57e-01	5.46e-01	**6.29e-01**	4.89e-01	**5.91e-01**	5.76e-01	5.57e-01	5.38e-01	**6.08e-01**
	std	9.66e-02	1.01e-01	1.29e-01	6.28e-02	1.35e-01	8.55e-02	9.66e-02	1.01e-01	1.10e-01	7.99e-02
CF8	mean	3.40e-01	**4.79e-01**	4.30e-01	**5.14e-01**	3.23e-01	**4.87e-01**	3.53e-01	**4.89e-01**	3.40e-01	**4.79e-01**
	std	1.38e-01	6.21e-02	5.10e-02	3.51e-02	1.36e-01	6.54e-02	1.29e-01	8.72e-02	1.47e-01	8.32e-02
CF9	mean	5.86e-01	5.88e-01	5.81e-01	5.73e-01	5.85e-01	5.72e-01	5.84e-01	5.88e-01	5.86e-01	5.88e-01
	std	1.85e-02	1.24e-02	1.40e-02	2.46e-02	1.96e-02	2.61e-02	2.01e-02	1.24e-02	1.85e-02	1.24e-02
CF10	mean	**2.90e-01**	2.14e-01	**2.90e-01**	2.15e-01	**2.90e-01**	1.53e-01	**2.90e-01**	2.37e-01	**2.90e-01**	2.02e-01
	std	5.41e-02	1.11e-01	5.41e-02	4.97e-02	5.41e-02	9.28e-02	5.41e-02	6.22e-02	5.41e-02	7.62e-02
Wilc. test (I-D-S)		1–6–3		1–4–5		1–3–6		1–6–3		1–3–6	

from the objective functions, which involve a significant number of locally optimal fronts. To demonstrate this point, let us consider function CF5. Figure 3.1 displays the convergence issues observed when trying to approximate the true Pareto front using C-MOEA/D for this problem, independently from the use of the repair procedure. Anyway, note that all the solutions in Figure 3.1 are feasible. Now, in Figure 3.2 (left), the unconstrained version of CF5 is solved again with MOEA/D. The first observation is that, when compared with Figure 3.1, the true Pareto front of the constrained problem differs from the unconstrained true Pareto front only in its lower part, i.e. for $0.5 \leq f_1 \leq 1$. In addition, the similarity with Figure 3.1 regarding convergence tends to confirm that the poor results observed in Figure 3.1 may not be attributable to difficulties for fulfilling the constraints, but rather for escaping from locally optimal sub-fronts and finding the optimal one. This would also explain the high percentage of cases for which there is no significant difference obtained by using the gradient-based repair method (56.7% for MOEA/D, 47% for NSGA-II). Figure 3.2 (right) illustrates the same convergence issues for problem CF3.

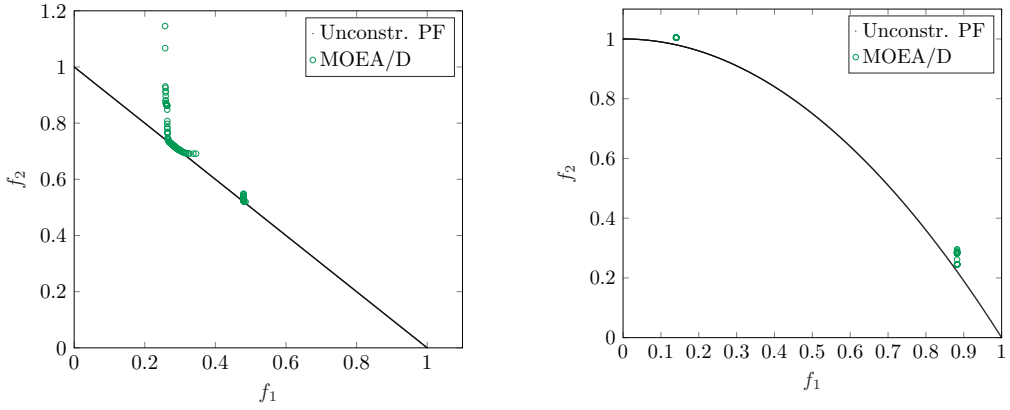


Figure 3.2: Final Pareto front approximation of unconstrained functions CF5 (left) and CF3 (right).

3.5.2 LIRCMOP test problems

As mentioned previously, these test problems were built in such a way that the search space contains large infeasible regions, as well as disconnected feasible regions (islands), involving difficulties for the search algorithm, which might be stuck in a suboptimal island enclosed by infeasible regions (Fan et al., 2019b). The results obtained with the different constraint handling techniques, without and with the gradient-based repair, are presented in Tables 3.5 to 3.8. For this test suite, it can be appreciated that the use of the gradient-based repair significantly improves the performance of the six constraint-handling techniques, with the exception of the improved ε-constraint technique with NSGA-II, which shows divided results. Besides, from a global perspective, it can be observed that MOEA/D outperforms NSGA-II on this benchmark, considering for example the HV values for both algorithms using CDP, MOEA/D obtains higher values on all problems.

More accurately, problems LIRCMOP1-4 present large infeasible regions and their $\mathcal{PF}$ are located on the feasible region boundaries. For these problems, the original techniques are able to find at least a part of the true PFs. However, because of the

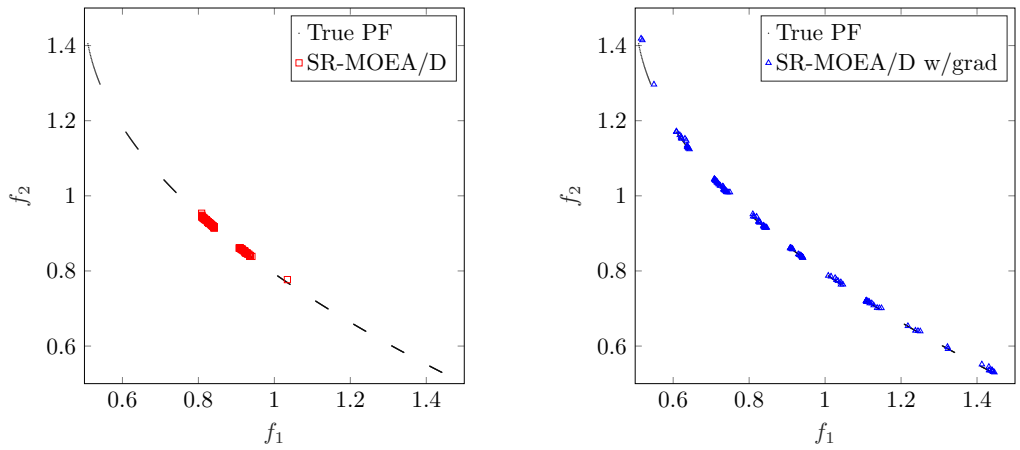


Figure 3.3: Final Pareto front approximation of LIRCMOP4 function. SR-MOEA/D without (left) and with (right) gradient-based repair.

Table 3.5: IGD values and CPU time (in seconds) obtained on LIRCMOP test problems with MOEA/D.

		CDP-MOEA/D		ATP-MOEA/D		C-MOEA/D		SR-MOEA/D		ε-MOEA/D		Imp ε-MOEA/D	
		—	w/grad	—	w/grad	—	w/grad	—	w/grad	—	w/grad	—	w/grad
LIRCMOP1	mean	2.11e-01	**4.99e-02**	7.98e+07	**5.25e-02**	1.51e-01	**1.23e-02**	2.07e-01	**3.16e-02**	5.65e-02	**7.94e-03**	7.84e-02	**4.24e-02**
	std	4.86e-02	2.15e-02	9.05e+07	3.47e-02	6.23e-02	1.12e-02	5.36e-02	2.53e-02	4.52e-02	1.76e-03	4.69e-02	1.79e-02
	CPU time	6.92e+01	4.81e+01	7.68e+01	4.65e+01	7.02e+01	5.36e+01	7.70e+01	5.58e+01	7.90e+01	5.24e+01	7.68e+01	5.02e+01
LIRCMOP2	mean	1.61e-01	**8.85e-02**	1.48e+08	**3.71e-02**	1.13e-01	**3.71e-02**	1.27e-01	**5.82e-02**	5.89e-02	**1.37e-02**	7.09e-02	**2.21e-02**
	std	4.46e-02	3.45e-02	1.62e+08	2.03e-02	4.76e-02	2.80e-02	5.66e-02	3.45e-02	3.85e-02	1.25e-02	4.35e-02	1.99e-02
	CPU time	7.65e+01	5.29e+01	7.49e+01	4.62e+01	6.89e+01	4.53e+01	6.65e+01	4.93e+01	6.88e+01	4.48e+01	6.85e+01	4.80e+01
LIRCMOP3	mean	2.08e-01	**1.03e-02**	3.38e+09	**1.90e-01**	1.95e-01	**7.08e-03**	1.83e-01	**9.70e-03**	1.34e-01	**5.41e-03**	1.66e-01	**3.47e-03**
	std	3.38e-02	9.14e-03	2.87e+09	1.54e-01	4.56e-02	8.84e-04	5.61e-02	1.09e-02	7.98e-02	2.56e-03	6.65e-02	5.65e-04
	CPU time	6.69e+01	5.46e+01	7.51e+01	5.29e+01	7.71e+01	5.82e+01	8.07e+01	6.21e+01	7.78e+01	5.85e+01	7.37e+01	5.29e+01
LIRCMOP4	mean	1.99e-01	**9.08e-03**	4.30e+09	**1.27e-01**	1.83e-01	**8.00e-03**	1.73e-01	**6.72e-03**	1.71e-01	**4.81e-03**	1.77e-01	**3.26e-03**
	std	3.16e-02	6.78e-03	2.83e+09	8.10e-02	3.74e-02	3.30e-03	4.49e-02	3.83e-03	6.13e-02	1.79e-03	5.05e-02	4.73e-04
	CPU time	7.04e+01	5.46e+01	7.78e+01	5.56e+01	6.91e+01	5.33e+01	6.82e+01	5.33e+01	7.13e+01	5.12e+01	6.97e+01	6.00e+01
LIRCMOP5	mean	1.22e+00	**3.39e-02**	7.43e-01	**2.76e-03**	1.20e+00	**5.28e-03**	1.25e+00	**2.88e-02**	1.22e+00	**3.39e-02**	1.00e-02	**2.90e-03**
	std	6.15e-02	7.04e-02	5.11e-01	3.48e-04	1.49e-02	3.30e-03	2.46e-01	1.30e-02	6.15e-02	7.04e-02	1.21e-02	4.13e-04
	CPU time	7.97e+01	7.59e+01	7.90e+01	7.71e+01	7.28e+01	7.31e+01	7.94e+01	7.63e+01	7.77e+01	7.94e+01	8.04e+01	8.31e+01
LIRCMOP6	mean	1.35e+00	**3.04e-02**	1.35e+00	**2.88e-03**	1.35e+00	**4.30e-03**	1.35e+00	**3.75e-02**	1.35e+00	**3.04e-02**	6.93e-03	**2.90e-03**
	std	1.79e-04	6.32e-02	2.74e-04	2.63e-04	1.54e-04	1.38e-03	2.20e-04	4.16e-02	1.79e-04	6.32e-02	1.08e-02	3.02e-04
	CPU time	6.51e+01	7.23e+01	7.73e+01	7.78e+01	7.10e+01	7.00e+01	6.76e+01	6.86e+01	6.37e+01	6.95e+01	7.42e+01	7.62e+01
LIRCMOP7	mean	1.53e+00	**6.64e-03**	6.59e-01	**2.88e-03**	1.48e+00	**3.12e-03**	1.53e+00	**5.77e-03**	1.53e+00	**6.64e-03**	9.38e-03	**2.99e-03**
	std	4.76e-01	1.26e-02	7.73e-01	1.53e-03	5.37e-01	2.86e-03	4.75e-01	9.84e-03	4.76e-01	1.26e-02	1.06e-02	6.12e-04
	CPU time	7.65e+01	8.44e+01	8.91e+01	8.72e+01	8.35e+01	8.35e+01	7.42e+01	7.47e+01	6.90e+01	7.14e+01	6.98e+01	6.87e+01
LIRCMOP8	mean	1.56e+00	**3.19e-03**	1.26e+00	**2.47e-03**	1.64e+00	**2.51e-03**	1.63e+00	**2.48e-03**	1.56e+00	**3.19e-03**	7.80e-03	**2.67e-03**
	std	3.72e-01	2.95e-03	6.82e-01	1.50e-04	2.36e-01	2.98e-04	2.11e-01	2.12e-04	3.72e-01	2.95e-03	3.52e-03	1.71e-04
	CPU time	6.76e+01	7.38e+01	7.73e+01	7.73e+01	7.06e+01	7.26e+01	7.18e+01	7.17e+01	6.80e+01	7.08e+01	7.12e+01	7.22e+01
LIRCMOP9	mean	5.84e-01	**3.72e-01**	**2.75e-01**	3.64e-01	4.28e-01	**3.65e-01**	6.93e-01	**3.73e-01**	5.84e-01	**3.72e-01**	2.20e-01	**1.00e-02**
	std	8.51e-02	5.83e-03	1.17e-01	6.77e-02	6.08e-02	6.41e-02	7.93e-02	6.63e-03	8.51e-02	5.83e-03	1.14e-01	3.32e-03
	CPU time	7.65e+01	7.67e+01	8.91e+01	8.06e+01	7.87e+01	7.60e+01	8.19e+01	6.65e+01	6.68e+01	6.69e+01	6.74e+01	6.60e+01
LIRCMOP10	mean	5.28e-01	**2.34e-01**	**4.43e-03**	6.20e-03	1.33e-01	**2.54e-02**	4.88e-01	**2.54e-01**	5.28e-01	**2.34e-01**	**3.71e-03**	5.50e-03
	std	1.85e-01	8.60e-02	9.55e-04	2.25e-03	8.10e-02	1.71e-02	1.45e-01	8.39e-02	1.85e-01	8.60e-02	5.16e-04	7.23e-04
	CPU time	7.30e+01	7.15e+01	8.90e+01	8.03e+01	7.43e+01	7.31e+01	7.34e+01	7.39e+01	7.02e+01	7.18e+01	8.66e+01	7.79e+01
LIRCMOP11	mean	6.12e-01	**2.54e-01**	7.06e-03	2.27e-02	1.43e-01	**7.97e-02**	6.28e-01	**2.43e-01**	6.12e-01	**2.54e-01**	**3.33e-03**	8.18e-03
	std	1.58e-01	1.12e-01	2.02e-03	2.81e-02	4.77e-02	4.80e-02	1.15e-01	8.73e-02	1.58e-01	1.12e-01	1.49e-03	2.83e-03
	CPU time	7.30e+01	8.10e+01	8.85e+01	7.49e+01	8.01e+01	7.33e+01	8.08e+01	7.68e+01	7.75e+01	7.75e+01	7.73e+01	6.85e+01
LIRCMOP12	mean	4.25e-01	**1.75e-01**	1.28e-01	1.54e-01	2.75e-01	**1.61e-01**	4.76e-01	**1.90e-01**	4.25e-01	**1.75e-01**	7.14e-02	7.04e-02
	std	7.50e-02	2.48e-02	4.90e-02	2.02e-02	2.81e-02	3.35e-02	9.25e-02	2.84e-02	7.50e-02	2.48e-02	5.84e-02	4.68e-02
	CPU time	8.35e+01	8.16e+01	9.38e+01	7.60e+01	7.84e+01	7.05e+01	7.92e+01	7.52e+01	8.19e+01	8.03e+01	7.78e+01	6.65e+01
LIRCMOP13	mean	1.30e+00	**6.40e-02**	8.76e-01	**6.41e-02**	1.17e+00	**6.38e-02**	1.25e+00	**6.40e-02**	1.30e+00	**6.40e-02**	6.78e-02	**6.39e-02**
	std	5.11e-02	1.27e-03	5.94e-01	9.31e-04	3.54e-01	1.24e-03	1.89e-01	1.04e-03	5.11e-02	1.27e-03	1.04e-03	8.75e-04
	CPU time	1.13e+02	1.90e+02	1.47e+02	1.93e+02	1.43e+02	1.92e+02	1.43e+02	1.94e+02	1.11e+02	1.89e+02	1.70e+02	1.98e+02
LIRCMOP14	mean	1.25e+00	**6.45e-02**	8.43e-01	**6.42e-02**	1.13e+00	**6.39e-02**	1.21e+00	**6.42e-02**	1.25e+00	**6.45e-02**	6.51e-02	**6.41e-02**
	std	4.68e-02	1.20e-03	5.79e-01	1.42e-03	3.47e-01	1.11e-03	1.85e-01	1.50e-03	4.68e-02	1.20e-03	8.47e-04	1.15e-03
	CPU time	1.20e+02	1.91e+02	1.52e+02	1.82e+02	1.40e+02	1.75e+02	1.38e+02	1.61e+02	9.84e+01	1.57e+02	1.53e+02	1.54e+02
Wilc. test (I-D-S)		0–0–14		2–2–10		0–0–14		0–0–14		0–0–14		2–1–11	

Table 3.6: IGD values and CPU time (in seconds) obtained on LIRCMOP test problems with NSGA-II.

		CDP-NSGA-II		ATP-NSGA-II		C-NSGA-II		ε-NSGA-II		Imp ε-NSGA-II	
		—	w/grad	—	w/grad	—	w/grad	—	w/grad	—	w/grad
LIRCMOP1	mean	2.69e-01	**2.54e-01**	3.70e+09	**1.90e+05**	3.84e+03	**2.39e-01**	1.83e-01	**8.34e-02**	**2.67e-01**	2.24e+05
	std	2.09e-02	2.44e-02	1.28e+09	2.96e+05	1.49e+04	2.93e-02	1.53e-02	2.67e-02	7.68e-02	4.16e+05
	CPU time	6.16e+01	6.88e+01	1.91e+01	3.51e+01	1.01e+02	5.33e+01	5.16e+01	5.32e+01	5.98e+01	3.46e+01
LIRCMOP2	mean	2.26e-01	2.32e-01	1.29e+09	**5.03e+05**	8.92e+03	**2.12e-01**	1.71e-01	**8.44e-02**	**2.47e-01**	8.21e+05
	std	2.14e-02	1.59e-02	1.08e+09	5.94e+05	2.94e+04	1.73e-02	1.02e-02	1.65e-02	8.35e-02	9.35e+05
	CPU time	6.17e+01	6.88e+01	1.76e+01	3.24e+01	9.31e+01	4.69e+01	4.62e+01	5.07e+01	6.05e+01	3.54e+01
LIRCMOP3	mean	2.67e-01	**2.23e-01**	2.99e+09	**1.20e+07**	2.74e+05	**1.95e+00**	1.97e-01	**4.91e-02**	2.38e-01	**3.37e-02**
	std	2.86e-02	2.80e-02	1.25e+09	2.42e+07	3.99e+05	4.46e+00	2.79e-02	3.28e-02	7.95e-02	1.61e-02
	CPU time	6.51e+01	7.78e+01	1.71e+01	3.98e+01	1.07e+02	7.43e+01	4.70e+01	5.43e+01	5.58e+01	4.24e+01
LIRCMOP4	mean	2.49e-01	**2.08e-01**	2.55e+09	**1.42e+07**	2.85e+05	**2.28e+01**	1.90e-01	**3.87e-02**	2.63e-01	**3.20e-02**
	std	2.50e-02	2.27e-02	1.23e+09	1.55e+07	3.83e+05	1.06e+02	2.12e-02	2.59e-02	8.47e-02	1.75e-02
	CPU time	6.23e+01	7.58e+01	2.35e+01	4.26e+01	1.19e+02	8.01e+01	5.16e+01	6.03e+01	5.54e+01	4.51e+01
LIRCMOP5	mean	1.22e+00	**5.14e-01**	2.09e+00	**2.94e-01**	1.23e+00	**4.21e-01**	1.22e+00	**5.14e-01**	3.14e-01	**2.89e-01**
	std	5.30e-03	3.82e-01	4.90e+00	3.77e-02	8.89e-03	3.12e-01	5.30e-03	3.82e-01	5.08e-02	3.69e-02
	CPU time	1.69e+01	5.19e+01	1.75e+01	4.57e+01	1.78e+01	4.84e+01	1.61e+01	4.83e+01	2.13e+01	4.77e+01
LIRCMOP6	mean	1.34e+00	**3.93e-01**	2.64e+00	**3.16e-01**	1.44e+00	**3.34e-01**	1.34e+00	**3.93e-01**	3.39e-01	3.28e-01
	std	5.40e-05	1.86e-01	4.54e+00	5.97e-02	1.57e-01	6.39e-02	5.40e-05	1.86e-01	5.89e-02	6.27e-02
	CPU time	1.67e+01	4.53e+01	1.73e+01	4.63e+01	1.69e+01	4.37e+01	1.57e+01	4.24e+01	2.16e+01	4.60e+01
LIRCMOP7	mean	4.18e-01	**1.25e-01**	6.40e+00	**1.33e-01**	3.44e-01	**1.27e-01**	4.18e-01	**1.25e-01**	**1.42e-01**	2.99e-01
	std	5.70e-01	1.26e-02	8.71e+00	2.56e-02	5.55e-01	1.72e-02	5.70e-01	1.26e-02	2.43e-02	1.70e-01
	CPU time	2.09e+01	6.13e+01	1.72e+01	5.08e+01	1.68e+01	5.29e+01	1.84e+01	5.19e+01	2.50e+01	4.64e+01
LIRCMOP8	mean	1.15e+00	**1.99e-01**	2.63e+00	**1.92e-01**	1.22e+00	**1.99e-01**	1.15e+00	**1.99e-01**	**2.11e-01**	2.54e-01
	std	6.86e-01	3.21e-02	5.96e+00	3.19e-02	9.44e-01	2.70e-02	6.86e-01	3.21e-02	3.22e-02	8.33e-02
	CPU time	1.66e+01	5.08e+01	1.57e+01	4.83e+01	1.54e+01	4.88e+01	1.59e+01	4.86e+01	2.28e+01	5.18e+01
LIRCMOP9	mean	1.01e+00	**8.40e-01**	8.64e-01	**6.80e-01**	1.30e+00	**7.49e-01**	1.01e+00	**8.40e-01**	7.89e-01	**4.72e-01**
	std	5.60e-02	1.56e-01	2.87e-01	2.36e-01	2.25e+00	2.31e-01	5.60e-02	1.56e-01	2.22e-01	3.15e-01
	CPU time	1.93e+01	4.68e+01	7.54e+01	5.11e+01	2.09e+01	4.43e+01	1.85e+01	4.43e+01	2.19e+01	4.55e+01
LIRCMOP10	mean	9.15e-01	**6.35e-01**	**2.67e-01**	5.03e-01	9.25e-01	**5.95e-01**	9.15e-01	**6.35e-01**	3.12e-01	4.67e-01
	std	1.01e-01	2.29e-01	1.52e-01	2.92e-01	1.09e-01	2.34e-01	1.01e-01	2.29e-01	1.81e-01	3.22e-01
	CPU time	1.94e+01	4.52e+01	2.43e+01	5.17e+01	1.94e+01	4.58e+01	1.97e+01	3.97e+01	2.16e+01	4.00e+01
LIRCMOP11	mean	8.34e-01	**4.95e-01**	3.97e-01	4.95e-01	4.93e-01	4.91e-01	8.34e-01	4.95e-01	4.09e-01	4.59e-01
	std	1.04e-01	1.96e-01	1.27e-01	3.75e-01	2.87e-01	2.51e-01	1.04e-01	1.96e-01	2.08e-01	3.07e-01
	CPU time	1.63e+01	4.35e+01	4.35e+01	4.69e+01	3.32e+01	4.12e+01	1.58e+01	4.03e+01	2.27e+01	4.07e+01
LIRCMOP12	mean	8.59e-01	**3.75e-01**	7.92e-01	6.96e-01	5.47e-01	**4.52e-01**	8.59e-01	**3.75e-01**	**5.41e-01**	7.13e-01
	std	1.56e-01	2.13e-01	3.23e-01	3.11e-01	4.05e-01	2.66e-01	1.56e-01	2.13e-01	1.51e-01	1.67e-01
	CPU time	1.79e+01	4.45e+01	6.81e+01	6.00e+01	2.09e+01	4.60e+01	1.93e+01	4.83e+01	2.07e+01	4.64e+01
LIRCMOP13	mean	1.31e+00	**6.81e-02**	1.33e+00	**6.77e-02**	1.32e+00	**6.81e-02**	1.31e+00	**6.81e-02**	1.08e-01	6.80e-02
	std	7.70e-04	1.65e-03	4.13e-03	1.50e-03	2.05e-03	1.91e-03	7.70e-04	1.65e-03	2.23e-01	1.78e-03
	CPU time	4.23e+01	1.71e+02	6.52e+01	1.61e+02	3.99e+01	1.59e+02	4.17e+01	1.68e+02	1.66e+02	1.67e+02
LIRCMOP14	mean	1.26e+00	**6.96e-02**	1.29e+00	**6.95e-02**	1.28e+00	**6.95e-02**	1.26e+00	**6.96e-02**	1.11e-01	**7.04e-02**
	std	7.93e-04	1.76e-03	4.30e-03	2.02e-03	2.12e-03	2.20e-03	7.93e-04	1.76e-03	2.14e-01	2.57e-03
	CPU time	4.16e+01	1.45e+02	6.19e+01	1.47e+02	4.00e+01	1.39e+02	3.83e+01	1.37e+02	1.43e+02	1.49e+02
Wilc. test (I-D-S)		0–1–13		1–2–11		0–2–12		0–0–14		5–4–5	

Table 3.7: HV values obtained on LIRCMOP test problems with MOEA/D.

		CDP-MOEA/D		ATP-MOEA/D		C-MOEA/D		SR-MOEA/D		ε-MOEA/D		Imp ε-MOEA/D	
		—	w/grad	—	w/grad	—	w/grad	—	w/grad	—	w/grad	—	w/grad
LIRCMOP1	mean	3.16e-01	**4.65e-01**	6.14e-02	**4.72e-01**	3.71e-01	**5.20e-01**	3.19e-01	**4.89e-01**	4.66e-01	**5.30e-01**	4.43e-01	**4.81e-01**
	std	4.27e-02	2.77e-02	8.95e-02	3.81e-02	6.05e-02	2.02e-02	4.80e-02	3.43e-02	5.35e-02	4.81e-03	5.13e-02	2.36e-02
LIRCMOP2	mean	6.51e-01	**7.37e-01**	1.11e-01	**8.23e-01**	7.24e-01	**8.16e-01**	6.96e-01	**7.78e-01**	7.89e-01	**8.55e-01**	7.75e-01	**8.42e-01**
	std	5.71e-02	5.01e-02	1.95e-01	2.22e-02	6.31e-02	4.51e-02	7.36e-02	5.25e-02	5.18e-02	1.59e-02	5.81e-02	2.54e-02
LIRCMOP3	mean	3.23e-01	**5.06e-01**	2.83e-02	**3.42e-01**	3.34e-01	**5.12e-01**	3.43e-01	**5.07e-01**	3.87e-01	**5.15e-01**	3.58e-01	**5.19e-01**
	std	2.97e-02	1.33e-02	7.50e-02	8.62e-02	3.90e-02	2.06e-03	5.13e-02	1.54e-02	7.26e-02	2.65e-03	5.84e-02	1.19e-03
LIRCMOP4	mean	5.87e-01	**8.11e-01**	0.00e+00	**6.59e-01**	6.06e-01	**8.13e-01**	6.22e-01	**8.15e-01**	6.24e-01	**8.17e-01**	6.16e-01	**8.20e-01**
	std	3.66e-02	6.56e-03	0.00e+00	8.34e-02	4.67e-02	3.48e-03	5.44e-02	3.82e-03	6.98e-02	2.06e-03	5.60e-02	1.29e-03
LIRCMOP5	mean	0.00e+00	**8.12e-01**	2.88e-01	**8.71e-01**	0.00e+00	**8.67e-01**	0.00e+00	**8.28e-01**	0.00e+00	**8.12e-01**	8.58e-01	**8.71e-01**
	std	0.00e+00	1.14e-01	3.30e-01	6.08e-04	0.00e+00	6.07e-03	0.00e+00	2.41e-02	0.00e+00	1.14e-01	2.52e-02	6.52e-04
LIRCMOP6	mean	0.00e+00	**5.13e-01**	0.00e+00	**5.38e-01**	0.00e+00	**5.35e-01**	0.00e+00	**4.97e-01**	0.00e+00	**5.13e-01**	5.33e-01	**5.38e-01**
	std	0.00e+00	5.60e-02	0.00e+00	5.23e-04	0.00e+00	2.41e-03	0.00e+00	2.97e-02	0.00e+00	5.60e-02	1.08e-02	6.19e-04
LIRCMOP7	mean	5.49e-02	**6.22e-01**	3.55e-01	**6.26e-01**	7.18e-02	**6.26e-01**	5.30e-02	**6.23e-01**	5.49e-02	**6.22e-01**	6.18e-01	**6.26e-01**
	std	1.71e-01	1.17e-02	2.71e-01	2.28e-03	1.91e-01	2.27e-03	1.65e-01	1.05e-02	1.71e-01	1.17e-02	9.79e-03	1.26e-03
LIRCMOP8	mean	3.97e-02	**6.26e-01**	1.47e-01	**6.27e-01**	1.54e-02	**6.27e-01**	1.64e-02	**6.27e-01**	3.97e-02	**6.26e-01**	6.19e-01	**6.27e-01**
	std	1.26e-01	4.15e-03	2.36e-01	5.98e-04	8.59e-02	6.22e-04	6.93e-02	5.85e-04	1.26e-01	4.15e-03	3.38e-03	3.65e-04
LIRCMOP9	mean	3.79e-01	**5.20e-01**	**5.77e-01**	5.26e-01	4.89e-01	**5.17e-01**	2.80e-01	**5.17e-01**	3.79e-01	**5.20e-01**	6.09e-01	**6.78e-01**
	std	7.55e-02	1.35e-02	4.32e-02	3.11e-02	4.66e-02	3.27e-02	6.03e-02	1.29e-02	7.55e-02	1.35e-02	3.65e-02	3.50e-03
LIRCMOP10	mean	4.55e-01	**8.46e-01**	8.55e-01	**9.49e-01**	7.74e-01	**9.39e-01**	4.97e-01	**8.35e-01**	4.55e-01	**8.46e-01**	8.56e-01	**9.49e-01**
	std	2.02e-01	4.36e-02	8.66e-04	1.30e-03	4.97e-02	5.74e-03	1.64e-01	4.29e-02	2.02e-01	4.36e-02	4.90e-04	4.45e-04
LIRCMOP11	mean	3.62e-01	**6.43e-01**	8.36e-01	8.26e-01	7.30e-01	**7.84e-01**	3.31e-01	**6.47e-01**	3.62e-01	**6.43e-01**	**8.39e-01**	8.35e-01
	std	1.18e-01	9.54e-02	1.61e-03	1.61e-02	4.06e-02	3.94e-02	8.64e-02	6.82e-02	1.18e-01	9.54e-02	3.20e-04	2.09e-03
LIRCMOP12	mean	5.24e-01	**6.55e-01**	6.81e-01	6.69e-01	6.21e-01	**6.64e-01**	4.84e-01	**6.46e-01**	5.24e-01	**6.55e-01**	7.15e-01	7.11e-01
	std	4.85e-02	1.25e-02	2.86e-02	1.24e-02	6.78e-03	2.00e-02	5.74e-02	1.17e-02	4.85e-02	1.25e-02	3.11e-02	2.58e-02
LIRCMOP13	mean	2.65e-03	**7.49e-01**	2.49e-01	**7.49e-01**	6.64e-02	**7.49e-01**	1.70e-02	**7.50e-01**	2.65e-03	**7.49e-01**	7.37e-01	**7.50e-01**
	std	1.29e-02	2.41e-03	3.42e-01	2.41e-03	1.80e-01	2.05e-03	7.11e-02	1.65e-03	1.29e-02	2.41e-03	2.39e-03	2.02e-03
LIRCMOP14	mean	3.97e-03	**7.55e-01**	2.61e-01	**7.55e-01**	7.28e-02	**7.56e-01**	2.10e-02	**7.55e-01**	3.97e-03	**7.55e-01**	7.50e-01	**7.55e-01**
	std	1.47e-02	1.52e-03	3.57e-01	1.96e-03	1.93e-01	2.24e-03	7.96e-02	1.78e-03	1.47e-02	1.52e-03	1.93e-03	1.78e-03
Wilc. test (I-D-S)		0–0–14		1–2–11		0–0–14		0–0–14		0–0–14		1–1–12	

Table 3.8: HV values obtained on LIRCMOP test problems with NSGA-II.

		CDP-NSGA-II		ATP-NSGA-II		C-NSGA-II		ε-NSGA-II		Imp ε-NSGA-II	
		—	w/grad	—	w/grad	—	w/grad	—	w/grad	—	w/grad
LIRCMOP1	mean	2.75e-01	**2.86e-01**	4.80e-03	1.50e-02	2.33e-01	**2.96e-01**	3.38e-01	**4.39e-01**	**2.68e-01**	5.90e-03
	std	1.80e-02	1.88e-02	2.67e-02	4.79e-02	8.35e-02	2.08e-02	1.87e-02	1.57e-02	6.04e-02	3.28e-02
LIRCMOP2	mean	5.76e-01	5.71e-01	0.00e+00	8.30e-03	4.65e-01	**5.92e-01**	6.54e-01	**7.49e-01**	**5.90e-01**	0.00e+00
	std	2.36e-02	1.89e-02	0.00e+00	4.62e-02	1.88e-01	1.79e-02	1.42e-02	2.10e-02	7.01e-02	0.00e+00
LIRCMOP3	mean	2.78e-01	**3.14e-01**	0.00e+00	4.06e-03	1.09e-01	**2.14e-01**	3.33e-01	**4.75e-01**	2.95e-01	**4.81e-01**
	std	2.63e-02	2.65e-02	0.00e+00	2.26e-02	1.24e-01	1.23e-01	2.59e-02	2.08e-02	5.96e-02	1.82e-02
LIRCMOP4	mean	5.33e-01	**5.72e-01**	0.00e+00	0.00e+00	2.16e-01	**4.45e-01**	6.13e-01	**7.76e-01**	5.46e-01	**7.86e-01**
	std	2.57e-02	2.73e-02	0.00e+00	0.00e+00	2.45e-01	2.07e-01	2.50e-02	2.78e-02	7.00e-02	1.98e-02
LIRCMOP5	mean	0.00e+00	**3.21e-01**	1.98e-01	**4.24e-01**	0.00e+00	**3.61e-01**	0.00e+00	**3.21e-01**	4.18e-01	4.26e-01
	std	0.00e+00	1.83e-01	1.98e-01	4.49e-02	0.00e+00	1.51e-01	0.00e+00	1.83e-01	6.00e-02	4.45e-02
LIRCMOP6	mean	0.00e+00	**2.67e-01**	5.70e-02	**3.03e-01**	0.00e+00	**2.93e-01**	0.00e+00	**2.67e-01**	2.93e-01	2.96e-01
	std	0.00e+00	5.64e-02	1.07e-01	2.25e-02	0.00e+00	2.55e-02	0.00e+00	5.64e-02	2.57e-02	2.19e-02
LIRCMOP7	mean	4.19e-01	**5.26e-01**	1.69e-01	**5.20e-01**	4.36e-01	**5.24e-01**	4.19e-01	**5.26e-01**	**5.14e-01**	4.08e-01
	std	1.92e-01	8.87e-03	1.42e-01	1.92e-02	1.19e-01	1.11e-02	1.92e-01	8.87e-03	1.57e-02	6.86e-02
LIRCMOP8	mean	1.65e-01	**4.95e-01**	2.40e-01	**4.97e-01**	2.05e-01	**4.95e-01**	1.65e-01	**4.95e-01**	**4.87e-01**	4.43e-01
	std	2.19e-01	1.03e-02	1.49e-01	1.28e-02	2.18e-01	1.00e-02	2.19e-01	1.03e-02	1.24e-02	3.88e-02
LIRCMOP9	mean	1.34e-01	**2.48e-01**	3.09e-01	3.41e-01	1.99e-01	**3.06e-01**	1.34e-01	**2.48e-01**	3.41e-01	**4.57e-01**
	std	3.29e-02	1.08e-01	8.58e-02	1.31e-01	1.44e-01	1.30e-01	3.29e-02	1.08e-01	6.84e-02	1.57e-01
LIRCMOP10	mean	1.05e-01	**4.64e-01**	6.32e-01	5.64e-01	6.41e-02	**4.92e-01**	1.05e-01	**4.64e-01**	5.85e-01	5.81e-01
	std	8.73e-02	1.77e-01	1.44e-01	2.23e-01	1.05e-01	1.87e-01	8.73e-02	1.77e-01	1.72e-01	2.52e-01
LIRCMOP11	mean	2.43e-01	**4.84e-01**	5.74e-01	5.38e-01	**5.90e-01**	4.97e-01	2.43e-01	**4.84e-01**	5.63e-01	**6.47e-01**
	std	4.95e-02	1.42e-01	1.02e-01	1.80e-01	1.35e-01	1.92e-01	4.95e-02	1.42e-01	1.26e-01	1.55e-01
LIRCMOP12	mean	3.49e-01	**5.86e-01**	3.48e-01	4.23e-01	4.89e-01	5.32e-01	3.49e-01	**5.86e-01**	**5.38e-01**	4.47e-01
	std	1.16e-01	7.68e-02	1.67e-01	1.25e-01	1.48e-01	1.14e-01	1.16e-01	7.68e-02	4.67e-02	6.28e-02
LIRCMOP13	mean	3.89e-04	**7.47e-01**	1.10e-04	**7.47e-01**	2.40e-04	**7.47e-01**	3.89e-04	**7.47e-01**	7.21e-01	7.47e-01
	std	2.41e-04	2.92e-03	1.63e-04	2.44e-03	1.85e-04	3.56e-03	2.41e-04	2.92e-03	1.34e-01	2.25e-03
LIRCMOP14	mean	1.42e-03	**7.48e-01**	4.81e-04	**7.48e-01**	1.08e-03	**7.48e-01**	1.42e-03	**7.48e-01**	7.19e-01	**7.46e-01**
	std	4.54e-04	2.63e-03	4.65e-04	2.78e-03	3.98e-04	2.47e-03	4.54e-04	2.63e-03	1.33e-01	3.39e-03
Wilc. test (I-D-S)		0–1–13		0–8–6		1–1–12		0–0–14		5–4–5	

constrained search spaces, they cannot identify the whole $\mathcal{PF}$. That is, the search is stressed in some parts of the Pareto front, in particular those that are found first and from which the search algorithm does not manage to move away. On the other hand, when the gradient-based repair is incorporated, the MOEA is encouraged to continue searching promising regions even once some Pareto optimal solutions have been found. Indeed, since the true Pareto fronts are surrounded by infeasible regions, the new solutions produced from Pareto optimal solutions found so far (either by mating or mutation) are likely to be slightly infeasible and are thus easily repaired using the constraints' gradient information. Figure 3.3 illustrates this phenomenon for LIRCMOP4 function. It can be observed from Tables 3.5 and 3.7, that the repair process significantly improves every canonical constraint handling technique for all the problems under study when using MOEA/D.

Problems LIRCMOP5-8 have infeasible regions that may be difficult to cross when approaching the true PFs. These problems are particularly difficult for all the constraint-handling techniques studied here (with the exception of the improved ε-constraint), none of them is able to reach $\mathcal{PF}$, as they get stuck in sub-optimal fronts which are bounded by constraints. Conversely, when the gradient-based repair is used, every algorithm approximates the true Pareto fronts efficiently. Figures 3.4 and 3.5 show clearly how the gradient-based repair permits the individuals to pass over the infeasible region to attain the true $\mathcal{PF}$. Note that the regions dividing the true Pareto Fronts and the obtained approximations (Figures 3.4 and 3.5, left) are infeasible regions. Similar to problems LICRMOP1-4, the gradient-based repair significantly improves every canonical constraint-handling method for all these problems (except for the combination of NSGA-II with the improved ε constraint, for which the repair strategy slightly deteriorates the results). Furthermore, even the improved ε-constraint method, which was actually designed to be suited for this kind of problems, exhibits a significant enhancement by using the repair technique within MOEA/D.

Problems LIRCMOP9-12 contain, in addition to large infeasible regions, constraints that divide the $\mathcal{PF}$ into a number of disconnected segments. Once more, the results show that the repair of constraints presents overall good performance for these problems. Finally, functions LIRCMOP13 and LIRCMOP14 consist of three-objective problems, where the true Pareto fronts are located at the boundaries of the feasible region. The numerical results show that each algorithm with the gradient-based repair embedded obtains a good approximation of the $\mathcal{PF}$. Indeed, all the techniques reach the virtually highest possible hypervolume value for these problems, in every execution (for LIRCMOP13 AND 14: $\approx$ 7.50e-01 and $\approx$ 7.55e-01, respectively). It must be highlighted that, even the canonical methods showing poor performances (e.g., ATP, SR or CDP) achieve an excellent performance (better than that of canonical improved ε-method) once infeasible solutions are repaired.

3.5.3 EQC, Eq-DTLZ and Eq-IDTLZ test problems

The results obtained for problems containing equality constraints are presented in Tables 3.9 to 3.12. For problem EQC6 in Tables 3.9 and 3.10, the "—" means that no information is available regarding the true Pareto front, so that a reference set cannot be built to compute the IGD indicator. The results according to the IGD and HV metrics show that incorporating the gradient-based repair significantly outperforms the canonical constraint-handling methods for almost all problems. Besides, repairing infeasible solutions for problems with equality constraints seems to be necessary in

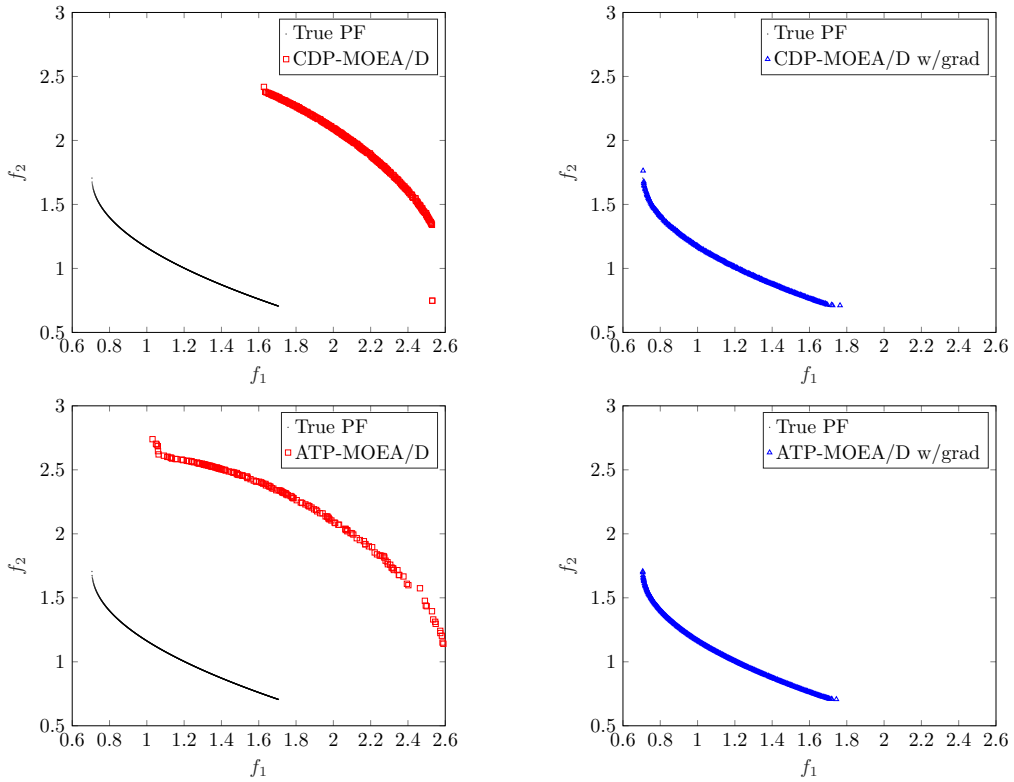


Figure 3.4: Final Pareto front approximation of LIRCMOP5 function. CDP-MOEA/D and ATP-MOEA/D without (left) and with (right) gradient-based repair.

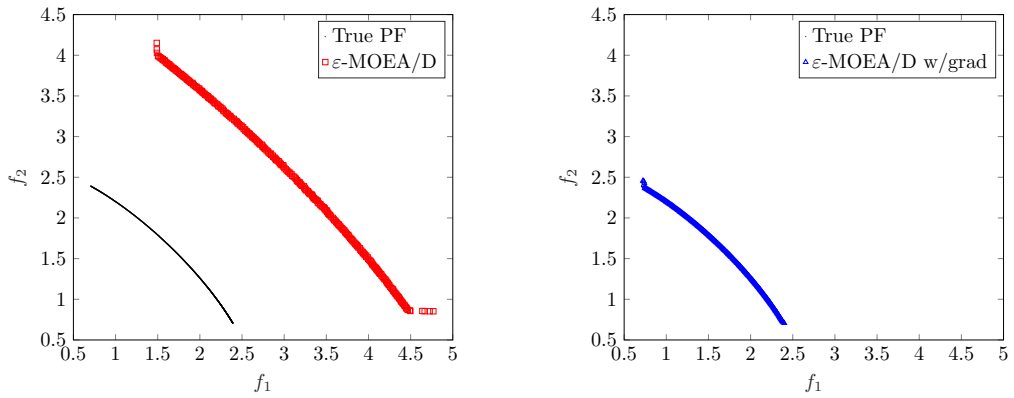


Figure 3.5: Final Pareto front approximation of LIRCMOP7 function. ε-MOEA/D without (left) and with (right) gradient-based repair.

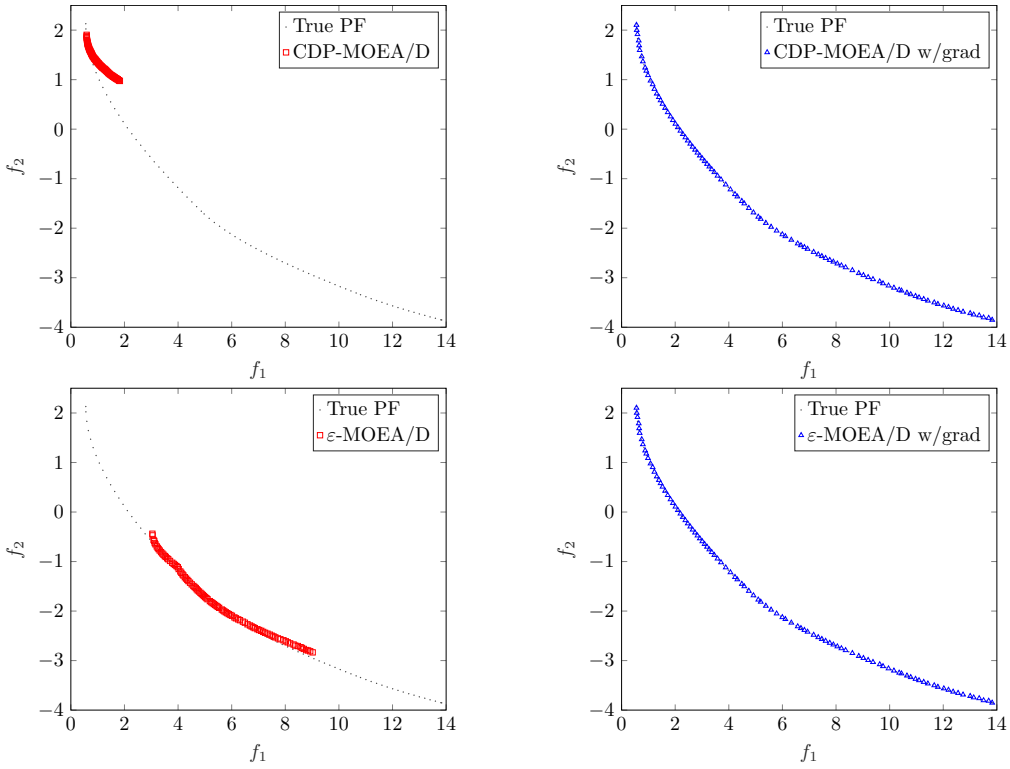


Figure 3.6: Final Pareto front approximation of EQC1 function. CDP-MOEA/D and ε-MOEA/D without (left) and with (right) gradient-based repair.

order to obtain acceptable quality solutions, or even feasible solutions (see the mean feasibility ratio of the final population over the 31 runs, presented in Tables 3.11 and 3.12).

Problem EQC1 (Das and Dennis, 1998) has only two equality constraints, however, none of the six constraint-handling techniques studied here is capable of approaching the whole Pareto front. For these canonical techniques, it seems that the exploration is stopped to some extent once a feasible non-dominated solution is found, due to their inability of escaping from a subregion defined by equality constraints. In contrast, when infeasible solutions are repaired with the constraints' gradient, the exploration is pursued all along the evolutionary process and, in this way, all the techniques approximate the Pareto front obtaining the maximum HV value ($\approx$ 9.09e-01). It must be emphasized that even the ATP method, which was unable to find a single feasible solution in its canonical form, now converges to the true PF. Figure 3.6 plots the Pareto front of the median run according to the HV indicator for CDP and ε-constraint methods with MOEA/D, without and with the implementation of the gradient-based repair.

Problems EQC2-3 (Cuate et al., 2020a) are modifications of the classical ZDT1 problem (Zitzler et al., 2000), including only one equality (quadratic) constraint, which involves only 2 out of the 30 decision variables. Problem EQC2 is solved by every algorithm when employing the gradient-based repair strategy, obtaining approximations of $\mathcal{PF}$ that show a good convergence and a uniform distribution. According to the numerical results of the IGD/HV indicators, the use of the constraints' gradient information seems more important than the base constraint-handling technique.

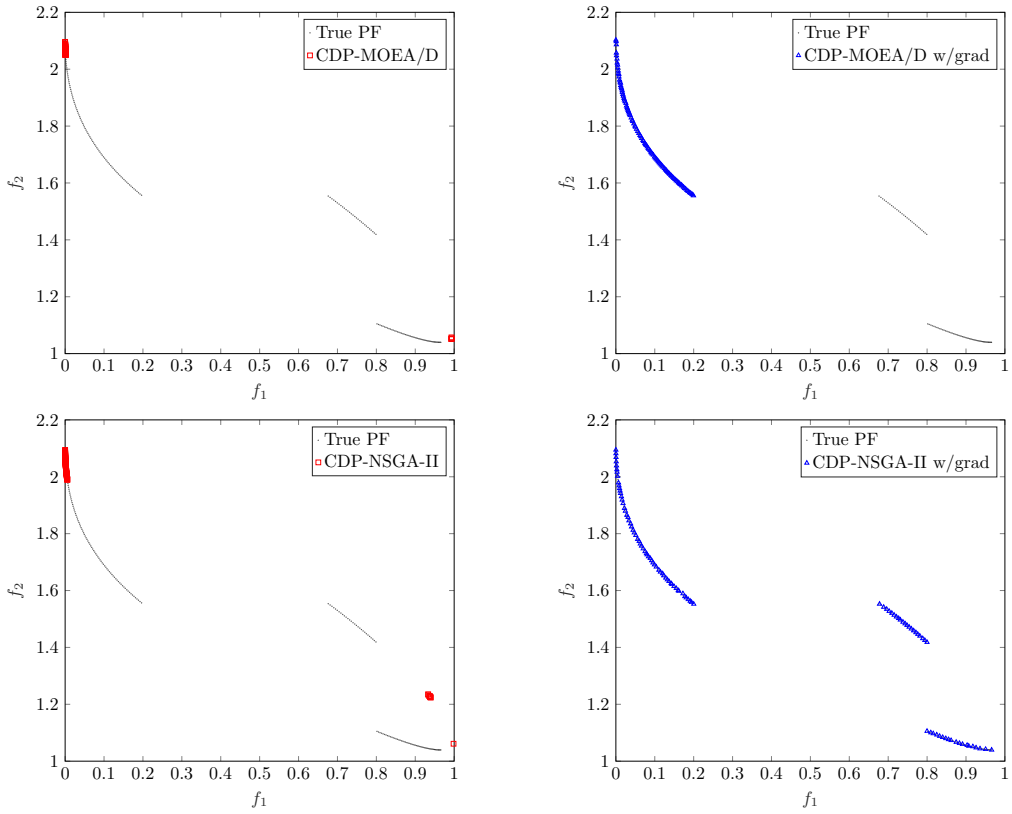


Figure 3.7: Final Pareto front approximation of EQC3 function. CDP-MOEA/D and CDP-NSGA-II without (left) and with (right) gradient-based repair.

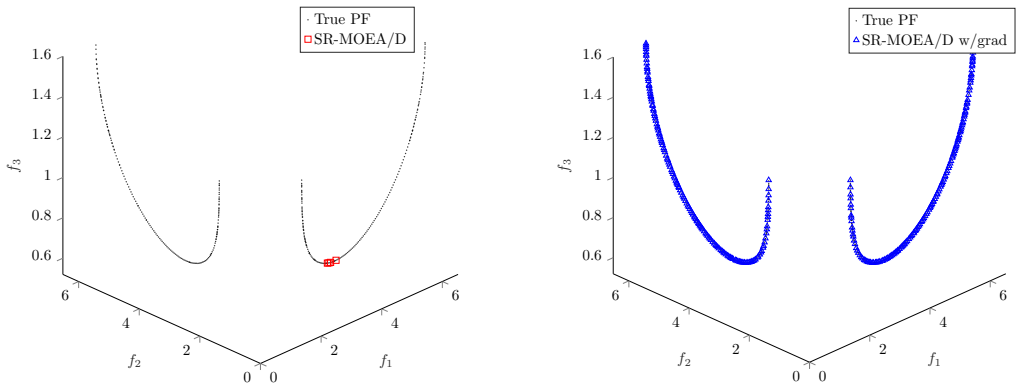


Figure 3.8: Final Pareto front approximation of EQC5 function. SR-MOEA/D without (left) and with (right) gradient-based repair.

Table 3.9: IGD values and CPU time (in seconds) obtained on EQC test problems with MOEA/D.

		CDP-MOEA/D		ATP-MOEA/D		C-MOEA/D		SR-MOEA/D		ε-MOEA/D		Imp ε-MOEA/D	
		—	w/grad	—	w/grad	—	w/grad	—	w/grad	—	w/grad	—	w/grad
EQC1	mean	4.80e+00	**4.29e-02**	9.56e+09	**4.09e-02**	2.63e+00	**4.16e-02**	4.19e+00	**4.30e-02**	1.66e+00	**4.29e-02**	3.08e+00	**4.87e-02**
	std	2.03e+00	2.52e-03	3.76e+09	3.73e-03	9.19e-01	2.91e-03	1.63e+00	3.50e-03	1.18e+00	2.52e-03	1.34e+00	7.16e-03
	CPU time	6.38e+01	4.71e+01	6.78e+01	4.27e+01	5.55e+01	4.25e+01	5.69e+01	4.30e+01	6.98e+01	4.22e+01	5.88e+01	3.75e+01
EQC2	mean	7.24e-01	**6.34e-03**	2.27e-02	**6.38e-03**	6.25e-01	**6.34e-03**	6.44e-01	**6.48e-03**	5.81e-01	**6.34e-03**	1.16e-01	**6.33e-03**
	std	6.43e-01	3.98e-04	7.65e-03	3.99e-04	2.26e-01	4.69e-04	4.58e-01	4.59e-04	1.34e-01	3.98e-04	2.45e-02	4.48e-04
	CPU time	5.81e+01	3.88e+01	6.05e+01	4.04e+01	5.34e+01	3.56e+01	5.40e+01	3.45e+01	5.14e+01	3.73e+01	5.58e+01	4.14e+01
EQC3	mean	5.87e-01	2.84e-01	8.00e-02	**9.29e-02**	4.58e-01	**2.99e-01**	**4.46e-01**	3.38e-01	6.61e-01	**2.84e-01**	1.36e-01	**9.58e-02**
	std	9.18e-01	1.39e-01	1.70e-02	8.82e-02	3.26e-01	1.23e-01	5.82e-01	7.80e-02	1.69e-01	1.39e-01	3.23e-02	1.04e-01
	CPU time	6.07e+01	4.47e+01	6.49e+01	4.09e+01	5.53e+01	3.98e+01	5.55e+01	4.03e+01	5.46e+01	3.98e+01	5.34e+01	2.99e+01
EQC4	mean	2.68e-01	**1.17e-01**	5.22e+05	**1.22e-01**	2.36e-01	**1.16e-01**	2.85e-01	**1.17e-01**	2.25e-01	**1.17e-01**	2.25e-01	**1.18e-01**
	std	4.45e-02	3.33e-03	2.05e+06	4.11e-03	3.71e-02	3.61e-03	3.82e-02	3.44e-03	3.60e-02	3.94e-03	3.68e-02	4.08e-03
	CPU time	1.99e+02	1.19e+02	8.17e+01	1.55e+02	1.45e+02	1.79e+02	1.46e+02	1.26e+02	1.55e+02	1.37e+02	1.51e+02	1.21e+02
EQC5	mean	1.06e+00	**7.86e-03**	1.87e+08	**1.02e-02**	1.03e+00	**8.15e-03**	2.03e+06	**7.93e-03**	1.11e+00	**7.94e-03**	1.05e+00	**7.88e-03**
	std	7.24e-01	2.62e-04	1.25e+08	1.22e-03	6.76e-01	2.77e-04	4.76e+06	2.08e-04	6.77e-01	2.67e-04	7.02e-01	2.41e-04
	CPU time	2.59e+02	1.44e+02	3.00e+01	1.53e+02	1.34e+02	1.46e+02	5.84e+01	1.57e+02	2.77e+02	1.49e+02	2.55e+02	1.42e+02
EQC6	mean	—	—	—	—	—	—	—	—	—	—	—	—
	std	—	—	—	—	—	—	—	—	—	—	—	—
	CPU time	4.55e+01	3.89e+01	5.00e+01	3.61e+01	4.27e+01	3.57e+01	4.28e+01	3.92e+01	4.57e+01	3.68e+01	4.46e+01	3.61e+01
Eq-DTLZ1	mean	1.15e-01	**1.90e-03**	8.92e-02	**1.96e-03**	7.85e-02	**1.90e-03**	1.25e-01	**1.92e-03**	1.05e-01	**1.90e-03**	1.19e-01	**1.93e-03**
	std	3.33e-02	8.87e-05	3.73e-02	7.71e-05	2.87e-02	8.86e-05	4.19e-02	1.03e-04	3.70e-02	8.87e-05	5.87e-02	9.37e-05
	CPU time	2.07e+02	1.06e+02	4.99e+01	1.08e+02	1.02e+02	9.91e+01	1.49e+02	1.05e+02	1.72e+02	1.15e+02	3.01e+02	1.49e+02
Eq-DTLZ2	mean	2.06e-01	**5.59e-03**	1.13e-01	**5.66e-03**	3.58e-02	**5.59e-03**	2.39e-01	**5.59e-03**	4.36e-02	**5.59e-03**	1.60e-01	**5.59e-03**
	std	1.38e-01	3.42e-04	3.26e-02	4.66e-04	4.67e-02	3.42e-04	1.20e-01	3.03e-04	1.33e-02	3.42e-04	7.67e-02	2.77e-04
	CPU time	1.92e+02	8.94e+01	4.57e+01	9.64e+01	1.22e+02	1.02e+02	1.52e+02	8.91e+01	2.05e+02	9.76e+01	1.98e+02	9.29e+01
Eq-DTLZ3	mean	3.23e-01	**5.88e-03**	1.91e-01	**5.85e-03**	2.62e-01	**5.88e-03**	4.10e-01	**5.98e-03**	3.13e-01	**5.88e-03**	3.19e-01	**5.86e-03**
	std	1.17e-01	4.59e-04	7.09e-02	2.92e-04	8.72e-02	4.59e-04	1.19e-01	4.49e-04	8.85e-02	4.59e-04	1.13e-01	4.18e-04
	CPU time	1.99e+02	1.40e+02	6.61e+01	1.40e+02	1.25e+02	1.30e+02	1.43e+02	1.24e+02	1.18e+02	9.22e+01	1.25e+02	9.31e+01
Eq-IDTLZ1	mean	1.39e-01	**2.15e-03**	1.29e-01	**2.20e-03**	6.82e-02	**2.15e-03**	1.49e-01	**2.07e-03**	1.30e-01	**2.15e-03**	1.26e-01	**2.18e-03**
	std	5.98e-02	1.16e-04	4.80e-02	1.40e-04	2.82e-02	1.16e-04	6.74e-02	1.53e-04	3.96e-02	1.16e-04	5.07e-02	1.82e-04
	CPU time	2.67e+02	1.96e+02	9.22e+01	1.90e+02	1.52e+02	1.81e+02	2.45e+02	1.69e+02	1.93e+02	1.93e+02	2.00e+02	1.73e+02
Eq-IDTLZ2	mean	3.00e-01	**7.19e-03**	2.35e-01	**7.39e-03**	5.05e-02	**7.19e-03**	3.12e-01	**7.32e-03**	9.54e-02	**7.19e-03**	1.67e-01	**7.25e-03**
	std	1.39e-01	8.51e-04	7.79e-02	7.82e-04	4.48e-02	8.51e-04	1.27e-01	4.30e-04	3.23e-02	8.51e-04	6.26e-02	6.31e-04
	CPU time	2.31e+02	2.04e+02	9.25e+01	2.12e+02	1.07e+02	1.73e+02	2.46e+02	2.15e+02	2.25e+02	1.83e+02	2.49e+02	2.06e+02
Eq-IDTLZ3	mean	3.57e-01	**8.54e-03**	5.04e-01	**1.02e-02**	1.93e-01	**8.54e-03**	3.99e-01	**8.51e-03**	2.53e-01	**8.54e-03**	2.66e-01	**9.92e-03**
	std	1.36e-01	1.87e-03	6.62e-01	5.48e-03	8.91e-02	1.87e-03	1.29e-01	1.28e-03	9.14e-02	1.87e-03	8.99e-02	7.80e-03
	CPU time	1.69e+02	1.73e+02	8.10e+01	1.84e+02	1.05e+02	1.93e+02	1.62e+02	1.78e+02	1.39e+02	1.68e+02	1.60e+02	1.71e+02
Wilc. test (I-D-S)		0–1–10		0–0–11		0–0–11		1–0–10		0–0–11		0–0–11	

Table 3.10: IGD values and CPU time (in seconds) obtained on EQC test problems with NSGA-II.

		CDP-NSGA-II		ATP-NSGA-II		C-NSGA-II		ε-NSGA-II		Imp ε-NSGA-II	
		—	w/grad	—	w/grad	—	w/grad	—	w/grad	—	w/grad
EQC1	mean	1.54e+05	**4.48e-02**	9.70e+09	**1.04e-01**	2.14e+06	**5.63e-02**	1.42e+05	**4.48e-02**	1.22e+05	**3.68e-01**
	std	6.37e+05	3.81e-03	5.97e+09	2.48e-02	1.76e+06	5.36e-03	5.75e+05	3.81e-03	4.76e+05	2.21e-01
	CPU time	2.14e+01	2.45e+01	9.40e+00	2.27e+01	2.50e+01	2.32e+01	1.36e+01	2.22e+01	1.48e+01	2.26e+01
EQC2	mean	4.70e-01	**4.96e-03**	6.71e+06	**5.03e-03**	4.42e-01	**4.93e-03**	1.97e-01	**4.96e-03**	2.37e-01	**5.04e-03**
	std	1.90e-01	1.30e-04	7.35e+06	1.28e-04	1.68e-01	1.17e-04	1.27e-01	1.30e-04	1.12e-01	2.16e-04
	CPU time	1.79e+01	2.39e+01	2.31e+01	2.31e+01	1.73e+01	2.31e+01	1.50e+01	2.32e+01	1.78e+01	2.31e+01
EQC3	mean	3.94e-01	**2.48e-02**	3.07e+06	**3.69e-02**	4.62e+04	**2.08e-02**	2.92e-01	**2.48e-02**	3.12e-01	**6.76e-02**
	std	2.20e-01	3.02e-02	2.47e+06	3.15e-02	2.57e+05	2.87e-02	1.56e-01	3.02e-02	1.70e-01	8.09e-04
	CPU time	1.98e+01	2.53e+01	2.46e+01	2.56e+01	1.86e+01	2.46e+01	1.59e+01	2.49e+01	1.92e+01	2.30e+01
EQC4	mean	3.30e-01	**1.22e-01**	1.00e+07	**1.23e-01**	4.52e-01	**1.23e-01**	3.21e-01	**1.21e-01**	3.05e-01	**1.32e-01**
	std	6.67e-02	4.50e-03	1.23e+07	3.87e-03	3.56e-02	4.74e-03	2.61e-02	4.57e-03	2.70e-02	5.71e-03
	CPU time	1.39e+02	1.17e+02	1.35e+02	1.17e+02	6.43e+01	1.23e+02	1.56e+02	1.22e+02	1.35e+02	1.03e+02
EQC5	mean	2.80e+00	**8.45e-03**	2.82e+08	**9.41e-02**	2.40e+00	**2.96e-02**	2.89e+00	**8.46e-03**	1.30e+05	**8.52e-03**
	std	5.31e-01	4.07e-04	1.60e+08	4.41e-02	8.28e-01	6.34e-03	8.01e-01	3.44e-04	7.26e+05	4.82e-04
	CPU time	2.96e+01	3.11e+01	5.06e+01	3.88e+01	4.63e+01	6.09e+01	3.08e+01	3.45e+01	2.24e+01	3.32e+01
EQC6	mean	—	—	—	—	—	—	—	—	—	—
	std	—	—	—	—	—	—	—	—	—	—
	CPU time	2.85e+01	3.03e+01	2.75e+01	2.76e+01	2.82e+01	2.72e+01	2.72e+01	2.82e+01	2.70e+01	2.70e+01
Eq-DTLZ1	mean	7.99e-02	**4.37e-03**	2.18e+06	**3.23e-03**	5.36e-02	**4.54e-03**	5.93e-02	**4.37e-03**	6.27e-02	**3.18e-03**
	std	3.91e-02	1.34e-03	2.44e+06	9.63e-04	2.69e-02	1.22e-03	2.14e-02	1.34e-03	2.72e-02	7.77e-04
	CPU time	4.98e+01	5.59e+01	6.32e+01	4.86e+01	3.18e+01	6.57e+01	4.74e+01	5.67e+01	5.95e+01	4.92e+01
Eq-DTLZ2	mean	1.23e-01	**1.53e-02**	1.43e+06	**1.11e-02**	8.47e-02	**1.52e-02**	7.27e-02	**1.53e-02**	7.32e-02	**1.10e-02**
	std	6.35e-02	3.46e-03	1.85e+06	3.58e-03	3.06e-02	4.86e-03	2.13e-02	3.46e-03	3.69e-02	3.13e-03
	CPU time	6.28e+01	6.08e+01	5.96e+01	6.11e+01	3.83e+01	6.54e+01	5.35e+01	5.55e+01	4.93e+01	5.35e+01
Eq-DTLZ3	mean	2.28e-01	**2.87e-02**	1.88e+06	**2.44e-02**	1.89e-01	**2.71e-02**	1.95e-01	**2.87e-02**	1.66e-01	**2.02e-02**
	std	8.73e-02	4.20e-03	2.47e+06	7.28e-03	7.60e-02	4.29e-03	5.79e-02	4.20e-03	4.69e-02	5.32e-03
	CPU time	3.86e+01	3.76e+01	4.97e+01	4.44e+01	2.88e+01	4.74e+01	4.02e+01	4.89e+01	4.71e+01	5.44e+01
Eq-IDTLZ1	mean	1.17e-01	**4.62e-03**	1.39e+06	**3.70e-03**	8.05e-02	**4.49e-03**	8.19e-02	**4.62e-03**	9.94e-02	**3.74e-03**
	std	4.59e-02	1.21e-03	2.54e+06	8.39e-04	3.29e-02	1.28e-03	3.03e-02	1.21e-03	4.04e-02	9.57e-04
	CPU time	2.23e+02	1.67e+02	5.39e+01	1.65e+02	3.59e+01	1.76e+02	2.37e+02	1.85e+02	1.96e+02	1.57e+02
Eq-IDTLZ2	mean	1.10e-01	**1.13e-02**	1.94e+06	**9.42e-03**	6.89e-02	**1.03e-02**	4.76e-02	**1.13e-02**	5.57e-02	**1.10e-02**
	std	5.38e-02	3.92e-03	2.09e+06	2.46e-03	1.42e-02	3.44e-03	1.48e-02	3.92e-03	2.39e-02	3.05e-03
	CPU time	1.60e+02	1.96e+02	1.05e+02	1.86e+02	2.82e+01	2.04e+02	1.78e+02	2.14e+02	1.62e+02	2.12e+02
Eq-IDTLZ3	mean	1.76e-01	**2.47e-02**	1.94e+06	**2.15e-02**	1.42e-01	**2.36e-02**	1.15e-01	**2.47e-02**	1.34e-01	**2.07e-02**
	std	7.11e-02	6.52e-03	2.38e+06	6.09e-03	4.51e-02	5.44e-03	2.98e-02	6.52e-03	7.16e-02	5.53e-03
	CPU time	1.29e+02	1.91e+02	4.96e+01	1.70e+02	2.69e+01	1.89e+02	1.41e+02	1.71e+02	1.40e+02	1.80e+02
Wilc. test (I-D-S)		0–0–11		0–0–11		0–0–11		0–0–11		0–0–11	

Table 3.11: HV values and I_F mean values obtained on EQC test problems with MOEA/D.

		CDP-MOEA/D		ATP-MOEA/D		C-MOEA/D		SR-MOEA/D		ε-MOEA/D		Imp ε-MOEA/D	
		—	w/grad	—	w/grad	—	w/grad	—	w/grad	—	w/grad	—	w/grad
EQC1	mean	3.41e-01	**9.09e-01**	0.00e+00	**9.09e-01**	5.04e-01	**9.09e-01**	3.65e-01	**9.09e-01**	7.65e-01	**9.09e-01**	6.32e-01	**9.08e-01**
	std	2.21e-01	1.17e-04	0.00e+00	2.20e-04	1.53e-01	1.60e-04	2.15e-01	1.28e-04	7.73e-02	1.17e-04	1.04e-01	5.77e-04
	I_F	1.0000	1.0000	0.0000	0.9242	0.0865	0.9497	0.9687	1.0000	1.0000	1.0000	1.0000	0.4539
EQC2	mean	1.68e-01	**9.08e-01**	8.97e-01	**9.08e-01**	2.08e-01	**9.08e-01**	1.71e-01	**9.07e-01**	3.40e-01	**9.08e-01**	7.32e-01	**9.08e-01**
	std	1.06e-01	5.46e-04	3.58e-03	6.96e-04	1.50e-01	6.25e-04	1.19e-01	6.62e-04	1.41e-01	5.46e-04	4.89e-02	5.62e-04
	I_F	1.0000	1.0000	0.0535	1.0000	0.4297	1.0000	0.9994	1.0000	1.0000	1.0000	1.0000	1.0000
EQC3	mean	2.25e-01	**6.71e-01**	7.51e-01	**7.57e-01**	3.54e-01	**6.67e-01**	2.37e-01	**6.49e-01**	1.81e-01	**6.71e-01**	6.93e-01	**7.43e-01**
	std	1.25e-01	5.54e-02	7.55e-03	3.84e-02	2.64e-01	5.10e-02	1.12e-01	3.21e-02	1.20e-01	5.54e-02	8.30e-02	4.03e-02
	I_F	1.0000	1.0000	0.0119	0.8035	0.4926	0.9552	0.9984	1.0000	1.0000	1.0000	1.0000	0.4410
EQC4	mean	7.71e-01	**8.30e-01**	3.76e-01	**8.30e-01**	7.84e-01	**8.30e-01**	7.65e-01	**8.30e-01**	7.95e-01	**8.30e-01**	7.95e-01	**8.30e-01**
	std	2.12e-02	4.16e-04	1.47e-01	5.99e-04	1.18e-02	7.48e-04	1.71e-02	5.81e-04	9.65e-03	6.67e-04	7.95e-03	4.67e-04
	I_F	1.0000	1.0000	0.0013	0.8053	0.2989	0.9931	0.3395	1.0000	1.0000	1.0000	1.0000	0.9725
EQC5	mean	5.75e-01	**7.77e-01**	0.00e+00	**7.77e-01**	5.91e-01	**7.77e-01**	3.12e-01	**7.77e-01**	5.78e-01	**7.77e-01**	5.88e-01	**7.77e-01**
	std	1.13e-01	4.86e-05	0.00e+00	2.79e-04	1.03e-01	3.43e-05	1.99e-01	3.51e-05	8.43e-02	4.41e-05	8.89e-02	4.79e-05
	I_F	1.0000	1.0000	0.0000	0.2027	0.1068	0.3392	0.1591	0.9920	1.0000	1.0000	1.0000	1.0000
EQC6	mean	0.00e+00	**1.04e+00**	0.00e+00	**1.04e+00**	0.00e+00	**1.04e+00**	0.00e+00	**1.04e+00**	0.00e+00	**1.04e+00**	0.00e+00	**7.62e-01**
	std	0.00e+00	6.23e-04	0.00e+00	1.12e-03	0.00e+00	4.48e-04	0.00e+00	2.79e-04	0.00e+00	1.50e-04	0.00e+00	2.88e-01
	I_F	0.0000	1.0000	0.0000	0.5568	0.0000	0.5623	0.0000	0.9990	0.0000	1.0000	0.0000	0.1613
Eq-DTLZ1	mean	4.17e-01	**8.41e-01**	4.76e-01	**8.41e-01**	5.78e-01	**8.41e-01**	4.01e-01	**8.41e-01**	4.86e-01	**8.41e-01**	4.48e-01	**8.41e-01**
	std	1.25e-01	2.43e-04	1.24e-01	2.74e-04	1.09e-01	2.43e-04	1.25e-01	5.04e-04	1.02e-01	2.43e-04	1.05e-01	3.14e-04
	I_F	1.0000	1.0000	0.0011	0.9911	0.1206	1.0000	0.9990	1.0000	1.0000	1.0000	1.0000	0.9873
Eq-DTLZ2	mean	3.58e-01	**6.60e-01**	4.21e-01	**6.60e-01**	5.77e-01	**6.60e-01**	3.47e-01	**6.60e-01**	5.44e-01	**6.60e-01**	4.43e-01	**6.60e-01**
	std	9.96e-02	3.86e-04	6.93e-02	4.63e-04	5.68e-02	3.86e-04	8.20e-02	3.34e-04	2.64e-02	3.86e-04	6.30e-02	4.45e-04
	I_F	1.0000	1.0000	0.0013	0.9828	0.1170	1.0000	0.9983	1.0000	1.0000	1.0000	1.0000	0.9780
Eq-DTLZ3	mean	2.89e-01	**6.57e-01**	2.94e-01	**6.58e-01**	3.51e-01	**6.57e-01**	2.57e-01	**6.51e-01**	3.22e-01	**6.57e-01**	3.18e-01	**6.55e-01**
	std	8.33e-02	2.17e-03	8.41e-02	1.75e-03	9.42e-02	2.17e-03	8.31e-02	2.47e-02	6.40e-02	2.17e-03	6.16e-02	1.26e-02
	I_F	1.0000	1.0000	0.0015	0.9977	0.1368	1.0000	0.9994	1.0000	1.0000	1.0000	1.0000	0.9972
Eq-IDTLZ1	mean	1.34e-01	**3.72e-01**	1.30e-01	**3.72e-01**	2.24e-01	**3.72e-01**	1.25e-01	**3.72e-01**	1.36e-01	**3.72e-01**	1.43e-01	**3.72e-01**
	std	5.74e-02	4.30e-04	4.50e-02	4.41e-04	3.85e-02	4.30e-04	5.51e-02	4.93e-04	4.50e-02	4.30e-04	4.35e-02	4.18e-04
	I_F	1.0000	1.0000	0.0003	0.9717	0.1200	1.0000	0.9995	1.0000	1.0000	1.0000	1.0000	0.9723
Eq-IDTLZ2	mean	2.07e-01	**4.94e-01**	2.54e-01	**4.95e-01**	4.42e-01	**4.94e-01**	1.98e-01	**4.95e-01**	3.77e-01	**4.94e-01**	3.04e-01	**4.95e-01**
	std	8.96e-02	3.16e-04	5.16e-02	3.44e-04	5.90e-02	3.16e-04	7.19e-02	3.61e-04	3.22e-02	3.16e-04	5.84e-02	2.74e-04
	I_F	1.0000	1.0000	0.0004	0.9325	0.1265	1.0000	0.9983	1.0000	1.0000	1.0000	1.0000	0.9323
Eq-IDTLZ3	mean	1.64e-01	**4.90e-01**	1.51e-01	**4.85e-01**	2.69e-01	**4.90e-01**	1.38e-01	**4.90e-01**	2.26e-01	**4.90e-01**	2.23e-01	**4.86e-01**
	std	7.36e-02	5.21e-03	7.38e-02	1.54e-02	7.34e-02	5.21e-03	5.48e-02	3.93e-03	4.71e-02	5.21e-03	6.50e-02	2.14e-02
	I_F	1.0000	1.0000	0.0004	0.9888	0.1327	1.0000	0.9996	1.0000	1.0000	1.0000	1.0000	0.9888
Wilc. test (I-D-S)		0–0–12		0–0–12		0–0–12		0–0–12		0–0–12		0–0–12	

Table 3.12: HV values and I_F mean values obtained on EQC test problems with NSGA-II.

		CDP-NSGA-II		ATP-NSGA-II		C-NSGA-II		ε-NSGA-II		Imp ε-NSGA-II	
		—	w/grad	—	w/grad	—	w/grad	—	w/grad	—	w/grad
EQC1	mean	1.16e-01	**9.07e-01**	0.00e+00	**9.02e-01**	1.59e-02	**9.06e-01**	4.67e-01	**9.07e-01**	3.32e-01	**8.82e-01**
	std	1.67e-01	3.00e-04	0.00e+00	2.34e-03	6.67e-02	5.13e-04	1.50e-01	3.00e-04	1.51e-01	1.27e-02
	I_F	0.9355	1.0000	0.0000	0.7348	0.0039	0.9277	0.9355	1.0000	0.9355	0.5639
EQC2	mean	3.64e-01	**9.10e-01**	7.02e-02	**9.10e-01**	4.57e-01	**9.10e-01**	7.33e-01	**9.10e-01**	5.85e-01	**9.10e-01**
	std	1.74e-01	1.72e-04	1.71e-01	2.73e-04	1.99e-01	2.28e-04	1.21e-01	1.72e-04	1.32e-01	3.30e-04
	I_F	1.0000	1.0000	0.0019	0.9890	0.3342	0.9997	1.0000	1.0000	1.0000	0.9835
EQC3	mean	3.36e-01	**7.76e-01**	5.53e-02	**7.72e-01**	3.31e-01	**7.76e-01**	5.68e-01	**7.76e-01**	4.64e-01	**7.63e-01**
	std	1.62e-01	3.42e-03	1.42e-01	2.86e-03	1.20e-01	3.11e-03	9.38e-02	3.42e-03	1.43e-01	3.81e-03
	I_F	1.0000	1.0000	0.0016	0.9894	0.3384	0.9994	1.0000	1.0000	1.0000	0.4981
EQC4	mean	7.38e-01	**8.22e-01**	3.76e-02	**8.22e-01**	7.07e-01	**8.21e-01**	7.45e-01	**8.22e-01**	7.51e-01	**8.21e-01**
	std	4.56e-02	2.32e-03	9.41e-02	2.31e-03	1.04e-02	2.60e-03	1.17e-02	2.06e-03	9.80e-03	2.15e-03
	I_F	1.0000	1.0000	0.0006	0.9409	0.1333	0.986	1.0000	1.0000	1.0000	0.8506
EQC5	mean	3.00e-01	**7.77e-01**	0.00e+00	**7.43e-01**	3.52e-01	**7.70e-01**	2.40e-01	**7.77e-01**	2.52e-01	**7.77e-01**
	std	1.43e-01	6.86e-05	0.00e+00	1.65e-02	1.68e-01	2.03e-03	1.74e-01	7.87e-05	1.78e-01	9.08e-05
	I_F	1.0000	1.0000	0.0000	0.2889	0.0765	0.5703	0.9798	1.0000	0.9677	1.0000
EQC6	mean	0.00e+00	**1.03e+00**	0.00e+00	0.00e+00	0.00e+00	**9.90e-01**	0.00e+00	**1.04e+00**	0.00e+00	**2.01e-01**
	std	0.00e+00	1.32e-02	0.00e+00	0.00e+00	0.00e+00	4.29e-02	0.00e+00	8.26e-04	0.00e+00	4.17e-01
	I_F	0.0000	1.0000	0.0000	0.0000	0.0000	0.5400	0.0000	1.0000	0.0000	0.1935
Eq-DTLZ1	mean	5.28e-01	**8.36e-01**	5.24e-02	**8.37e-01**	6.67e-01	**8.35e-01**	6.03e-01	**8.36e-01**	5.85e-01	**8.37e-01**
	std	1.09e-01	3.17e-03	1.07e-01	2.75e-03	1.01e-01	2.60e-03	8.85e-02	3.17e-03	1.01e-01	2.30e-03
	I_F	1.0000	1.0000	0.0013	0.9851	0.1691	1.0000	1.0000	1.0000	1.0000	0.9684
Eq-DTLZ2	mean	5.13e-01	**6.52e-01**	5.50e-02	**6.55e-01**	5.62e-01	**6.52e-01**	6.00e-01	**6.52e-01**	5.85e-01	**6.55e-01**
	std	5.84e-02	3.12e-03	7.37e-02	2.75e-03	3.24e-02	4.93e-03	1.78e-02	3.12e-03	3.86e-02	2.54e-03
	I_F	1.0000	1.0000	0.0019	0.9770	0.1459	1.0000	1.0000	1.0000	1.0000	0.9633
Eq-DTLZ3	mean	3.73e-01	**6.33e-01**	7.43e-02	**6.36e-01**	4.52e-01	**6.36e-01**	4.01e-01	**6.33e-01**	4.28e-01	**6.39e-01**
	std	7.50e-02	7.36e-03	9.14e-02	8.69e-03	7.19e-02	5.95e-03	7.33e-02	7.36e-03	5.33e-02	7.57e-03
	I_F	1.0000	1.0000	0.0019	0.9828	0.1477	1.0000	1.0000	1.0000	1.0000	0.9690
Eq-IDTLZ1	mean	1.48e-01	**3.68e-01**	4.82e-02	**3.69e-01**	2.10e-01	**3.68e-01**	1.76e-01	**3.68e-01**	1.65e-01	**3.68e-01**
	std	4.81e-02	2.50e-03	5.20e-02	2.09e-03	3.77e-02	2.72e-03	3.62e-02	2.50e-03	5.24e-02	2.72e-03
	I_F	1.0000	1.0000	0.0028	0.9881	0.1691	0.1912	1.0000	1.0000	1.0000	0.9694
Eq-IDTLZ2	mean	3.51e-01	**4.95e-01**	2.80e-02	**4.95e-01**	4.32e-01	**4.95e-01**	4.34e-01	**4.95e-01**	4.16e-01	**4.95e-01**
	std	4.91e-02	8.11e-04	4.39e-02	3.86e-04	1.44e-02	6.80e-04	2.54e-02	8.11e-04	3.58e-02	7.18e-04
	I_F	1.0000	1.0000	0.0013	0.9815	0.1476	1.0000	1.0000	1.0000	1.0000	0.9720
Eq-IDTLZ3	mean	2.73e-01	**4.74e-01**	3.43e-02	**4.78e-01**	3.21e-01	**4.77e-01**	3.22e-01	**4.74e-01**	3.05e-01	**4.78e-01**
	std	4.36e-02	1.39e-02	5.31e-02	7.94e-03	4.01e-02	9.26e-03	3.20e-02	1.39e-02	4.84e-02	1.11e-02
	I_F	1.0000	1.0000	0.0014	0.9874	0.1531	1.0000	1.0000	1.0000	1.0000	0.9789
Wilc. test (I-D-S)		0–0–12		0–1–11		0–0–12		0–0–12		0–0–12	

Concerning problem EQC3, it constitutes a more difficult problem as the true Pareto front is disconnected. For the MOEA/D algorithm, it is quite difficult to consistently determine the disconnected segment of $\mathcal{PF}$, even though the gradient-based repair method improves the performance of the original algorithms.

Problems EQC4-5 (Cuate et al., 2020a) involve three-objective functions, with only three decision variables and one and two equality constraints, respectively. For these problems, the same diversity issue as for problems EQC1-3 is observed: the canonical constraint-handling techniques lack the capacity of continuing exploring regions different from those found first, corresponding to a reduced part of the true PF. The search process gets trapped in equality-constrained search space regions. This phenomenon is displayed in Figure 3.8 (left) for problem EQC5, clearly illustrating how the Stochastic Ranking without repair barely identifies a few points on $\mathcal{PF}$, while incorporating the repair strategy allows to find the complete front. Therefore, the effect of the gradient-based repair strategy is that new individuals may survive and explore in a smooth way these equality-constrained search spaces, with the possibility of visiting promising regions, as observed in Figure 3.8 (right) for the same problem. The numerical results shown in Tables 3.9 to 3.12 highlight the ability of the gradient-based repair to achieve the best possible approximation of the true PF, for the population size and diversity criterion used. As previously, even the methods that exhibited a poor performance in their canonical form, like ATP or SR, have now an excellent performance once the repair process is carried out.

Problem EQC6 (Pintaric and Kravanja, 2006; Rangaiah, 2009) is the so-called Williams-Otto process optimization problem. It has ten decision variables and six equality constraints, and thus can be considered as the most difficult problem treated here (with respect of constraint satisfaction). The formulation considered involves the maximization of two objectives: the net present value (NPW) and the profit before taxes (PBT). As observed in Table 3.11, none of the canonical constraint-handling method is able to find any single feasible solution in any run, not even the CDP method, whose priority is yet the search for feasible solutions, all along the evolutionary process. The satisfaction of six equality constraints involving ten decision variables seems to be a very difficult task for these algorithms. But once again, the gradient information of constraints permits to find the Pareto optimal front. A revisited formulation of this problem was used elsewhere for problem solution (Ouattara, 2011; Rangaiah, 2009): at each evaluation of the objective function, all equality constraints are satisfied by solving a system of nonlinear equations. This methodology, although computationally intensive, has proven to be efficient for solving the problem. For comparison purposes, we have implemented this solution strategy and it showed CPU times approximately 30 times higher than those reported with the repair performed here. Besides, that reformulation methodology cannot be applied as a general-purpose method.

Benchmark problems Eq-DTLZ and Eq-IDTLZ are scalable in the number of decision variables, objectives functions and equality constraints (Cuate et al., 2020b). They are built from the DTLZ problems (Deb et al., 2005) and hyper-spheres are added as equality constraints. We only considered here the three objective problems, involving one equality constraint that forms a circle in the search space of the variables x_1 and x_2. The Eq-DTLZ1 and Eq-IDTLZ1 contain seven decision variables, whereas the rest contain twelve decision variables. According to the results, the canonical techniques present difficulties for preserving diversity in the objective space, i.e., offspring solutions, as a product of recombination operators, are likely to be infeasible

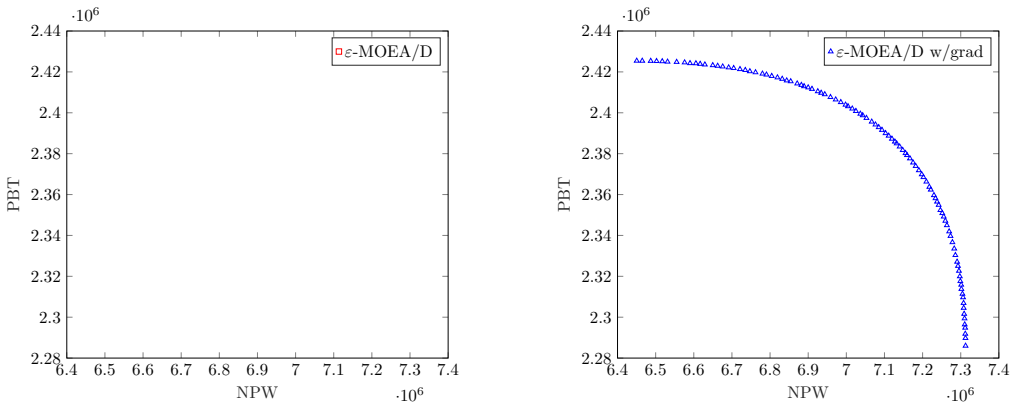


Figure 3.9: Final Pareto front approximation of EQC6 function. ε-MOEA/D without (left; no feasible solution found) and with (right) gradient-based repair.

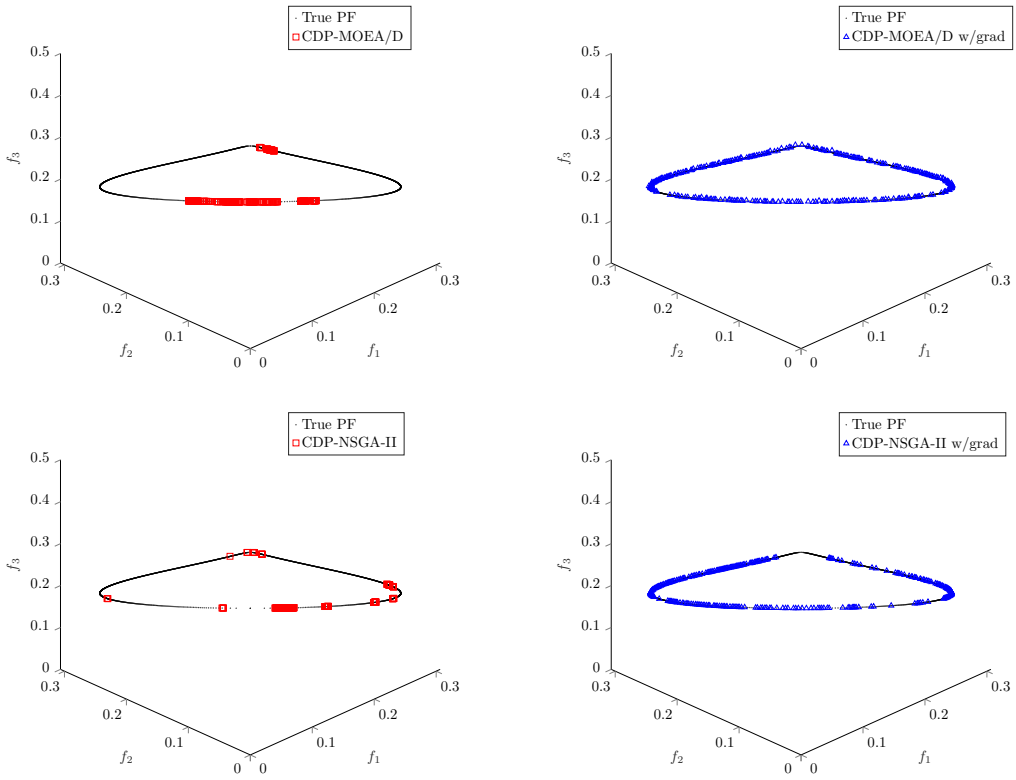


Figure 3.10: Final Pareto front approximation of Eq-DTLZ1 function. CDP-MOEA/D and CDP-NSGA-II without (left) and with (right) gradient-based repair.

because of the narrow feasible region. Thus, the evolutionary process is stopped at early generations once parents converge to only some regions of the $\mathcal{PF}$. Conversely, the gradient-based repair enable the population to preserve diversity by repairing offspring solutions that might be at the vicinity of the Pareto front but still being infeasible. In this way, promising but infeasible solutions are now repaired and properly evaluated by the MOEA in the selection step. Numerical and graphical results (see Table 3.9 to 3.12 and Figure 3.10) consistently indicate the superiority of the methods employing the gradient-based repair of solutions.

3.6 Conclusions

This chapter presented a comparative study highlighting the benefits of using the constraints' gradient information to repair infeasible solutions when solving CMOPs. This strategy was embedded and combined with six state-of-the-art constraint-handling schemes, classically used within MOEA/D and NSGA-II: constraint dominance principle, penalty function (ATP), C-MOEA/D, stochastic ranking, ε-constraint and improved ε-constraint. The use of the constraints' gradient-based repair strategy significantly enhances the canonical algorithms over the test suites studied here (CF, LIRCMOP, EQC, Eq-DTLZ and Eq-IDTLZ). These instances allowed to evaluate and analyze the performance of these techniques for different problem features. Indeed, some functions contain a significant number of local optima (both in the objective space and in the constrained search space), that may be bounded by inequality constrained regions producing islands in the search space or disconnected true Pareto fronts enclosed by infeasible regions. Also, equality constrained problems are considered (which is rather unusal in the study of CMOPs), which generate particular difficulties to reach the whole $\mathcal{PF}$.

The obtained results have shown that the information of constraints' gradient can make the canonical constraint-handling methods much more robust, enabling the population to get across infeasible regions and promoting the diversity during the construction of the Pareto front approximation. Besides, the results observed in the test problems containing equality constraints allow to conclude that repairing infeasible individuals is particularly useful for problems with narrow feasible spaces, as those produced by equality constraints, in which canonical methods do not work properly. The repair procedure considered here repairs those promising solutions that are infeasible, so that the MOEA can continue searching for a well-distributed approximation of the Pareto front. Moreover, when the computation of the gradient is performed, the computational time does not significantly increase compared to that of canonical algorithms, at least for problems containing up to 30 decision variables.

Additionally, the work carried out in this chapter promote the research on the use of mathematical properties (like the gradient) that can be exploited in order to enhance the search process and guide the MOEA population towards promising regions. The encouraging results obtained with the gradient-based repair strategy on the academic benchmarks tackled here thus lead to apply a similar methodology for the solution of complex PSE problems. Therefore, the following chapters are devoted to the extension of the present study to a real-world engineering problem, the optimal design of hydrogen supply chains (HSC), that encompasses both equality and inequality constraints.

Part II

Application to the optimal design of hydrogen supply chains

Chapter 4

Optimal design of hydrogen supply chains: general context

Contents

After a first part devoted to the empirical evaluation of computational techniques adapted to the solution of constrained multiobjective optimization problems, the lessons learned from this numerical study are now applied in this second part to a real-world PSE problem, namely the sustainable design of hydrogen supply chains. This chapter provides the general background regarding this topic, setting the basis for the proper understanding of the subsequent chapters, offering the reader related information about the challenges of hydrogen deployment. First, the relevance of hydrogen for providing clean solutions in the energy transition context is discussed in Section 4.1. Then, in Section 4.2, the application to the mobility sector is described and the main features of the hydrogen supply chain are presented. A brief description of the design problem, highlighting the specific issues that arise for designing the optimal hydrogen supply chains is developed in Section 4.3. Finally, the conclusions and the work orientation are presented in Section 4.4.

4.1 Hydrogen context

4.1.1 The role of hydrogen for global decarbonization

The growing general concern about the depletion of conventional energy sources, such as oil and gas, as well as the degradation of the environment caused by the combustion of these fossil fuels, have motivated the search for a more sustainable energy model based on renewable energy systems. In this context, hydrogen stands as a potential low-carbon alternative because it can play multiple roles in the energy transition (see Figure 4.1). These can be summarized in the following points:

- Hydrogen can be produced from renewable sources like biomass, wind or solar energy, or from a variety of fossil fuels coupled with processes that capture, utilize or storage the by-product CO_2.

- It can be used in multiple sectors such as industry, building and transportation, providing meaningful reductions of CO_2 emissions.

- Hydrogen can hence store surplus power from renewable sources when the grid cannot absorb it (IEA, 2019).

According to the Hydrogen Council (2017), hydrogen is expected to cover 18% of global energy demand by 2050. These long-term estimates are in part established considering the government policies to limit the global greenhouse gases (GHG) emissions, in line with the 2015 Paris Climate Agreement. Besides, in 2019, the European Green Deal set up a framework of regulation and legislation with targets to reach net zero global warming emissions by 2050. Within this agreement, hydrogen is considered a key instrument for meeting the Green Deal objectives. Further, energy storage is often viewed as an electrical power storage through mechanical, electrical and electrochemical storage systems. In the current energy system, grid-scale energy storage is typically short-term and used to maintain stability, in order to address peaks (i.e., on the minute and hour scale) up to daily imbalances. Seasonal storage may be needed in the future for high levels of renewable penetration based mainly on solar and wind generation, and can be achieved at terawatt (TW) level through hydrogen or synthetic methane (Davies et al., 2020). This is why hydrogen is mentioned as one of the best options to store electrical energy, even better than pumped hydro, compressed air or batteries. Beyond its potential role in providing chemical storage of electricity, hydrogen can also act as an energy carrier for energy and industrial applications where it is difficult to replace fossil fuels, thus contributing to the decarbonization of transportation, buildings or industry. Hydrogen can thus be viewed as a "coupling sector" technology (Brey, 2020).

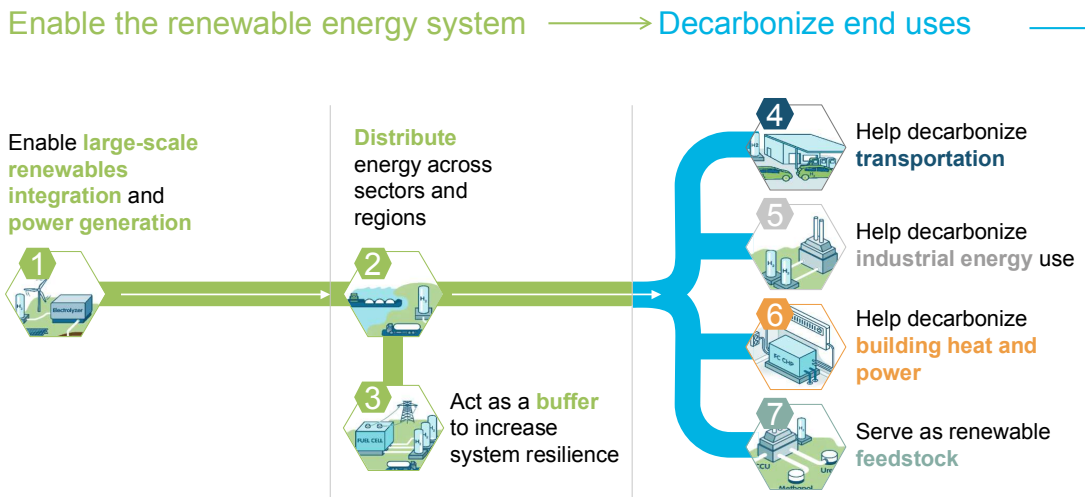


Figure 4.1: Hydrogen can play 7 roles in the energy transition. Taken from Hydrogen Council (2017).

4.1.2 Hydrogen as a driver for the transport decarbonization

In this respect, the transportation sector is of high importance because it is responsible for 26% of GHG emissions in EU, 28% in U.S., and 23% worldwide in recent years (European Environment Agency, 2018; Sims et al., 2014), which is in part explained by the fact that this sector relies almost completely on oil (Hydrogen Council, 2017). Consequently, several technologies have been proposed with the aim to decarbonize this sector, and particularly, the road transportation sector, including hybrid electric vehicles, battery electric vehicles (BEVs) and fuel cell electric vehicles (FCEVs). Regarding BEVs, on the one hand, they are at present in the market mainly because the electric grid is already available in most areas where cars typically need to be charged. Nevertheless, they present primarily two drawbacks, namely, high charging times, and for medium-to-large size vehicles, the need for heavy batteries. On the other hand, it is widely recognized that FCEVs are a necessary complement to BEVs, as FCEVs typically permit users longer ranges and fast fueling times, in comparison to BEVs (Lin et al., 2018), with the drawback that little or none fueling stations are yet available in most areas. Several studies have been conducted showing that BEV and FECV can provide climate benefits, though results depend strongly on several factors including the electrical mix used for battery charging and hydrogen production, the lifetime distance traveled by the vehicle, and the vehicle energy consumption. A Life Cycle Assessment is thus necessary to have a global overview of all the steps involved from well-to-wheel (see Cox et al., 2020).

Hence, even if the potential environmental and technological benefits of hydrogen in the transportation sector are encouraging, the shift to a hydrogen-based economy is still a challenge. Much of the future expansion of hydrogen utilization depends not only on technological developments and energy policies, but also on the hydrogen supply chain (HSC) deployment.

4.2 Hydrogen supply chain for mobility

As a general definition, a supply chain is an integrated network of facilities and transportation options for the supply, manufacture, storage, and distribution of materials and products (Garcia and You, 2015). An optimally designed supply chain should, through one or a variety of metrics, reflect the "best" configuration and operation of all of these elements (ibid.). In this sense, the optimal design of the hydrogen supply chain is defined as the infrastructure required to manufacture, store and deliver hydrogen to the consumer, so that one or multiple criteria are minimized. In this way, production processes are required to convert primary energy resources into hydrogen; storage units and terminals are needed to compensate for fluctuations in demand; distribution systems are essential for transporting hydrogen from the production facilities to the point of sale; and finally, dispensing or refueling stations allow the transfer of hydrogen to users at forecourt retail stations (Almaraz et al., 2013; Almaraz et al., 2015; Hugo et al., 2005; IEAGHG, 2017).

At each of the stages along the supply chain, a wide variety of potential technological options exist, as represented in Figure 1 in page 5. Thus, not only can hydrogen be manufactured from a variety of primary energy feedstocks, but it can also be distributed in a variety of physical forms using different technologies. Each one of the technological pathway options has its own unique advantages and disadvantages, and it must be emphasized that the degree of maturity is not the same among the

technologies. In what follows, the some technological options are explained in more details, they will be used in the case studies treated in the subsequent chapters.

4.2.1 Hydrogen production

The local market conditions and availability of regional primary energy feedstock have a large impact on the selection of supply chains pathways. As previously explained, hydrogen has the benefits of improving security of fuel supplies since it can be produced from diverse primary energy sources, such as fossil fuels and renewable energy sources, i.e., wind, biomass, or solar energy (Almaraz et al., 2014a; Valente et al., 2020). In the following, two important hydrogen production technologies are presented, namely, steam methane reforming and water electrolysis.

4.2.1.1 Steam reforming

Steam methane reforming (SMR) is the most widespread technology for hydrogen production from natural gas at large scale, since it has the lowest capital costs of the hydrogen production. This production process uses a catalyzer (typically nickel) to facilitate the thermochemical reaction of natural gas and water at temperatures of around 850°C and a pressure of 2.5 MPa (Sørensen and Spazzafumo, 2018; Velazquez Abad and Dodds, 2017). The methane found in natural gas reacts with steam to produce a synthesis gas consisting of hydrogen and carbon monoxide, represented as follows:

$$CH_4(g) + H_2O(g) \rightarrow CO(g) + 3H_2(g), \quad \Delta H = 206\,\text{kJ mol}^{-1} \tag{4.1}$$

In order to obtain a high conversion efficiency, the syngas is saturated with further steam which yields additional hydrogen as the CO is converted to CO_2:

$$CO(g) + H_2O(g) \rightarrow CO_2(g) + H_2(g), \quad \Delta H = -41\,\text{kJ mol}^{-1} \tag{4.2}$$

The heat required in equation (4.1) is obtained from the heat surplus produced in reaction described in equation (4.2) and by burning a proportion of the methane feedstock (typically 30–40% of it) (IEA, 2019; IEAGHG, 2017). The global SMR reaction is thus endothermic (165 kJ mol^{-1}) and it yields 4 mol of H_2 and 1 mol of CO_2 for each one of methane (see equations 4.1 and 4.2). Hydrogen produced through SMR has an average carbon intensity of 328 gCO_2-eq/kWh (National Academy of Engineering, 2004). This fossil type of hydrogen is commonly referred to as "grey hydrogen". Nevertheless, significant reductions of these carbon emissions could be achieved using CCUS technologies.

The production cost of hydrogen from natural gas is influenced by various technical and economic factors, with gas prices and capital expenditure (CAPEX) being the two most important. The data used in this work regarding SMR production technology is provided in Appendix B.

Low-carbon alternative: CCUS

Given the fact that SMR processes entail significant CO_2 emissions, a promising option is the integration of carbon capture, utilization and storage (CCUS). This can be carried out in several ways. For instance, CO_2 can be separated from the

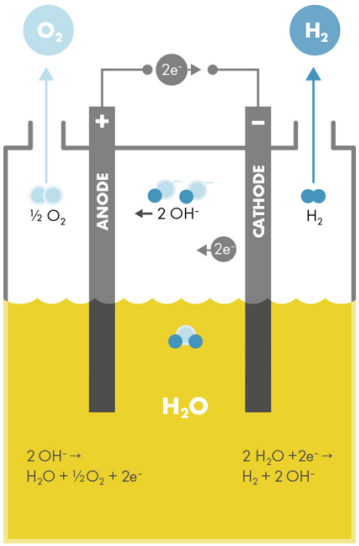


Figure 4.2: Basic water electrolysis process. Taken from *Electrolysers* (n.d.).

high-pressure syngas stream, reducing emissions by up to 60%. CO_2 can also be captured from the more diluted furnace flue gas. This can boost the level of overall emission reduction to 90% or more (Dodds, 2015; IEA, 2019; IEAGHG, 2017).

Adding CCUS to SMR plants leads, in average, to cost increases of about 50% for investment costs and about 10% for fuel, the exact amounts depending on the design. It also leads on average to a doubling of operational cost as a result of CO_2 transportation and storage costs (IEA, 2019). The data used in this work regarding SMR w/CCUS production technology is presented in Appendix B.

4.2.1.2 Water electrolysis

Water electrolysis is a process that consists in splitting water into hydrogen and oxygen. In its basic form, electrolyzers are composed of an anode, a cathode and a conductive medium (electrolyte), as represented in Figure 4.2.

An electric current generates a flow of positive-charged ions (such as hydrogen) to the cathode where these electrons are reduced. Conversely, negatively charged ions move to the anode losing electrons and oxidizing. In the process, the water breaks down, generating hydrogen and oxygen (Dodds, 2015; Sørensen and Spazzafumo, 2018). The overall process is represented by the following chemical equation:

$$H_2O(l) \rightleftharpoons H_2(g) + \frac{1}{2}O_2(g), \quad \Delta H = 286\,\text{kJ mol}^{-1} \tag{4.3}$$

A key advantage of electrolysis over fossil-based technologies is the high purity of the produced hydrogen (greater than 99.999%), which is suitable for powering fuel cells without further purification. Besides, as electricity generation from solar, wind and wave is intermittent, the deployment of high levels of these technologies might lead to significant disparities between electricity supply and demand at different times. One option is to integrate electrolysis technologies into national energy systems, so that excess electricity is converted into hydrogen through power-to-gas processes (Carrera Guilarte and Azzaro-Pantel, 2020). In this vein, electrolyzers can offer a

flexible load that providing low-cost balancing services to the power system while producing hydrogen for mobility applications, industrial uses, or injection to the gas grid. Two main electrolyzer technologies that are mature today are alkaline electrolysis and proton exchange membrane electrolysis. Their main technical and economic characteristics are described in the following.

Alkaline electrolysis

Alkaline electrolysis (AEL) is a mature technology for H_2 production up to the MW scale, and represents the most widely used electrolytic technology on a commercial level worldwide. In alkaline electrolyzers a direct current is applied between an anode and a cathode submerged in an electrolyte (an alkaline solution such as sodium or potassium hydroxide), which causes hydrogen to form in the cathode and oxygen in the anode (see Figure 4.4). The hydrogen production rate is proportional to the current passing through the electrodes. The reaction is endothermic and reversible at ambient temperature (Velazquez Abad and Dodds, 2017), since the diffusion of oxygen into the cathode compartment reduces the efficiency of the electrolyzer, reacting with the hydrogen present on the cathode side to form water (Miller et al., 2020).

Commercial electrolyzers have a hydrogen production efficiency of 68%–80%, which depends on the cell voltage (typically 1.9–2.2 V), temperature (343–363 K), electrolyte flow conditions, and the operating pressure (Hanke-Rauschenbach et al., 2015). AEL cannot work below specific loads (typically below 30%) for safety reasons because the O_2 production rate decreases, thus increasing the H_2 cross-over concentration to dangerous levels (lower explosion limit >4 mol% H_2) (Miller et al., 2020; Taibi et al., 2018).

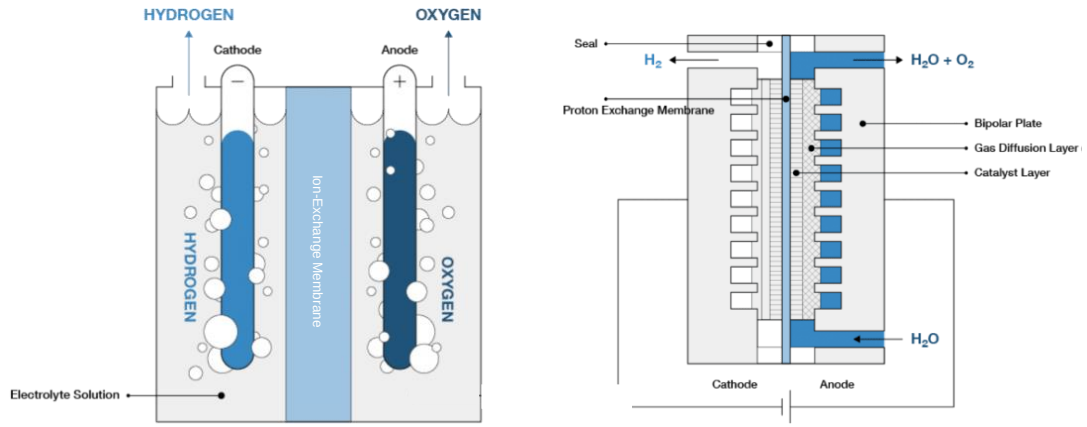


Figure 4.3: Comparison of water electrolysis cells and chemistries using either an AEL (left) or a PEM (right). Taken from Cummins Inc. (n.d.).

Proton exchange membrane electrolysis

Proton exchange membrane (PEM) are electrolyzers smaller than AEL as the electrolyte is a solid polymer material rather than a liquid. Oxygen from the water molecules and positively charged hydrogen ions are formed in the anode. The electrons flow through an external circuit and the hydrogen ions move to the cathode through the membrane, where they combine with the electrons to form hydrogen gas

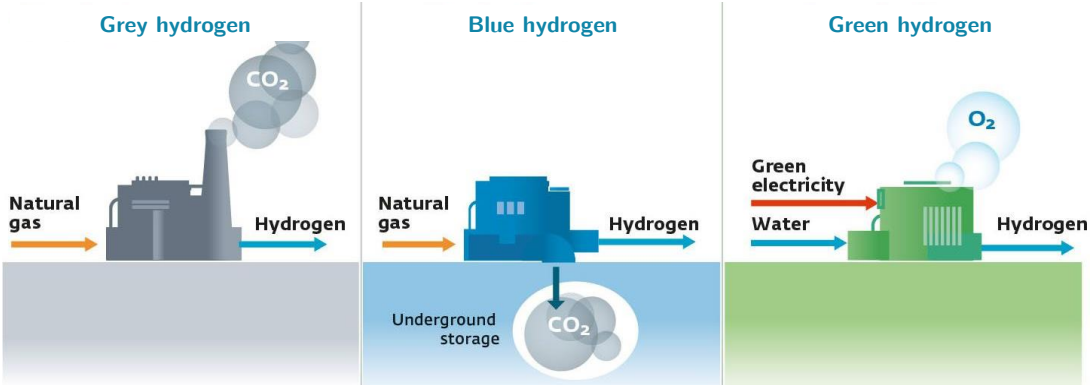


Figure 4.4: Comparison of different hydrogen production pathways and their corresponding color labels. Taken from *theworldofhydrogen* (n.d.).

(Velazquez Abad and Dodds, 2017), see Figure 4.4. PEM electrolyzers have a faster dynamic response and wider load ranges than alkaline electrolyzers; however, they have higher capital costs as they require expensive catalysts (IEA, 2019). Systems can be maintained in stand-by mode with minimal power consumption and are able to operate for a short time period (10–30 minutes) at higher capacities than the nominal load, i.e., beyond 100%, and up to 200% (Eichman and Flores-Espino, 2016; Taibi et al., 2018).

4.2.1.3 On hydrogen colors

In recent years, colors have been used to refer to different sources of hydrogen production. "Black", "grey" or "brown" refer to the production of hydrogen from coal, natural gas and lignite respectively. "Blue" is commonly used for the production of hydrogen from fossil fuels with CO_2 emissions reduced by the use of CCUS. "Green" is a term applied to production of hydrogen from electrolysis powered by renewable electricity. In general, there are no established colors for hydrogen from biomass, nuclear or different varieties of grid electricity. Nevertheless, color terminology is to be used carefully as the environmental impacts of each of these production routes can vary considerably by energy source, region and type of CCUS applied; they range from 328 gCO_2-eq/kWh for hydrogen produced from natural gas, and can be as low as 32 gCO_2-eq/kWh for hydrogen produced by wind-powered electrolysis (Bhandari et al., 2014).

4.2.1.4 Centralized and distributed hydrogen production

Besides, unlike most other fuel infrastructures, hydrogen can be produced either centrally or in a distributed manner. A centralized production option would be analogous to current gasoline supply chains, where the economy of scale are capitalized upon within an industrial context and large quantities are produced at a central site and then distributed. Centralized plants promise higher hydrogen production efficiency but require more capital investment and a substantial hydrogen transport and delivery infrastructure.

Alternatively, through the use of electrolyzers, hydrogen can be produced closer to the point of use, i.e. on-site, in smaller quantities. Such a scenario would exploit

the production of hydrogen at the forecourt refueling stations, thereby alleviating the significant transportation cost (Hugo et al., 2005; Hydrogen Council, 2020; IEA, 2019). However, a decentralized approach often results in higher costs as production efficiencies are generally lower and because on-site production facilities are often dimensioned to cover peak demand, especially when no storage is foreseen or possible (Haeseldonckx and D'haeseleer, 2011).

4.2.2 Hydrogen transportation

As hydrogen is to play a meaningful role in clean and flexible energy systems, delivery infrastructure choices and costs are thus critically important. Three main options exist for hydrogen distribution: 1) trucking of compressed hydrogen, 2) trucking of liquefied hydrogen, and 3) use of pipelines. Choosing the best option depends on different parameters such as the distance of the demand center from the production site, the amount of transferred hydrogen and the existing infrastructure (Almaraz et al., 2015; Hydrogen Council, 2020; IEA, 2019).

Besides, hydrogen can be transported in gas or liquid form, but compressed hydrogen (at 700 bar pressure) has only 15% of the energy density of gasoline, so transporting the equivalent amount of energy would require nearly seven times the same space. Liquid hydrogen tanker trucks are often used instead where there is reliable demand and the liquefaction costs can be offset by the lower unit costs of hydrogen transport (IEA, 2019). Highly insulated cryogenic tanker trucks can carry up to 4 000 kg of liquefied hydrogen, and are commonly used today for local journeys and up to 4 000 km. According to IEA (ibid.), it is reasonable to expect that, over the next decade, compressed gas tube trailers and liquid hydrogen tanks will remain the main distribution modes, just as distribution of gasoline and diesel to geographically dispersed refueling stations is mostly carried out using trucks today.

In many countries there is an extensive natural gas pipeline network that could be used to transport and distribute hydrogen. Also, new infrastructures could also be developed with dedicated pipeline network, potentially allowing large-scale overseas hydrogen transport (Hydrogen Council, 2017; IEA, 2019). In this work, hydrogen transport through pipelines is not considered because of the size (relatively small) of the geographic region studied.

4.2.3 Hydrogen storage and distribution

Today hydrogen is most commonly stored as a gas or liquid in tanks for small-scale mobile and stationary applications. The most appropriate storage medium depends on the volume to be stored, the duration of storage, the required speed of discharge, and the geographic availability of different options. Tanks storing compressed or liquefied hydrogen have high discharge rates and efficiencies of around 99%, making them appropriate for smaller-scale applications where a local stock of fuel or feedstock needs to be readily available (IEA, 2019; Taibi et al., 2018).

4.3 The problem of hydrogen supply chain design

Accounting for the framework in the previous sections, the HSC design problem consists of a four-echelon supply chain for transportation sector (energy source–production–transportation–storage and distribution). The network considers several

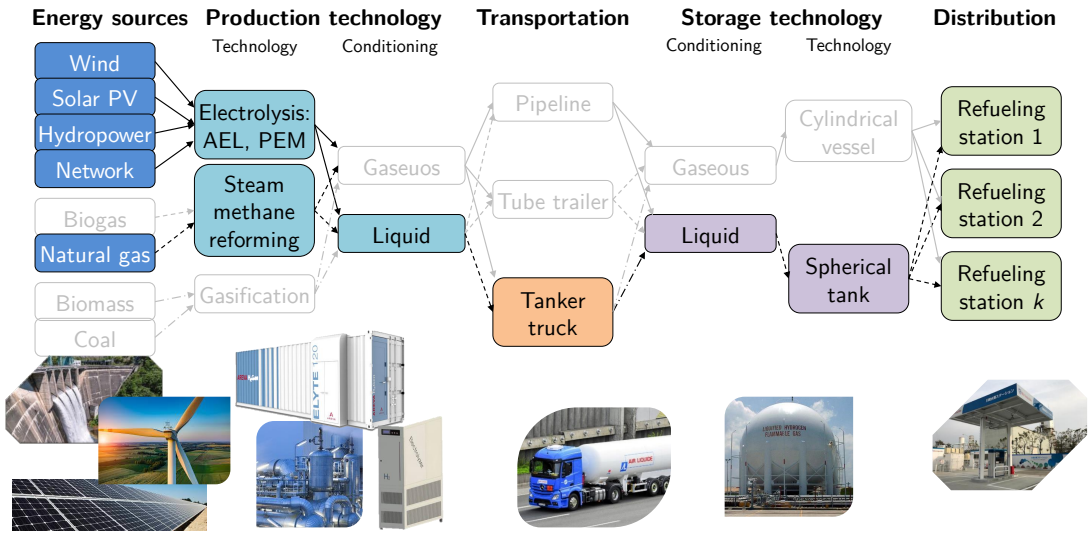


Figure 4.5: Hydrogen supply chain technologies considered in this work.

energy sources as feedstock to different production technologies, each production technology having different possible sizes with corresponding capacity bounds, capital costs and product unit costs. A set of options is available for transportation from production sites to storage facilities. Before and after transportation, it can be necessary to condition hydrogen to a suitable physical form for transportation and then, for storage. Depending on this final physical form, several storage technologies are available with different capacity sizes. The supply chain describing the options considered in this work is represented in Figure 4.5.

Besides, the optimal design of the hydrogen supply chain has to take into account not only the economic aspects but also the environmental impacts of each potential alternative in the network, so that a realistic evaluation of the HSC can be done. In this context, the scientific objective of this work is to provide an answer to the following questions:

- Where hydrogen production plants, storage units should be located to meet a demand for hydrogen in specific time periods by minimizing costs and/or environmental impacts?
- How does the electricity from renewable sources impact the selection of grids for hydrogen production?

In the next chapters, these questions will be addressed by appropriate mathematical modeling and optimization tools.

4.4 Conclusions

In this chapter, the concept of the hydrogen supply chain (HSC) has been introduced with a particular focus on mobility applications. This information highlighted the fact that the hydrogen supply chain for mobility application results in a more complex network than that for on-site industrial applications. A hydrogen supply chain involves technological bricks for energy sources, production, transportation and

storage, and refueling stations. All these activities need to be integrated in a systematic modeling framework that will be the core of the next chapters.

Chapter 5

Optimal design of hydrogen supply chains: strengths and weaknesses of multiobjective evolutionary algorithms

Contents

5.1 Introduction

After the general presentation describing, in the previous chapter, the framework of HSC for the mobility sector, this chapter is devoted to the corresponding design problem, formulated as a constrained multiobjective optimization problem. Such a problem has to take into account the different sources of energy from which hydrogen can be produced, the different production and storage technologies available, the location of hydrogen plants and storage facilities, their respective production and storage capacities and the mode and rate of transportation between production units and storage facilities in the network. Besides, the design of a cost-efficient hydrogen

infrastructure has to consider the varying demand over time, so that a multi-period model has to be employed.

Due to the aforementioned characteristics, the HSC design problem is often formulated as a difficult optimization problem (NP-hard problem) (Yoo et al., 2010), involving both discrete and continuous variables: the former account for the existence and location of production and storage facilities of a specific capacity, while the latter represent the hydrogen flow rates from production to storage units in order to satisfy a given demand. Moreover, from a perspective of sustainability, not only cost minimization but also environmentally-friendly production modes need to be accounted for, leading to a multiobjective formulation of the problem.

This chapter is organized as follows. In Section 5.2, an extensive literature review of related works is presented, with a particular focus on solution approaches. In Section 5.3, the main characteristics and assumptions of the problem under study are described, along with a mathematical model representing the sustainable HSC design problem. Note that this model is drawn from previous studies and that the main contribution of this chapter is the development of a metaheuristic-based solution technique. Therefore, the research is devoted to investigate the performance of a multiobjective evolutionary algorithm to tackle this real-world problem. Particularly, the performance of the canonical version of MOEA/D, coupled with the gradient-based repair as a constraint handling technique, is examined. The obtained results are presented and discussed in Section 5.6. Finally, the conclusions drawn in Section 5.7 point the way to the design of a hybrid approach, developed in the next chapter.

5.2 Literature review

5.2.1 Mathematical programming for mono- and multiobjective optimization

Several approaches have been proposed for the design of hydrogen supply chains in multiple works, each one considering different assumptions and characteristics of the supply chain. One of the first studies proposing mathematical modeling tools in this area is found in Heever and Grossmann (2003). This work focused on the integration of production planning and reactive scheduling for the optimization of a refinery hydrogen network. It addressed only the operational level of an existing network, so that the hydrogen supply chain design is out of the scope. The problem is formulated as a multi-period mixed-integer nonlinear program (MINLP) devoted to minimization of costs and solved through exact techniques (DICOPT++, Brooke et al. (1998)).

Later on, in a pioneering work, Hugo et al. (2005) proposed a generic mathematical model, including different primary energy feedstocks, different production and storage technologies, distribution types, potential sites for location of production units, and the dynamic change of hydrogen demand over time. The resulting mixed integer linear programming (MILP) model considered two conflicting objectives (economic and environmental). The authors do not specify the solution approach employed, but it is presumably an exact one.

In Almansoori and Shah (2006), the authors provided a formal model encompassing every echelon of the supply chain with an illustrative case study in UK. Initially, the model only considered a constant deterministic demand with the minimization of

the total cost of the hydrogen infrastructure as the objective function. Subsequently, in Almansoori and Shah (2009), the original model was modified to account for multiple periods, each one with a different hydrogen demand, and then, uncertainty on demand was taken into consideration in Almansoori and Shah (2012). Besides, this model has been extended in several works to consider multiple objectives simultaneously, namely, the greenhouse gas emissions (GHG) and safety, with case studies in Germany (Almansoori and Betancourt-Torcat, 2016), France (Almaraz et al., 2015; Almaraz et al., 2014b), Korea (Kim and Moon, 2008) and Portugal (Câmara et al., 2019). It must be highlighted that, regarding the handling of multiple objectives, the solution approach adopted in all these works is always the same one, i.e., ε-constraint method. Therefore, because of the combinatorial aspect of the problem that involves prohibitive computational times, large-scale instances cannot be treated or the accuracy of the Pareto front approximations are generally neglected.

5.2.2 Bilevel decomposition

With the aim of alleviating the numerical difficulties associated to the solution of multiobjective large-scale instances, two significant strategies based on bilevel decomposition have been proposed in Guillén-Gosálbez et al. (2010) and Sabio et al. (2010). In Guillén-Gosálbez et al. (2010), the authors minimize the total cost and an environmental criterion according to the principles of life cycle assessment, related to the supply chain. The original problem is reformulated into a bilevel (master-slave) optimization problem. In the master problem, the integer variables representing the existence of production plants and storage facilities are removed, those associated to the number of transportation units are relaxed (considered as continuous), and auxiliary binary variables are added to represent the selection of each type of production unit, storage facility and transportation mode. The master problem provides as an output the type of production, storage and transport technologies that should be used in the supply chain (it does not indicate their optimal sizes , but only if they will be used or not). Regarding the lower level, it is constituted by the original MILP model, but this latter is solved only for the subset of technologies predicted at the upper level. In this manner, the master and slave problems are solved iteratively until the bounds of each subproblem converge within a specified tolerance. It is important to note that the multiobjective aspect of the original problem remains, in both master and slave subproblems, tackled via the ε-constraint method in both levels. So, in order to speed up the solution process, the authors propose the use of integer and logic cuts in the upper level.

In another notable work (Sabio et al., 2010), the authors considered the total cost and financial risks of the HSC as the two objectives to be minimized. They also proposed a master-slave decomposition approach, based on the assumption that, in practical problems, the continuous relaxation of the integer variables of the full-space model provides tight lower bounds. Consequently, the master subproblem solves the relaxed original model by reformulating all the integer variables as continuous, while the slave problem solves the original problem but considers the relaxed variables in the master problem as parameters once they have been rounded to the next integer. This decomposition strategy allows obtaining lower and upper bounds of the original problem efficiently. It must be emphasized that both master and slave problems are still modeled as bi-objective MILP problems that exhibit a reduced complexity in comparison to the original model. The multiobjective aspect in both subproblems is

addressed again using the ε-constraint method.

More recently, in Woo and Kim (2019), the authors employed genetic algorithms coupled with exact techniques to solve the optimal design of the HSC with replenishment cycles, modeled as a mixed-integer nonlinear programming problem (MINLP). A bilevel approach is proposed as a solution strategy, the upper-level being solved by a binary-coded genetic algorithm that handles some variables, in such a way that the lower level solves a linearized model resulting from the upper level variables, considered as parameters. For solving the MILP problem in the lower level, the CPLEX solver is employed. However, the work only addresses a single-objective problem (economic criterion) and several characteristics of the supply chain are not modeled, e.g., availability of different energy feedstock and facility sizes.

Similar approaches to that introduced in Woo and Kim (ibid.) have been proposed in other areas of engineering. The integration of biomass technologies is investigated in a systematic way in Fazlollahi and Maréchal (2013), taking into account the multi-period and multi-criteria features of the problem. The resulting MINLP model considers three objectives to be minimized: the annual investment cost, the operating cost and the overall CO_2 emissions of the system. The solution methodology decomposes the original problem according to a master-slave structure. The upper level is solved using a dominance-based MOEA (named QMOO, Leyland (2002)) that considers the type of district conversion technologies as well as their corresponding maximum sizes (continuous and discrete variables). The lower level, formulated as a MILP problem, optimizes the utilization rate and the operation strategy of district conversion technologies. It is solved using a branch-and-bound technique.

In addition, in Setak et al. (2019), a three-echelon generic supply chain is modeled considering manufacturing plants, distribution centers and multiple retailers with an application dedicated to a pharmaceutical company. The problem, modeled as a Stackelberg game, is formulated as a mixed-integer nonlinear bilevel optimization problem with an objective targeted on distribution centers in the upper level and one devoted to retailers in the lower level. The authors proposed the use of a hybrid algorithm as the solution method as follows: the upper level is solved by a genetic algorithm, which handles all integer and some continuous variables, whereas the lower level solves a quadratic programming problem for each upper-level candidate solution using a deterministic local optimizer.

5.2.3 Main guidelines

Most of the multiobjective models proposed in the literature are solved using deterministic techniques, namely ε-constraint, or by means of utility functions, e.g., weighted sum. It is noteworthy that these methods can guarantee the optimality of the solutions found, and even more, some of them can find any Pareto optimal solution by using appropriate parameters. However, they present two main drawbacks, (1) multiple runs must be performed in order to obtain a set of trade-off solutions, which might be computationally prohibitive and (2) the obtained approximation of the Pareto front is not necessarily uniformly distributed, and thus some regions in the Pareto front might not be adequately explored.

Therefore, several works have been proposed in the literature with the aim of mitigating the computational difficulties evidenced when solving medium-to-large size instances of the multiobjective HSC design problem by a classical approach. These works propose to transform the original problem using a bilevel decomposition.

Nevertheless, since the resulting subproblems are still MILP problems, the ε-constraint method remains being computational prohibitive if more objectives or instances of larger size are to be considered. Other works in other areas of application perform similar model decomposition but use metaheuristic techniques for constructing the Pareto front approximation, and exact methods for solving the MILP lower problem. Regarding the use of MOEAs, most works rely on Pareto dominance-based algorithm, such as NSGA-II.

In this framework, the motivation of this work is to explore an alternative solution approach to the multiobjective HSC problem, with the purpose of efficiently providing the decision maker with a set of well-distributed solutions along the optimal Pareto front. In this chapter, as a first attempt to develop an appropriate solution tool, the MOEA/D algorithm (see Chapter 1 in page 34) is adapted for the solution of the HSC problem. The choice of this specific algorithm is motivated by the conclusions drawn in Part I, suggesting a slight superiority of MOEA/D over NSGA-II, in most of the academic problems studied. In particular, the adapted version of MOEA/D to work on constrained search spaces, i.e., ε-MOEA/D with gradient-based repair, which obtained promising results in Chapter 3, is accounted for in this chapter. But first, the statement of the HSC design problem and a mathematical representation, recognized in the devoted literature, are presented in the next sections.

5.3 Problem statement

In this work, the problem of hydrogen supply chains design is represented through the mathematical model proposed in Almaraz et al. (2015) and Almaraz et al. (2014b), extending the work of Almansoori and Shah (2009). This model represents the hydrogen supply chain considering a time-dependent demand over a geographical area of study, and consists of a five-echelon supply chain for the transportation sector (energy source–production–transportation–storage and distribution). The total daily cost and the global warming potential are to be minimized simultaneously, that is, the objective is to provide decision-makers with a number of different trade-off configurations among the cost efficiency of the supply chain and the associated CO_2 emissions. Within this model, the whole system is designed to provide hydrogen supply over a given territory, which is divided into grids (nodes). Each grid has a hydrogen demand for a given time period (input data) and is considered as a node for potential production and storage locations. Figure 5.1 shows an example of the grid-wise division of a geographical area.

The model is established under the following assumptions:

1. The model is demand-driven and operates at steady-state conditions, contrary to a more realistic dynamic supply-demand model. This clearly implies that the optimal configuration of the network is one that minimizes both the objectives accounted for, subject to demand satisfaction in each grid for each time period.

2. Only some sizes of fixed capacities for production and storage facilities are available.

The HSC problem can therefore be stated as: given a geographical area divided into grids, each grid having an explicit hydrogen demand at each period that must be satisfied, investment and operational cost for production and storage facilities as well

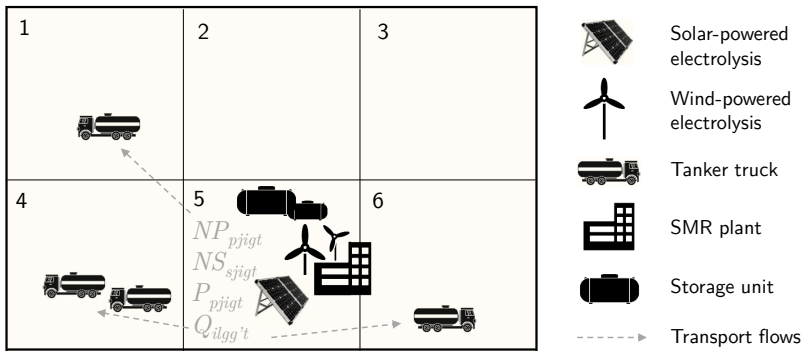


Figure 5.1: Example of grid discretization of geographic area. Here, only grid 5 is developed. $NP_{pjigt}, NS_{sjigt}, P_{pjigt}, Q_{ilgg't}$, stand for production and storage units, production rate, and transportation flows, respectively.

as their capacity constraints, costs and availability of energy sources, transportation costs and environmental information related to the network operation and installation; the objective is to determine the structure of the HSC and the transportation flows within the hydrogen supply chain that minimize its total costs and environmental impact. Such a supply chain is determined by the location, type, capacity and number of production and storage facilities in each grid (nodes); the transportation links (directed arcs) between grids, as well as the type and number of transportation units; production rates for each production plant as well as inventory amounts for storage units. This is depicted in Figure 5.2.

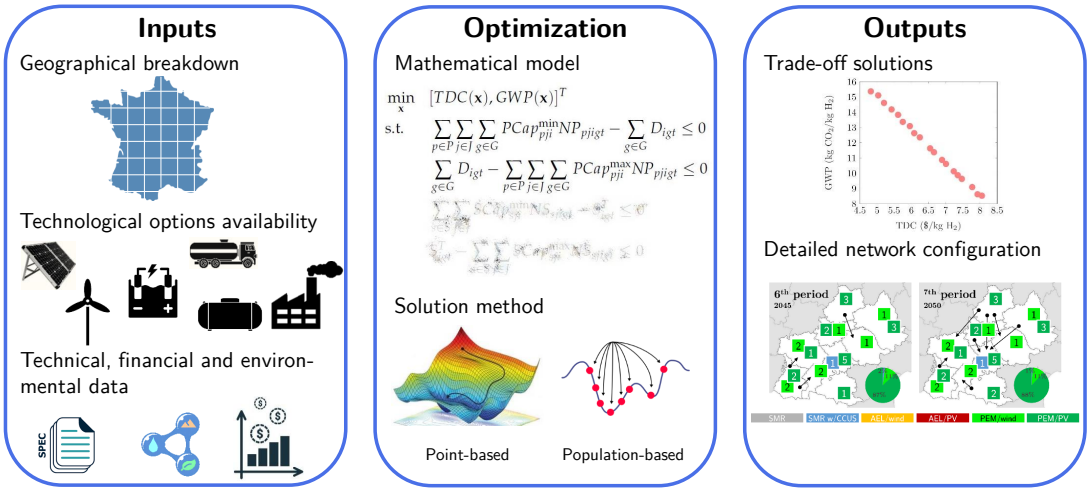


Figure 5.2: Methodological framework.

5.4 Mathematical model

In the mathematical model used as a basis in this work (Almaraz et al. (2015) and Almaraz et al. (2014b)), the total daily cost (TDC) of the supply chain is considered as the first objective and the global warming potential (GWP) as the second one. Obviously, both objectives are to be minimized and it is fair to anticipate that these

objectives are conflicting, since the cheapest configurations are generally also those with the highest environmental impact. The model includes appropriate equations accounting for investment costs related to plant installation and transportation routes, operational costs for production, storage and transportation, and also constraints for plant capacity, mass balance between grids and demand satisfaction.

5.4.1 Economic objective function

The TDC objective function, accounting for $|T|$ time periods, is calculated as the sum of the total daily cost over all periods as:

$$TDC = \sum_{t \in T} TDC_t \tag{5.1}$$

The total daily cost for each period t takes into account investment costs related to plant installation (FCC) and transportation (TCC), as well as operational costs related to production and storage (FOC), transportation (TOC) and use of energy source (ESC). It is computed as:

$$TDC_t = \frac{FCC_t + TCC_t}{\alpha CCF} + FOC_t + TOC_t + ESC_t \qquad \forall t \in T \tag{5.2}$$

where FCC_t represents the facility capital cost at each period t and thus considers only the cost associated to the installation of new production plants (IP_{pjigt}) and new storing facilities (IS_{sjigt}) at a given period as:

$$FCC_t = \frac{1}{LR_t} \sum_{j \in J} \sum_{i \in I} \sum_{g \in G} \left(\sum_{p \in P} PCC_{pji} IP_{pjigt} + \sum_{s \in S} SCC_{sji} IS_{sjigt} \right) \qquad \forall t \in T \tag{5.3}$$

where LR_t represents the learning rate, that is, a cost reduction associated to acquired experience over time by technology manufacturers. The decision variables corresponding to the number of production and storage facilities, NP_{pjigt} and NS_{sjigt}, are related to IP_{pjigt} and IS_{sjigt} by the following two equations, respectively, as:

$$NP_{pjigt} = \sum_{t=1}^{t} IP_{pjigt} \qquad \forall p \in P, \forall j \in J, \forall i \in I, \forall g \in G, \forall t \in T \tag{5.4}$$

$$NS_{sjigt} = \sum_{t=1}^{t} IS_{sjigt} \qquad \forall s \in S, \forall j \in J, \forall i \in I, \forall g \in G, \forall t \in T \tag{5.5}$$

The transportation capital cost (TCC_t) considers the flow rate of hydrogen among grids ($Q_{ilgg't}$), the transportation mode availability (TMA_l), the capacity of the transportation container ($TCap_{il}$), the average distance connecting two grids ($L_{lgg'}$), the average speed (SP_l) as well as the loading/unloading time (LUT_l). Also, a factor accounting for establishing a given transportation mode is used (TMC_{il}). It is calculated as:

$$TCC_t = TMC_{il} \sum_{i \in I} \sum_{l \in L} \sum_{g \in G, g' \neq g} \frac{Q_{ilgg't}}{TMA_l TCap_{il}} \left(\frac{2L_{lgg'}}{SP_l} + LUT_l \right) \qquad \forall t \in T \tag{5.6}$$

The facility operating cost (FOC_t) considers the efficient operation of each produc-

tion plant and storage facility. It is directly related to the amount of production and storage as:

$$FOC_t = \sum_{j\in J}\sum_{i\in I}\sum_{g\in G}\left(\sum_{p\in P} UPC_{pji}P_{pjigt} + \sum_{s\in S} USC_{sji}S^T_{igt}\right) \qquad \forall t \in T \tag{5.7}$$

where UPC_{pji} and USC_{sji} are the unit costs of production and storage, respectively.

The transportation operating cost (TOC_t) consists of fuel, maintenance, labor and general costs, as formulated in the following equation:

$$\begin{aligned} TOC_t = \sum_{i\in I}\sum_{l\in L}\sum_{g\in G, g'\neq g} \frac{2L_{lgg'}Q_{ilgg't}}{TCap_{il}}\left(\frac{FP_l}{FE_l} + ME_l \right. \\ \left. + \left(DW_l + \frac{GE_l}{TMA_l}\right)\left(\frac{1}{SP_l} + \frac{LUT_l}{2L_{lgg'}}\right)\right) \qquad \forall t \in T \end{aligned} \tag{5.8}$$

where the terms inside the parenthesis accounts for the fuel costs (FP_l, FE_l), maintenance expenses (ME_t), driver wages (DW_l) and general maintenance costs (GE_l, TMA_t), which depends on the working time (related to the speed SP_l, the distance $L_{lgg'}$ and the loading/unloading times LUT_l).

The cost associated to the transportation of primary energy sources is computed as:

$$ESC_t = \sum_{e\in E}\sum_{g\in G} UIC_e IPES_{egt} \qquad \forall t \in T \tag{5.9}$$

where UIC_e is the unit import cost of energy source and $IPES_{egt}$ is the corresponding amount of imported energy source, which depends on the production rate and the availability of energy sources (A_{egt}) for a given grid g, at each period t, according to:

$$\begin{aligned} IPES_{egt} = \max\left(0, SSF\sum_{p\in P}\sum_{j\in J}\sum_{i\in I}\gamma_{epj}P_{pjigt} - A_{egt}\right) \qquad \forall e \in E, \\ \forall g \in G, \forall t \in T \end{aligned} \tag{5.10}$$

where SSF is a safety stock factor for storing a small inventory of energy source, while γ_{epj} represents the utilization rate of primary energy sources.

5.4.2 Environmental objective function

Regarding the second objective, the overall effects of greenhouse gases (GHG) of the network (production, storage and transportation) are accounted according to the following equation:

$$GWP = \sum_{t\in T}(PGWP_t + SGWP_t + TGWP_t) \tag{5.11}$$

where $PGWP_t$ is computed as the production rate of each production plant in the network, times a global warming potential factor associated:

$$PGWP_t = \sum_{p\in P}\sum_{j\in J}\sum_{i\in I}\sum_{g\in G} GW_p^{prod} P_{pjigt} \qquad \forall t \in T \tag{5.12}$$

Similarly, the global warming potential due to hydrogen storage is expressed as the production rate of each production plant in the network times a global warming potential factor:

$$SGWP_t = \sum_{p\in P}\sum_{j\in J}\sum_{i\in I}\sum_{g\in G} GW_i^{stock} P_{pjigt} \qquad \forall t \in T \tag{5.13}$$

The third term in equation (5.11) refers to the global warming potential due to transportation, it considers the average distance and the hydrogen flow rate between grids, the capacity of the transportation mode employed, its weight (W_l) and a transportation global warming potential factor. It is computed as:

$$TGWP_t = \sum_{i\in I}\sum_{l\in L}\sum_{g\in G, g'\neq g}\left(\frac{2L_{lgg'}Q_{ilgg't}}{TCap_{il}}\right) GW_i^{transp} W_l \qquad \forall t \in T \tag{5.14}$$

5.4.3 Model constraints

Capacity constraints

The installed production plants in the network (NP_{pjigt}) must allow to exactly satisfy the total hydrogen demand, for each type of hydrogen physical form i and for each period t. This is enforced by constraints (5.15) and (5.16), as follows:

$$\sum_{p\in P}\sum_{j\in J}\sum_{g\in G} PCap_{pji}^{\min} NP_{pjigt} - \sum_{g\in G} D_{igt} \leq 0 \qquad \forall i \in I, \forall t \in T \tag{5.15}$$

$$\sum_{g\in G} D_{igt} - \sum_{p\in P}\sum_{j\in J}\sum_{g\in G} PCap_{pji}^{\max} NP_{pjigt} \leq 0 \qquad \forall i \in I, \forall t \in T \tag{5.16}$$

Similarly, constraints (5.17) and (5.18) ensure that the installed storage facilities (NS_{sjigt}) will guarantee the total inventory within certain limits, for each storing product form i, each grid g and each period t:

$$\sum_{s\in S}\sum_{j\in J} SCap_{sji}^{\min} NS_{sjigt} - S_{igt}^T \leq 0 \qquad \forall i \in I, \forall g \in G, \forall t \in T \tag{5.17}$$

$$S_{igt}^T - \sum_{s\in S}\sum_{j\in J} SCap_{sji}^{\max} NS_{sjigt} \leq 0 \qquad \forall i \in I, \forall g \in G, \forall t \in T \tag{5.18}$$

Constraints (5.19) and (5.20) force the production rate of a given plant (P_{pjigt}) to be within the allowed (minimum and maximum) production capacities:

$$PCap_{pji}^{\min} NP_{pjigt} - P_{pjigt} \leq 0 \qquad \forall p \in P, \forall j \in J, \forall i \in I, \forall g \in G, \forall t \in T \tag{5.19}$$

$$P_{pjigt} - PCap_{pji}^{\max} NP_{pjigt} \leq 0 \qquad \forall p \in P, \forall j \in J, \forall i \in I, \forall g \in G, \forall t \in T \tag{5.20}$$

Mass balance constraints

Hydrogen demand must be satisfied exactly (that is, overproduction is not allowed), through local production and/or importation/exportation from/to other grid ($Q_{ilgg't}$). This mass balance needs to be fulfilled for each product physical form i, each grid g

and each period t, as stated in the following equation:

$$\sum_{p\in P}\sum_{j\in J} P_{pjigt} - \sum_{l\in L}\sum_{g'\in G, g'\neq g} \left(Q_{ilgg't} - Q_{ilg'gt}\right) - D_{igt} = 0 \qquad \forall i \in I, \forall g \in G, \forall t \in T \tag{5.21}$$

Bounds on decision variables

Constraints (5.22) and (5.23) define the decision variables type for the production and storage facilities (integer) and for production and flow rates (real non-negative).

$$NP_{pjigt}, NS_{sjigt} \in \mathbb{N} \qquad \forall p \in P, \forall s \in S, \forall j \in J, \forall i \in I, \forall g \in G, \forall t \in T \tag{5.22}$$

$$P_{pjigt}, Q_{ilgg'} \in \mathbb{R}_{\geq 0} \qquad \forall p \in P, \forall j \in J, \forall i \in I, \forall l \in L, \forall g \in G, \forall t \in T \tag{5.23}$$

5.4.4 Overall formulation

Therefore, the bi-objective HSC optimization problem can be represented mathematically as:

$$\min_{\mathbf{x}} \quad [TDC(\mathbf{x}), GWP(\mathbf{x})]^T \tag{5.24}$$

$$\text{s.t.} \quad \text{constraints } 5.15 - 5.23 \tag{5.25}$$

where $\mathbf{x}$ in equation (5.24) represents the vector of decision variables, i.e., $\mathbf{x} = [NP_{pjigt}, NS_{sjigt}, P_{pjigt}, Q_{ilgg't}]^T$. The exhaustive model nomenclature is available in Appendix B.

5.5 Solution strategy

The purpose of this chapter is to explore an alternative approach for the solution of the HSC problem, represented through the previous model. To this end, the multiobjective evolutionary algorithm based on decomposition (MOEA/D) is used. The constraint-handling technique adopted is the ε-constraint method (see Section 3.3.5 in Chapter 3) with the repair of infeasible solutions using the constraints' gradient information. The decision variables are $NP_{pjigt}, NS_{sjigt}, P_{pjigt}, Q_{ilgg't}$. These are all encoded as continuous variables, and variables NP_{pjigt}, NS_{sjigt} are rounded to the next integer only in the evaluation module. Therefore, constraints (5.22-5.23) about the nature of the variables are automatically fulfilled by the MOEA. With respect to the capacity constraints (5.15-5.18), they are not difficult to satisfy by stochastic operators, since these inequality constraints bound a large feasible region. Regarding constraints (5.19-5.21), these are more difficult to fulfill because multiple decision variables are involved simultaneously (NP_{pjigt} and P_{pjigt} in constraints 5.19-5.19, and P_{pjigt} and $Q_{ilgg't}$ in constraint 5.21). In particular, the mass balance constraint is the most difficult constraint to satisfy, represented by only one equality but that might involve a considerable number of decision variables.

Considering the repair of infeasible individuals, this is carried out only on the continuous variables genotype of each individual, i.e., on variables P_{pjigt} and $Q_{ilgg't}$. It is worth noting that, even if variables repair could be performed also for integer variables (because they are encoded as continuous ones), this is not done because this

would entail higher computational times and no improvement in the algorithm was observed, according to a preliminary analysis.

5.6 Computational experiments

To investigate the performance level of the specific MOEA/D implementation described above, three instances of small-to-medium sizes are studied. The obtained results are compared with those produced by a mathematical programming technique, taken as a reference since it is the classical approach proposed in the literature.

5.6.1 Case study 1

For these computational experiments, three increasing size instances are considered that correspond to the data for the Great Britain case study (Almansoori and Shah, 2006). Note that this case study is taken only for numerical purposes as it constitutes a simplified or reduced instance. Only one time period is considered, the importation of energy feedstock is neglected as well as appropriate energy sources for the sustainable evaluation of the HSC. Instances 1 and 2 represent a part of the entire geographic area of Instance 3; they contain 6, 12 and 34 grids, respectively. The main characteristics of these instances are:

- Two different production technologies considered: steam methane reforming (SMR) and biomass gasification.
- One size for production and storage is available.
- Only one physical form of hydrogen (liquid) and one distribution type (tanker truck).
- Only one period of time (snapshot model).

5.6.2 Parameter settings

The parameter settings for the experiments carried out are summarized as follows. MOEA/D has a population size (μ) equal to 51 for all three instances. The weighted Tchebycheff scalarizing function (see page 28) is used within MOEA/D. The probability of choosing parents locally is $\delta = 0$, the neighborhood size is $T = 0.1\mu$ and the maximum number of replacements allowed (in the selection step of the algorithm) is $n_r = 2$. An external archive is employed to store the non-dominated solutions found in the optimization process. The external archive has no size limit, however, at the end of an execution, the external archive may be reduced to contain at most 51 individuals (i.e., the same number of scalar problems as for the exact approach, as explained hereafter) to compute the quality of the obtained approximation, using the hypervolume as the selection criterion. Regarding the variation operator, jDE with polynomial mutation is employed with parameters: $p_m = 1/n, \eta_m = 20$. The number of function evaluations, NFE, is equal to 1e6 for all instances, so that similar CPU times to those required by the exact approach are employed. The ε-constraint method with gradient-based repair is used as a constraint handling technique. The parameter settings are $\theta = 0.2\mu, cp = 5, T_c = 0.2T_{max}$, $P_g = 1, R_g = 3$, step size for finite differences: 10^{-6}.

For the evolutionary algorithm, 11 independent runs were performed for each instance (in order to be able to analyze some median run). With respect to the exact approach, it consists in the CPLEX solver with the weighted Tchebycheff utility function (just like MOEA/D). Accordingly, the bi-objective optimization problem is decomposed into a number of scalar subproblems, in this case the number of weight vectors (equivalent to the number of scalar subproblems) is set to 51, equal to the population size for MOEA/D. The stopping criteria for each CPLEX execution are the optimality gap (lower than 0.01%) or the resource time limit (100 seconds p/point). The obtained results are compared in terms of the hypervolume indicator. As mentioned previously, a higher value of this indicator is preferred as it is correlated to both convergence and diversity of the solution set. The reference point to compute the hypervolume is located at $[1.1, 1.1]^T$ in the normalized objective space $[0, 1]^2$, for all instances. The MOEA/D algorithm was coded in MATLAB R2019a and the exact approach was implemented in GAMS environment (v23.9.5) using the solver CPLEX v12.4.0.1. All the computational experiments presented here were carried out with a processor Intel Xeon E3-1505M v6 at 3.00 GHz and 32 Go RAM.

5.6.3 Results and discussion

The obtained results are summed up in Table 5.1. Please note that the column referring to the optimality gap is only for the exact approach. The computational times (in seconds) are presented in the table; for the exact approach ("exact" column) the time corresponds the solution of the 51 scalar subproblems, corresponding to the production of one Pareto set; for the evolutionary algorithm ("MOEA" column) the time corresponds to each execution (i.e., also producing a complete Pareto set). The table also presents the performance of both approaches in terms of the hypervolume indicator. Note that the data corresponding to MOEA/D represent the mean and the standard deviation (in parenthesis) values of the hypervolume. Clearly, the exact approach strongly outperforms MOEA/D in all three instances, mainly because MOEA/D experiences difficulties for exploring adequately the feasible search space.

Table 5.1: Numerical results of both approaches for case study 1.

Instance	CPU time(s) exact	CPU time(s) MOEA	Optimality gap (%)	HV(CPLEX Tcheb.)	HV(ε-MOEA/D w/grad)
HSC06g	49.1	120.6	0.01	0.9812	0.8161 (0.0353)
HSC12g	558.6	294.5	0.01	1.0018	0.7437 (0.0310)
HSC34g	1 223.3	1 637.4	0.01	0.8896	0.6102 (0.0024)

In Figure 5.3, the Pareto front approximations of both approaches are provided for each instance. Please note that, in the figure, both *TDC* and *GWP* objectives are presented per unit of H_2 produced, this is done by dividing the original objectives by the total hydrogen demand. This is done throughout this thesis. Also, in the figure, the Pareto front approximations related to MOEA/D correspond to the median run with respect to the hypervolume. It is worth mentioning that all the corresponding solutions shown in the figure are feasible. It can be observed that the evolutionary algorithm exhibits a poor convergence, i.e., most solutions are far away from the front approximated by CPLEX (which is likely to be a discretization of the true Pareto front). The reason for this trend is that the exploration capacity of the MOEA is conditioned

by the constraints satisfaction. In other words, the search cannot progress towards optimal regions because (1) new offspring generated by the variation operators are likely to be infeasible, i.e. they do not represent valid supply chains, and (2) the constraint-handling strategy does not efficiently make use of those infeasible offspring, even if some solutions are repaired. In this respect, it is important to recall that the feasible region is composed of disconnected subregions (due to the discrete nature of the problem) and the gradient-based repair has an effect only on continuous variables. Therefore, the population might get trapped in some feasible subregions, from which it cannot escape even if the constraints' gradient is used for repairing infeasible offspring. This phenomenon is observed in each of the three instances. Besides, with respect to the largest instance (HSC34g), MOEA/D not only shows convergence difficulties but, in addition, it is not capable of exploring the entire objective space range for both the TDC and GWP criteria.

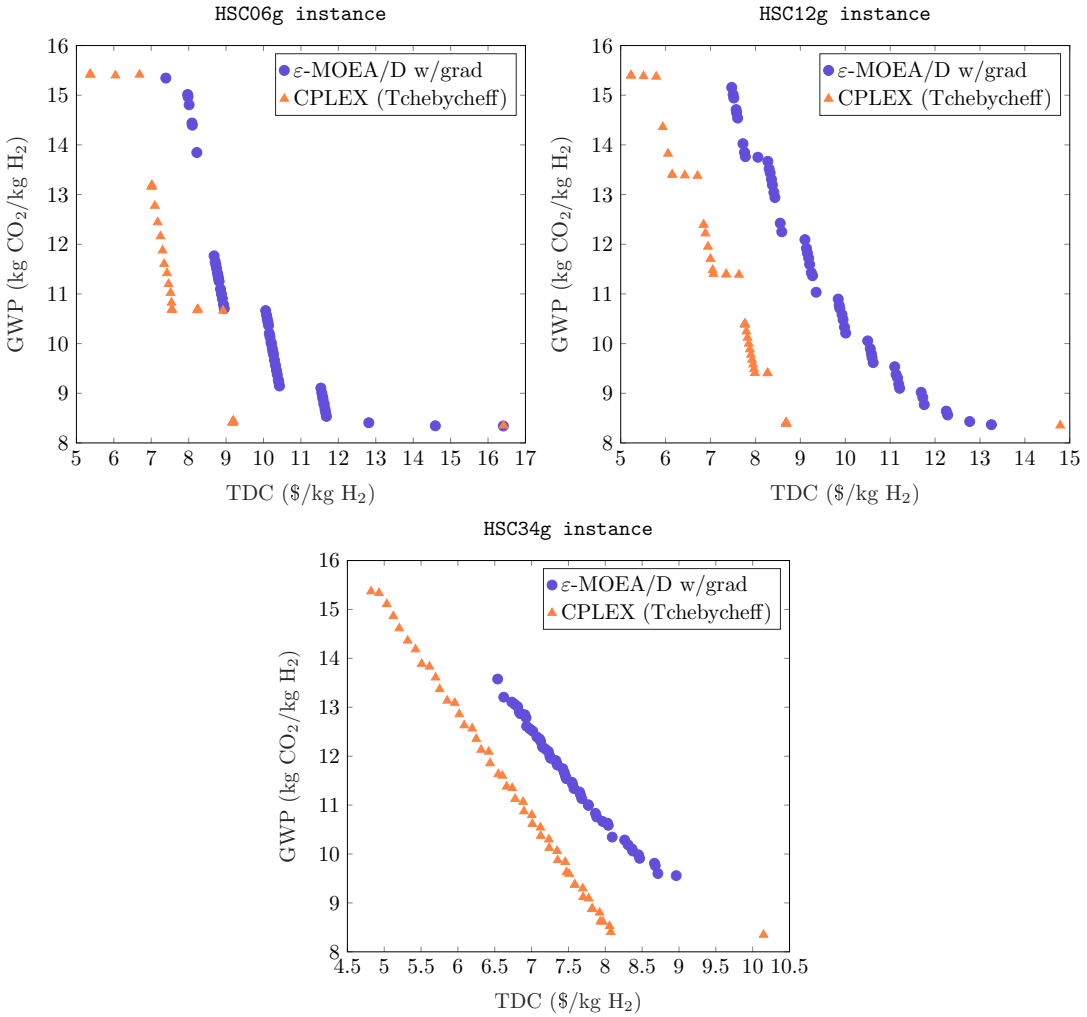


Figure 5.3: Final Pareto front approximations for case study 1.

5.7 Conclusions: strengths and weaknesses of a matheuristic approach

In this chapter, the first experiments carried out using a MOEA as a solution technique for the optimal design of hydrogen supply chains were presented. This optimization problem presents combinatorial aspects that can entail difficulties to classical optimization methods, but also features related to constraint satisfaction that make it difficult to treat by canonical MOEAs.

In particular, the performance of ε-MOEA/D, using the gradient-based repair as the constraint-handling technique, was investigated. According to the obtained results, this strategy experiences difficulties for determining the optimal configurations of the hydrogen supply chain. Thus, due to the fact that only continuous variables are repaired, solutions might get trapped in some feasible subregions and thus the search cannot progress properly towards the true Pareto front.

Nevertheless, these first experiments allowed to identify the main difficulties experienced by both exact and metaheuristic approaches, namely, the combinatorial and multiobjective aspects for the exact approach (evidenced mainly in multi-period instances) and the constraint-handling for the evolutionary algorithm. Therefore, accounting for the bilevel formulations proposed in previous studies on similar problems (see Section 5.2), a novel strategy for the solution of the HSC problem is introduced in the next chapter. This strategy decomposes the problem in such a way that some advantages can be taken of the strengths of every single approach, mitigating simultaneously their weaknesses.

Chapter 6

A novel strategy based on bilevel decomposition for HSCs design

Contents

6.1 Introduction

In the previous chapter, a metaheuristic-based solution technique was explored for the solution of the problem of sustainable design hydrogen supply chains, considering both economic and environmental criteria. However, the obtained results were rather disappointing, in particular a lack of convergence to the true Pareto front was observed, mainly due to the high number of constraints and continuous variables involved. Thus, this chapter presents a novel methodology based on a reformulation of the original problem into a bilevel optimization problem. More precisely, the upper-level problem tackles the installation problem (location and sizing the production plants and storage facilities), whereas the lower-level problem addresses the operation problem associated to the production and transportation aspects. This reformulation allows to employ a multiobjective evolutionary algorithm for the solution of the upper-level problem, while the lower level is solved using a linear programming solver. The proposed methodology is validated and compared with the Pareto front approximations obtained by CPLEX over nine increasing size instances.

The content of this chapter has been accepted for publication in the form of journal article. Victor H Cantú, Catherine Azzaro-Pantel, and Antonin Ponsich (accepted). "A novel matheuristic based on bi-level optimization for the multi-objective design of hydrogen supply chains". In: *Computers & Chemical Engineering*, p. 107370. DOI: https://doi.org/10.1016/j.compchemeng.2021.107370

The chapter is organized in the following way. Section 6.2 presents the motivation for this new solution strategy, as well as its core innovations and contributions. In the subsequent section, the proposed methodology for solving the HSC problem is presented with the bilevel reformulation, the main aspects of the hybrid solution strategy and some specific features of the tools proposed at each level. The empirical validation of the proposed methodology is provided in Section 6.4, where the results obtained for nine growing-size instances corresponding to the case of Great Britain and the Midi-Pyrénées region (France) are compared with those obtained using a classical approach. Finally, conclusions are drawn in Section 6.5.

6.2 Motivation for a hybrid strategy

As stated in the previous chapter, most of the related works proposed in the literature employ exact techniques for tackling the HSC problem. These classical techniques turn out to be computationally prohibitive for evaluating large-size instances of the problem. Besides, it is noteworthy that, even though strategies based on bilevel decomposition have been proposed (see Guillén-Gosálbez et al., 2010; Sabio et al., 2010), the multiobjective feature of the problem is invariably handled by the ε-constraint method, which might be inappropriate when considering more than two objectives or for attaining an accurate approximation of the Pareto front. In addition, similar decomposition approaches have been developed in other areas of application (see Fazlollahi and Maréchal, 2013), but they differ in the solution method: they employ metaheuristic techniques for approximating the Pareto front in the master problem, and exact methods for solving the MILP lower-level problem. Based on these guidelines, a hybrid methodology is proposed here as a solution tool for the HSC design problem.

The core innovations and contributions of this proposal are described in more details as follows:

1. A novel methodology is designed for the solution of the HSC design problem, in which the mathematical structure of the problem can be exploited. In this manner, the multiobjective, combinatorial and linear aspects of the problem can be tackled by appropriate solution methods.

2. Consequently, the original problem is reformulated as a bilevel multiobjective optimization problem, through a master-slave decomposition strategy. The upper-level (master) problem considers the installation of production and storage facilities (multiobjective combinatorial problem), while the lower level examines the problem associated to transportation and production rates (linear programming problem of low complexity).

3. A hybrid solution tool based on both evolutionary algorithms and linear programming (matheuristic) is developed for the solution of the resulting bilevel optimization problem, able to efficiently achieve a good approximation of the Pareto front in terms of convergence, distribution and number of Pareto solutions. More precisely, the solution of the master problem is performed by a multiobjective evolutionary algorithm (MOEA), taking full advantage of the ability of such techniques for dealing with the multiobjective and combinatorial features of the problem. Then, for each individual (partial solution)

proposed by the evolutionary algorithm, the slave problem is treated by a linear programming (LP) solver. Please note that the lower-level problem is also a multiobjective one, so that a scalarizing function is embedded within the LP formulation. In this way, the multiobjective bilevel problem is solved in an iteratively manner, in one single run.

4. Finally, the performance of the proposed methodology is evaluated over nine increasing size instances of the HSC design problem, inspired from Almaraz et al., 2014a. To this end, the performance of both deterministic and hybrid approaches is presented in terms of the hypervolume indicator, for the first time in the HSC literature. The results, analyzed through this indicator, show that the proposed hybrid approach outperforms the classical one (exact algorithm), i.e., the set of non-dominated solutions obtained by the hybrid approach are better distributed along the Pareto front, offering the decision makers a better picture of all the possible trade-off solutions. Moreover, the proposed methodology can provide approximation sets with a controllable number of efficient solutions, in reasonable computational times.

6.3 An efficient solution strategy

The complexity of the HSC problem, due to its combinatorial and multiobjective nature, deserves the development of an adapted solution strategy. To this end, the original problem (A)[1] is reformulated into a bilevel optimization problem. The idea behind this reformulation is to decompose the original problem into two subproblems of lower complexity, in such a way that each aspect of the original problem can be tackled by appropriate solution approaches. The two main parts of problem (A) are (1) the combinatorial problem related to facility installation and (2) the continuous problem associated to production and transportation operation. Then, in such a decomposition, the combinatorial problem can be tackled apart by a population-based metaheuristic, so that good-quality solutions can be obtained even for large-size instances in controllable computational times. Additionally, the multiobjective nature of this subproblem can be accounted for by the metaheuristic, in particular, using a MOEA. Now, regarding the continuous problem, it can be easily transformed into a canonical transportation problem applying a simple heuristic (detailed in Subsection 6.3.4), so that it can be solved efficiently by exact methods.

In what follows, the reformulation of problem (A) into a bilevel problem is formally provided. Then, the solution methodology is discussed with a specific focus on each level.

6.3.1 Bilevel formulation

In order to propose an efficient strategy to solve the optimization problem (A), a decomposition of the problem is proposed in this subsection. The resulting bilevel optimization problem is represented as follows:

$$\min_{\mathbf{x}} \quad [TDC(\mathbf{x}), GWP(\mathbf{x})]^T \tag{6.1}$$

[1]For the sake of readability, in this chapter the problem represented by equations (5.1-5.25) is named problem (A), it constitutes the original or classical formulation of the HSC problem, according to the models of Almansoori and Shah, 2009; Almaraz et al., 2015; Almaraz et al., 2014a.

$$\text{s.t.} \quad \sum_{p\in P}\sum_{j\in J}\sum_{g\in G} PCap^{\min}_{pji} NP_{pjigt} - \sum_{g\in G} D_{igt} \leq 0 \qquad \forall i \in I, \forall t \in T \tag{6.2}$$

$$\sum_{g\in G} D_{igt} - \sum_{p\in P}\sum_{j\in J}\sum_{g\in G} PCap^{\max}_{pji} NP_{pjigt} \leq 0 \qquad \forall i \in I, \forall t \in T \tag{6.3}$$

$$\sum_{s\in S}\sum_{j\in J} SCap^{\min}_{sji} NS_{sjigt} - S^{T}_{igt} \leq 0 \qquad \forall i \in I, \forall g \in G, \forall t \in T \tag{6.4}$$

$$S^{T}_{igt} - \sum_{s\in S}\sum_{j\in J} SCap^{\max}_{sji} NS_{sjigt} \leq 0 \qquad \forall i \in I, \forall g \in G, \forall t \in T \tag{6.5}$$

$$NP_{pjigt}, NS_{sjigt} \in \mathbb{N} \qquad \forall p \in P, \forall s \in S, \forall j \in J, \forall i \in I, \forall g \in G, \forall t \in T \tag{6.6}$$

$$\min_{\mathbf{y}} \quad [TDC(\mathbf{y}), GWP(\mathbf{y})]^{T} \tag{6.7}$$

$$\text{s.t.} \quad PCap^{\min}_{pji} NP_{pjigt} - P_{pjigt} \leq 0 \qquad \forall p \in P, \forall j \in J, \forall i \in I, \forall g \in G, \forall t \in T \tag{6.8}$$

$$P_{pjigt} - PCap^{\max}_{pji} NP_{pjigt} \leq 0 \qquad \forall p \in P, \forall j \in J, \forall i \in I, \forall g \in G, \forall t \in T \tag{6.9}$$

$$\sum_{p\in P}\sum_{j\in J} P_{pjigt} - \sum_{l\in L}\sum_{g'\in G, g'\neq g} \left(Q_{ilgg't} - Q_{ilg'gt}\right) - D_{igt} = 0 \qquad \forall i \in I, \forall g \in G, \forall t \in T \tag{6.10}$$

$$P_{pjigt}, Q_{ilgg'} \in \mathbb{R}_{\geq 0} \qquad \forall p \in P, \forall j \in J, \forall i \in I, \forall l \in L, \forall g \in G, \forall t \in T \tag{6.11}$$

where $\mathbf{x}$ in equation (6.1) represents the vector of decision variables in the upper level, i.e., $\mathbf{x} = [NP_{pjigt}, NS_{sjigt}]^T$, and $\mathbf{y}$ is the vector of decision variables in the lower level, i.e., $\mathbf{y} = [P_{pjigt}, Q_{ilgg'}]^T$. Equations (6.1-6.6) refer to the upper-level problem, whereas equations (6.7-6.11) describe the lower-level problem. In addition to equations (6.1-6.11), the complete formulation of the bilevel problem also includes equations (5.1-5.14), which refer to the computation of TDC and GWP objective functions. This constitutes problem (B).

The upper-level (master) problem addresses the combinatorial problem associated to the structure of the supply chain, that is, it addresses the optimal location, sizing and selection of the technology type of production and storage facilities. Concerning the lower-level (slave) problem, it consists in finding the optimal production rates and transportation flows between grids for the network configuration predicted by the upper level.

6.3.2 Global description of the strategy

As stated previously, the proposed solution strategy to the optimization problem (B) consists in a hybrid approach, more precisely, a MOEA coupled with a linear programming solver. In this way, the upper-level integer variables NP_{pjigt} and NS_{sjigt} are handled by the evolutionary algorithm. Since these variables are generated through stochastic operators, they might require a repair mechanism in order to fulfill constraints (6.2-6.5), which state that production and storage facilities must satisfy the total demand in the network. Then, for each feasible (partial) solution, that is, for each feasible structure of the HSC provided by the evolutionary algorithm, the corresponding optimal continuous variables P_{pjigt} and $Q_{ilgg't}$ are computed by solving the multi-period operation problem, at the lower-level (see Figure 6.1).

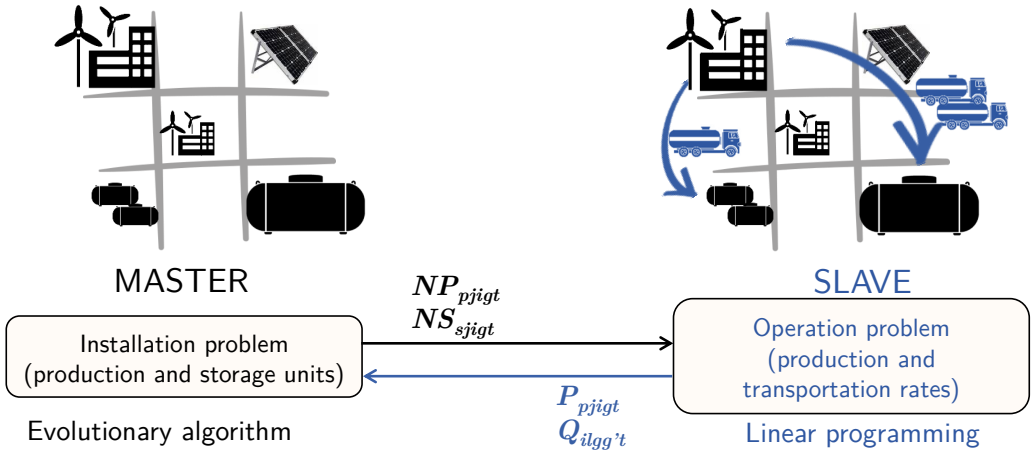


Figure 6.1: Simplified diagram of the hybrid (master-slave) approach. In black color the upper-level variables, whereas in blue color, the lower-level variables.

At this point, it is important to note that, even if the lower-level subproblem constitutes a bi-objective problem, only one solution is required to evaluate a given upper-level candidate solution. To this end, a utility function with a random weight vector is used (this is discussed in more details in Subsection 6.3.4). Moreover, in order to speed up the solution of the slave problem, only grids for which the installed production units can satisfy local demand are considered as potential exporting grids (sources), otherwise they are considered as demand grids (sinks). This heuristic rule reduces the complexity of the linear programming problem, by decreasing the number of potential edges among grids.

Once the slave problem has been solved for every master problem's candidate solution, the MOEA recovers the continuous variables (P_{pjigt}, $Q_{ilgg't}$) from the LP solver, in order to compute the upper-level objective functions. Then, each individual in the population has a fitness function value assigned by the MOEA, which evolves the population to the next generation according to its working mode, e.g., according to the selection paradigm used (Pareto dominance, decomposition or indicator-based). Finally, the process is repeated iteratively until a stopping criterion is reached. The proposed methodology is presented in the pseudocode of Algorithm 6. Please note that all lines correspond to the solution of the master subproblem, excepting lines 7 and 8, which refer to the slave subproblem treatment.

6.3.3 Upper-level problem solution approach

In this chapter, SMS-EMOA and MOEA/D (Emmerich et al., 2005; Zhang et al., 2009), described in Chapter 1, are considered for the solution of the bi-objective master problem. In what follows, some relevant aspects of the implementation of both MOEAs in our framework are discussed.

Line 1: Initialization The initialization step consists in defining the initial parameters for the given algorithm (see their pseudocode in pages 35, 37). The initial population is created in such a way that production and storage facilities are added randomly into the grids in the network, one by one, until a feasible solution is obtained (which ensures the respect of constraints 6.2-6.5).

Algorithm 6 Hybrid strategy procedure

```
1: initialize MOEA
2: while not terminate do
3:     generate offspring through variation operators
4:     for all individuals in population do
5:         for all t ∈ T do {for each period}
6:             if offspring solution violates equations (6.2-6.5) then
7:                 repair infeasible solution
8:             end if
9:             build transportation problem (identify sink and source grids)
10:            solve transportation problem (LP solver)
11:        end for
12:        compute master problem's objective functions
13:        assign fitness value according to MOEA's working paradigm
14:        evolve population according to MOEA's working paradigm
15:    end for
16: end while
17: return current Pareto set approximation
```

Line 3: Generation of offspring Offspring solutions are generated by means of stochastic operators, namely, SBX for SMS-EMOA and DE for MOEA/D, both with polynomial mutation. The integer decision variables NP_{pjigt}, NS_{sjigt} are encoded as continuous variables and are rounded to the next integer only in the evaluation steps.

Lines 6-8: Repair If the offspring violates constraints (6.2-6.5, i.e., the installed capacity is either greater or lower than the total network demand), it is repaired by randomly adding or removing production and storage units accordingly. Note that (6.2-6.5) are inequality constraints which define large feasible regions, and thus are not difficult to fulfill following this procedure.

Line 12: Evaluation The optimal variables $P_{pjigt}, Q_{ilgg't}$ are recovered from the LP solver in order to compute the master problem's objective functions. Note that these variables are not encoded as decision variables in any individual, at any step of the evolutionary algorithm.

Line 13: Fitness assignment A fitness is assigned to each offspring solution according to each MOEA (utility function for MOEA/D or contribution to the hypervolume for SMS-EMOA).

Line 14: Selection/Evolution The best μ individuals are chosen to survive to the next generation, according to the MOEA's working paradigm.

Lines 16-17: Evolution/Termination If the termination criterion is not met (number of function evaluations), offspring solutions are generated and the process is repeated, otherwise, the MOEA gives as an output the current population (or the external archive for MOEA/D).

6.3.4 Lower-level problem solution approach

A first key point for the solution of the slave problem is the construction of a feasible transportation problem using the integer variables provided by the evolutionary module. To this end, an heuristic method is designed in this work. All grids in the network are considered as potential importing grids, that is, as potential "sinks" or "demand" nodes. Then, for each period t, a set of potential exporting grids G_t^E is determined, depending on the number and size of production plants installed, as follows: if, for a given grid g, the sum of the maximum production capacities of all plants installed is greater than its hydrogen demand, the grid g is considered as a potential "source" node and then $g \in G_t^E$. This can be mathematically formulated as:

$$D_{igt} \leq \sum_{p \in P} \sum_{j \in J} PCap_{pji}^{\max} NP_{pjigt} \implies g \in G_t^E \qquad \forall i \in I, \forall g \in G, \forall t \in T \tag{6.12}$$

In this way, the complexity of the resulting LP problem is considerably reduced, since only transportation links can occur from sources to sinks nodes. As it can be noted, this heuristic relies on the fact that grids that cannot produce enough hydrogen to satisfy their own demands (even if the installed facilities operate at their maximum capacity) are unlikely to export hydrogen to other grids, since this would require more transportation units, resulting in a more expensive solution.

Now, regarding the multiobjective aspect of the LP subproblems, this is tackled employing a utility function. In this work, the weighted Tchebycheff and the augmented achievement scalarizing function (AASF) are chosen, the former when MOEA/D is employed (in the upper level), whereas the latter is used when SMS-EMOA is applied to the solution of the upper-level problem. These two scalarizing functions are chosen, due to their ability to deal with non-convex front shapes and weakly dominated solutions (see equations 1.10 and 1.13 in Chapter 1). Moreover, since only one solution (not a set solutions) is needed to evaluate the upper-level problem, it is solved only once with a weight vector randomly generated such that $\mathbf{w} \in \mathbb{R}_+^k$ and $\sum_{j=1}^k w_j = 1$, with $k = 2$ for this problem. It is worth mentioning that the random generation of weight vectors might seem inappropriate to deal consistently with the lower-level subproblem. However, the experimental results validate this strategy for the studied problem. Anyway, a smart weight vector tuning mechanism is under the scope of future work.

6.4 Computational experiments

The efficiency of the proposed methodology for solving the bi-objective HSC problem is validated through a performance comparison with an exact technique (CPLEX with the weighted Tchebycheff scalarizing function). First, the case study 1 (Great Britain) from the previous chapter is addressed in the next subsection. Then, in Subsection 6.4.2 a more complex application related to the former Midi-Pyrénées region (now Occitanie region, France) is analyzed.

6.4.1 Case study 1

The performance of the proposed approach was first explored for the case study presented in the previous chapter, corresponding to the HSC deployment in Great

Britain. For further details concerning this application, the reader is referred to Section 5.6.1 of the previous chapter.

Parameter settings The experiments carried out used the same settings as in the previous chapter (see page 117), in order to highlight the benefits of using the matheuristic proposed in this chapter with respect to the canonical MOEA/D. Therefore, the evolutionary algorithm used as upper-level search engine is also MOEA/D with the weighted Tchebycheff scalarizing function. The population size (μ) is equal to 51 for all three instances. The external archive stores non-dominated solutions found in the optimization process and has no size limit. Nevertheless, for computing the quality of the obtained approximation, the external archive may be reduced at the end of an execution to contain at most 51 individuals, using the (least) contribution to the global hypervolume as a pruning criterion. Of course, this time MOEA/D does not use any constraint-handling technique, since the upper level, tackled by the MOEA, uses the simple above-mentioned repair mechanism to fulfill constraints (6.2-6.5). The lower-level subproblem is addressed through a scalarizing function (weighted Tchebycheff) with random weight vectors. The scalar problem is then solved using CPLEX solver v12.8.0, called from MATLAB.

As previously, 11 independent runs were performed for the evolutionary algorithm for each instance. With respect to the exact approach, its working mode and parameter settings are identical with those of the previous chapter. The obtained results are again compared in terms of the hypervolume indicator (with a reference point located at $[1.1, 1.1]^T$ in the normalized objective space). The hybrid algorithm was implemented in MATLAB R2019a, while the exact approach was solved within the GAMS environment (v23.9.5) with CPLEX v12.4.0.1. The computational experiments were carried out with a processor Intel Xeon E3-1505M v6 at 3.00 GHz and 32 Go RAM.

Results and discussion The numerical results obtained are presented in Table 6.1. In what respect the computational times, please note that the differences observed between the exact and hybrid techniques are the result of different stopping criteria for each approach, that is, the optimality gap (0.01%) for the exact approach and the number of function evaluations (1×10^6) for the hybrid approach. However, the CPU times used by both techniques have similar magnitude orders and the differences observed do not necessarily involve significant impact in algorithmic performance, as will be showed in the case study 2. Besides, in the column corresponding to the hybrid approach, the results represent the mean hypervolume values out of the 11 runs and in parenthesis the standard deviation. For the first two instances, the hypervolume indicator suggests that the hybrid approach slightly outperforms the classical one. This is displayed in Figure 6.2, which presents the final Pareto front approximations of both classical and hybrid approaches (that of the median run in terms of the found hypervolume for the matheuristic).

The first observation is that the hybrid technique produces more non-dominated solutions than the exact one, indeed the approximated sets provided by CPLEX contain less than 51 solutions, because the solver sometimes outputs the same solution for two scalar problems with similar weight vectors. As a consequence, if some parts of the front (the upper part for the two first instances) are identically described by both approaches, the matheuristic proposes a much finer discretization of some other

Table 6.1: Numerical results of both approaches for case study 1.

Instance	CPU time(s) exact	CPU time(s) hybrid	Optimality gap (%)	HV(CPLEX Tcheb.)	HV(hybrid)
HSC06g	49.1	164.6	0.01	0.9812	0.9844 (0.0023)
HSC12g	558.6	226.1	0.01	1.0018	1.0048 (0.0113)
HSC34g	1 223.3	1 909.0	0.01	0.8896	0.8848 (0.0032)

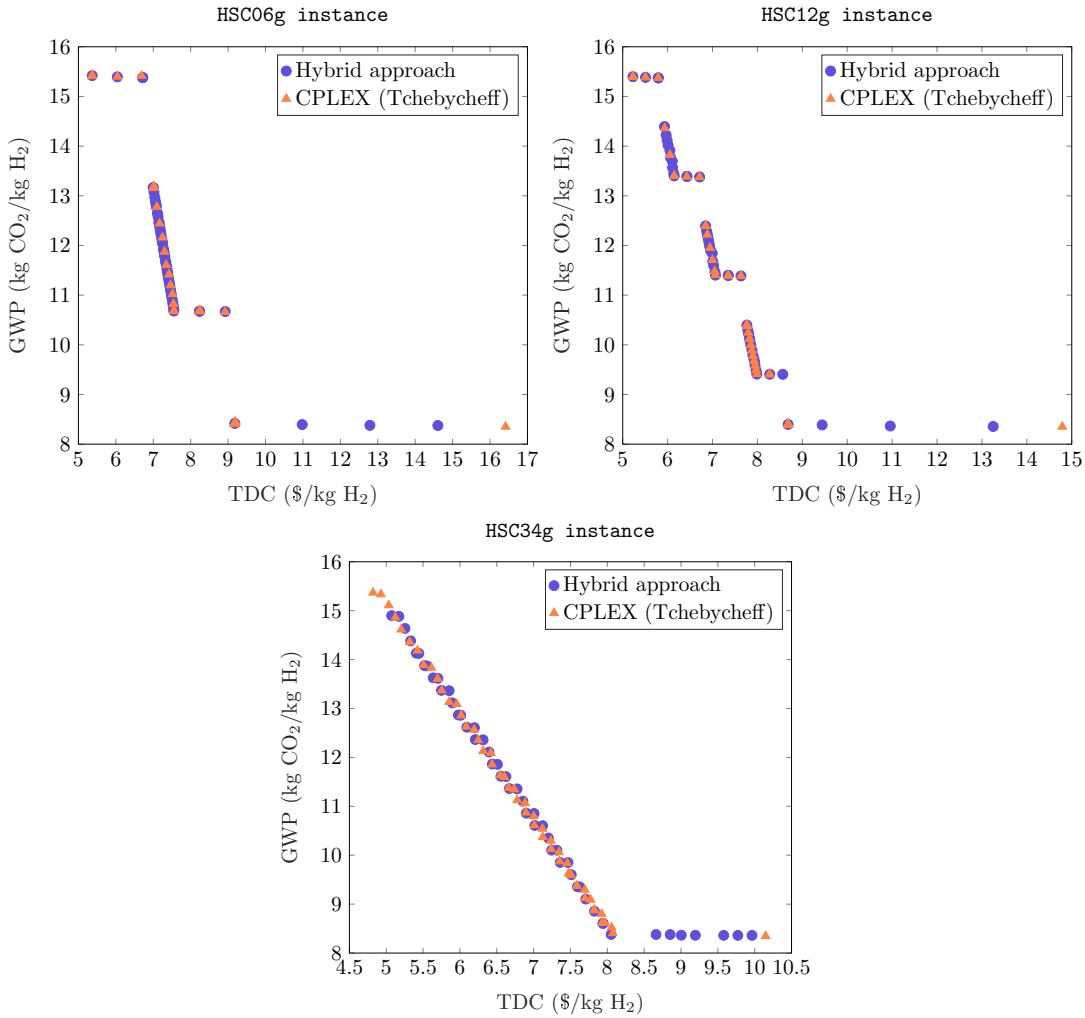


Figure 6.2: Pareto front approximations for case study 1.

segments of the front. As an example, consider the HSC12g instance: in the segment of the front corresponding to solutions entailing a global warming potential ranging from 13 to 15 kg CO_2/kg H_2, the classical approach found 5 efficient solutions, whereas the hybrid approach in the same segment provided 12 non-dominated solutions. For the HSC34g instance, the hypervolume indicates a slight superiority of the classical approach. This is corroborated in Figure 6.2, where the hybrid algorithm shows some difficulties to cover the whole range of the Pareto front, in particular the region corresponding to low TDC values.

Therefore, two conclusions can be drawn from this first case study. First, the hybrid approach proposed in this chapter strongly outperforms the solution strategy based on metaheuristics only (it is clear that the approximation sets and their corresponding hypervolume are much better now than when using MOEA/D in the previous chapter). On the other hand, there is no significant differences between the performance level of the matheuristic and the exact solution technique. This appears as a validation of the good behavior showed by the hybrid algorithm on this first case study involving a single period. So, in the following, a more complex case study, which considers multi-period instances and, thus, a greater combinatorial effect, is tackled next.

6.4.2 Case study 2

The instances for the HSC problem considered next correspond to the data for the case study of the Midi-Pyrénées region in France (Almaraz et al., 2014a), see Figure 6.3. Accordingly, the characteristics of the HSC considered are:

- Five primary energy sources are available, namely, solar photovoltaic, wind, hydro, French electrical mix (electricity from the grid) and natural gas.
- Three different production technologies: steam methane reforming (SMR), central and distributed electrolysis. Renewable energies can only serve as feedstock to electrolysis plants, while natural gas is exclusively destined to SMR technology.
- Three different production scales: small, medium and large. The large-scale size plant is only available for central electrolysis and SMR technologies.
- Once hydrogen is produced, it will be delivered to storage facilities via tanker trucks in liquid form.
- As for production plants, a four-size discrete range for storage facilities is available: mini, small, medium and large.
- The learning rate cost reductions due the accumulated experience is taken equal to 2% per period.

The detailed input data is presented in Appendix B. To validate the solution strategy, different size instances of the problem are generated, varying the number of grids and time periods. The network is divided into 8 or 22 grids, corresponding to administrative territories and departments, respectively. The planning horizon is set to year 2050, with three different periods divisions: 1 (2050), 4 (2020, 2030, 2040, 2050) and 7 periods (2020, 2025, 2030, 2035, 2040, 2045, 2050). The combination of

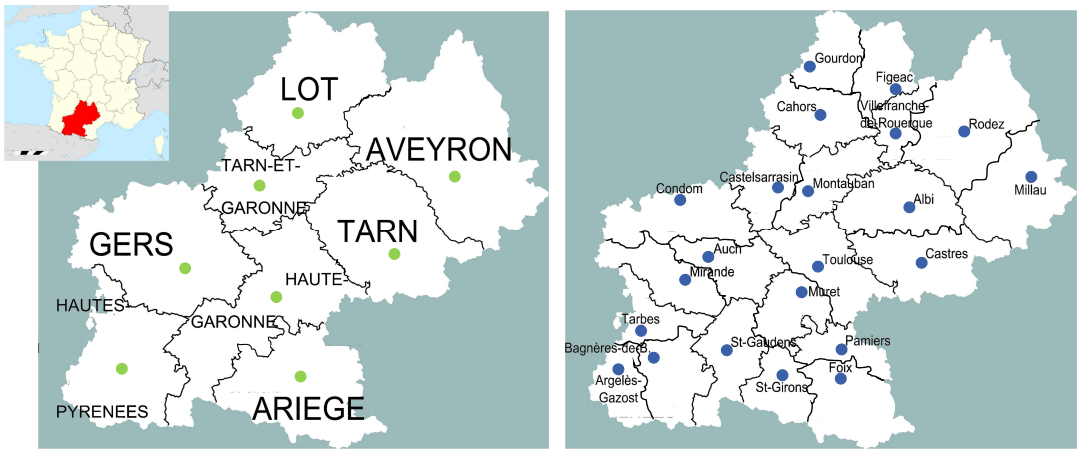


Figure 6.3: Midi-Pyrénées region case study. At the left, departments' names for 8-grid instances. At the right, territories' names for 22-grid instances.

these two aspects yields the six different instances studied here. The nomenclature used for instance names indicates the number of grids and periods, as an example, the instance *HSC22g04p* considers 22 demand grids and 4 time periods.

Parameter settings The performance of the proposed solution approach is compared to that of the exact technique usually considered in the literature (CPLEX with ε-constraint). To this end, the original MILP model, defined by equations (5.1-5.25), was implemented and solved within the GAMS environment. As before, the performance comparisons of solution sets of both approaches are carried out using the hypervolume indicator. Thus, solution sets of equal size are necessary to perform a fair comparison. For the exact approach, an approximation set containing 21 solutions is computed for each instance. We considered this arbitrary size (21) for the approximation set as adequate to display the shape of the true Pareto front and, at the same time, in order to avoid an intractable computational burden to the exact method. The stopping criteria for each CPLEX execution (21 executions for each instance) are (1) the solution found so far has an optimality gap lower than 0.01%, or (2) the computational time exceeds a given limit: the time limit per point is set to 1 000 seconds, but also, in order to track the any-time performance of each solution approach, setting the time limit per point at 100 seconds is also investigated.

Regarding the hybrid algorithm, in this case the SMS-EMOA is used, as it is particularly efficient to treat two-objective problems. The population size of the MOEA is set to 200 for 8-grid instances and 800 for 22-grid instances, according to a previous sensitivity analysis. The stopping criterion (number of function evaluations) is set so that similar computational times to those of the exact method are employed. The variation operators use the following (standard) parameters: SBX operator's probability and distribution $p_c = 1$ and $\eta_c = 20$, respectively; polynomial mutation's probability and distribution $p_m = 1/n$ and $\eta_m = 20$, respectively, where n is the number of decision variables. Also, since the Pareto set approximation provided by the MOEA typically contains more than 21 solutions, the final population is pruned (by removing sequentially the solutions showing the least contribution to the total hypervolume of the 200 solutions) so that only 21 well-distributed solutions are considered to calculate the HV. Finally, in order to perform appropriate comparison

Table 6.2: Numerical results of both approaches for case study 2.

Instance	Time limit p/point (s)	CPU time(s) exact	CPU time(s) hybrid	Optimality gap (%)	HV(ε-constr.)	HV(hybrid)
HSC08g01p	100	142.5	165.7	0.01	0.9552	0.9834 (0.0003)
HSC08g04p	100	1 711.7	1 872.1	0.06	0.8008	0.8099 (0.0002)
HSC08g04p	1 000	7 852.3	7 617.6	0.02	0.8008	0.8100 (0.0002)
HSC08g07p	100	2 056.8	1 945.1	0.23	0.7639	0.7680 (0.0004)
HSC08g07p	1 000	14 629.5	12 585.1	0.03	0.7639	0.7690 (0.0008)
HSC22g01p	100	1 674.7	1 809.2	0.41	0.9616	0.9794 (0.0022)
HSC22g01p	1 000	12 526.4	14 935.0	0.09	0.9618	0.9822 (0.0013)
HSC22g04p	100	2 432.3	2 156.1	2.31	0.7736	0.7776 (0.0035)
HSC22g04p	1 000	22 416.8	22 814.5	0.71	0.7908	0.7934 (0.0023)
HSC22g07p	100	2 425.4	2 366.3	–	Infeasible	0.7394 (0.0027)
HSC22g07p	1 000	22 950.4	24 088.0	0.95	0.7555	0.7563 (0.0014)

of results, the hybrid approach is run 11 times for each instance. In all figures, the obtained Pareto front approximations with the hybrid algorithm corresponds to the median run with respect to the hypervolume value.

The exact method was implemented in GAMS environment (v23.9.5) using CPLEX solver v12.4.0.1. For the hybrid approach, SMS-EMOA was implemented in MATLAB language (vR2019a) and the solution of the LP subproblems is performed using CPLEX solver v12.8.0, called from MATLAB. The computational experiments with both (deterministic and hybrid) approaches were carried out with the same computer hardware, i.e. a processor Intel Xeon E3-1505M v6 at 3.00 GHz and 32 Go RAM.

Results and discussion The obtained numerical results are displayed in Table 6.2. The column indicating the optimality gap refers to the average optimality gap over the 21 CPLEX executions required to produce an approximation set. Besides, Table 6.2 presents also intermediate results setting the time limit per point at 100 seconds. For the hybrid approach (last column), the results represent the mean hypervolume value and, in parentheses, the corresponding standard deviation over 11 runs. As mentioned previously, only 21 solutions from the last population are considered for the hypervolume computation.

First, small size instances were studied for a further validation of the hybrid solution method. The Midi-Pyrénées region is thus divided into its 8 main departments. From Table 6.2, it can be observed that the first instance, which considers a constant demand (i.e., one period), required less than 3 minutes for both approaches to obtain the Pareto front approximation. The final Pareto front approximations are displayed in Figure 6.4. It can be observed that the hybrid approach reproduces the non-dominated solutions found by CPLEX (red squares), and, additionally, it provides a much more detailed Pareto front approximation (blue points), with 200 non-dominated solutions.

From these 200 points, 21 are selected for the hypervolume computation in order to carry out a fair comparison with the output of the exact method (see Figure 6.4 right). However, it should be clear to the reader that, without pruning the final population for the above-mentioned purpose (of fair comparison between the hypervolumes of

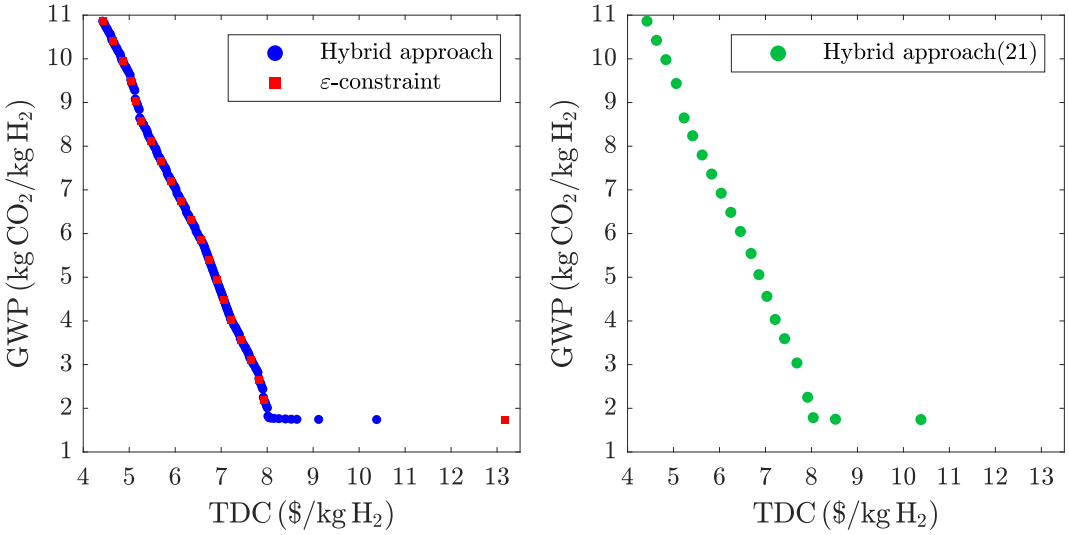


Figure 6.4: Final Pareto front approximations for HSC08g01p instance.

both approaches), the hypervolume corresponding to the complete approximation produced by the hybrid algorithm would drastically outperform those shown in the table for 21 points. From the numerical results in the table, it can be concluded that the 21-point approximation set proposed by the hybrid approach (HV 0.9834) is a better representation of the true Pareto front than that of the exact method (HV 0.9552), because solutions are better distributed along the Pareto front. Also, since the standard deviation is not significant (0.0003), it can be inferred that the matheuristic performance is consistent over all the executions.

Varying the number of periods

Then, a 4-period instance of the 8-grid network is considered. This time, some points solved by CPLEX with ε-constraint exceeded the time limit of 100 and 1 000 seconds, since the solutions found have average optimality gap equal to 0.06% and 0.02%, respectively. Thus, CPLEX already experiences convergence troubles for this small size case. Also, from the intermediate results (time limit per point equals to 100 seconds) shown in Table 6.2, it can be appreciated that both approximation sets obtained by the exact method present the same value of the hypervolume (0.8008), which may indicate that that both sets contain optimal solutions, even if convergence is not achieved (see their corresponding values of optimality gap).

With respect to the hybrid approach, the mean hypervolume values at both instants of the optimization (0.8099 and 0.8100, respectively) indicate a good performance for finding trade-off solutions of the HSC problem in all runs (standard deviation 0.0002). Note that at both instants, that is, at approximately 1 872 and 7 617 seconds, the hybrid approach outperforms the classical one because the solution set has a better distribution in the objective space. Moreover, it is important to emphasize that the evolutionary algorithm provides an approximation of the Pareto front that, if evaluated visually, could be considered as continuous (it contains 200 efficient solutions). This can be appreciated in Figure 6.5. In comparison with the single-period instance, the *TDC* objective for the HSC08g04p instance is significantly lower, which is explained by the learning rate factor that reduces capital costs in the subsequent

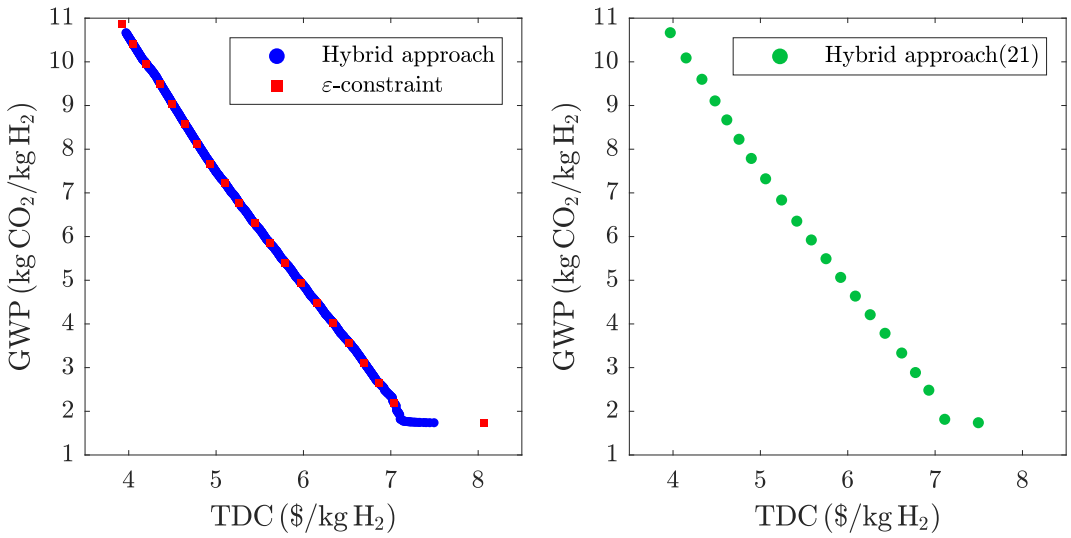


Figure 6.5: Final Pareto front approximations for HSC08g04p instance.

time periods, accounting for gained experience by technology manufacturers over time.

Now, regarding the 8-grid 7-period instance, the numerical results in Table 6.2 are once again in favor of the hybrid strategy. For the exact method, the optimality gap decreases between 100 and 1 000 seconds while the hypervolume remains steady, leading to the same conclusions as previously: the feasible solutions found might be optimal, but convergence cannot be guaranteed in a reasonable time. Moreover, due to the fact that the ε-constraint method performs an even partition of the objective space with respect to a given objective, the obtained efficient solutions, despite the possibility of being proven as optimal, might not be well distributed along the Pareto front. This explains the superiority of the hybrid approach, reflected by the values of the hypervolume indicator, which is signiicantly better for the matheuristic than for CPLEX (since the differences are much greater than the standard deviation of the hybrid hypervolume values).

Varying the grid number

The influence of the number of grids, associated to the spatial granularity of the territory partition, has also been studied. Thus, instances considering each main territory in the Midi-Pyrénées region are now discussed (22 grids). Regarding the HSC22g01p instance, the single time-period makes its combinatorial complexity similar to that of HSC08g07p. This can be confirmed by comparing the CPU times and optimality gaps for both instances in Table 6.2. Besides, the hypervolume indicator demonstrates a better performance of the hybrid approach for this instance, at both instants. The standard deviation values of the 11 runs performed by the MOEA are higher for the 22-grid instances than for those of 8 grids, but anyway, these values are much lower than the differences between the CPLEX and matheuristic approaches, suggesting that their difference is significant. Indeed, the hypervolume standard deviation is not significant considering the mean values, meaning that even the worst execution of the hybrid algorithm outperforms the exact method.

For the HSC22g04p instance, the required computational times increase consid-

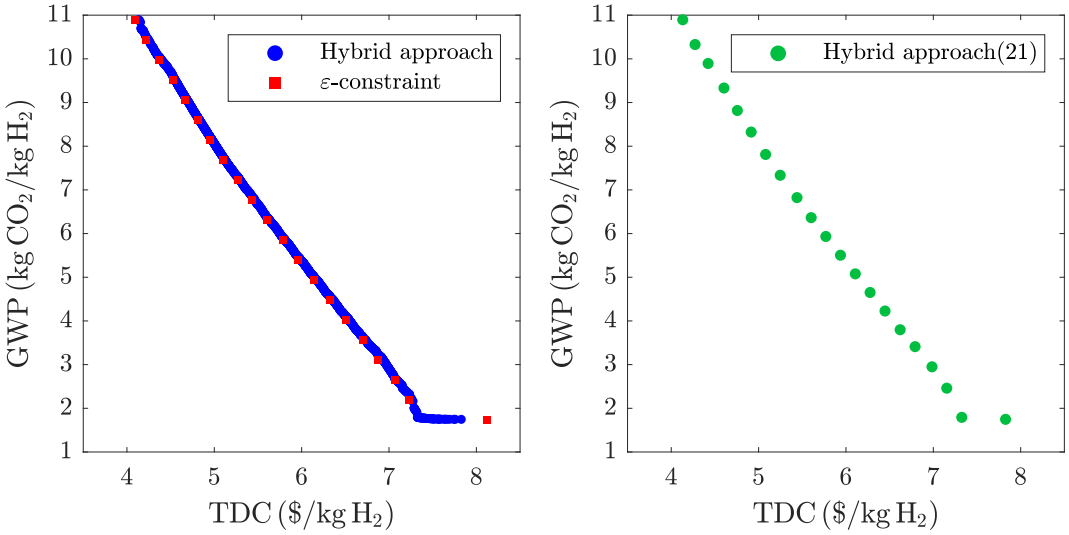


Figure 6.6: Final Pareto front approximations for HSC22g04p instance.

erably as well as the optimality gap for CPLEX. There is a significant difference between the obtained HV values at each instant, for both approaches: approximately 0.77 and 0.79, at 2 432 and 22 416 seconds, respectively. The mean HV values of the hybrid approach are greater than that of CPLEX with the ε-constraint method, however its standard deviation values do not allow to prove a significant superiority, when comparing equal number of solutions. Nevertheless, it is worth highlighting that the SMS-EMOA algorithm is able to provide 800 non-dominated solutions to decision-makers. The median solution according to the hypervolume obtained by the stochastic approach is plotted in Figure 6.6 together with the obtained solution by the deterministic method.

Finally, the largest instance, HSC22g07p describes the more realistic case study, with a forecast of hydrogen demand over 30 years split into 7 five-year periods considering the different territories in the Midi-Pyrénées region. Obtaining the proven optimal solution for every point in the Pareto approximation set using an exact approach would take prohibitive computational times, because of the large size of the search space. Table 6.2 highlights that for short CPU times the exact method does not find any feasible solution, contrary to the hybrid approach. For higher computational times (1 000 s. per point), the average optimality gap of the exact method's solution set is lower than 1%. The mean hypervolume value of the 11 runs performed by the hybrid approach is slightly superior to that of ε-constraint method. However, if the 800 non-dominated solutions identified by the matheuristic were accounted for in the hypervolume computation, a value of 0.7771 (not presented in Table 6.2) would be obtained, which is far superior to that of the exact technique (0.7555). The obtained Pareto fronts are shown in Figure 6.7. It can be observed that the "poor" distribution of the solution set obtained by CPLEX is located in the lower part of the Pareto front (high *TDC* values), where this method fails to identify accurately the knee-point of the front (as appreciated also for the other instances). Note that missing the knee part of the Pareto front (which encompasses the most interesting trade-off solutions) might be critical for the subsequent decision-making process.

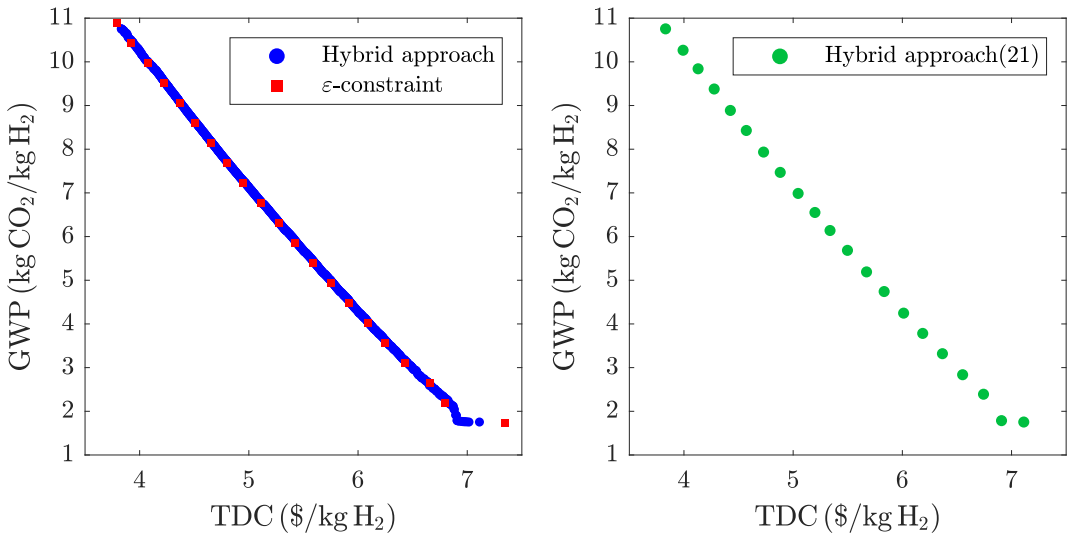


Figure 6.7: Final Pareto front approximations for HSC22g07p instance.

6.5 Conclusions

The optimal design of the hydrogen supply chain constitutes a current challenge to society, since it gives the basis for the evaluation of a cost-efficient hydrogen-based economy. Its design is far from being a trivial task, in particular when both economical and environmental aspects are considered. The mathematical model of the HSC involves combinatorial aspects that could make classical optimization methods inefficient for solving large-size instances of the problem, that is, for providing an accurate Pareto front approximation to decision-makers. Therefore, in this chapter a novel methodology for solving the HSC problem has been presented.

The hybrid solution methodology proposed is a matheuristic combining the benefits of multiobjective evolutionary algorithms and linear programming. This methodology efficiently solves the multiobjective HSC problem, providing the decision-makers with a set of efficient solutions well distributed along the Pareto front. Besides, this hybrid approach has proven to explore regions of the Pareto front that might be ignored when using exact techniques with the ε-constraint method, because this classical strategy can be constrained by the computational burden associated to the need for multiple executions to produce an approximation of the Pareto front. Moreover, according to the obtained results, the methodological framework is able to find good-quality solutions in short computational times even for large-size instances, contrary to ε-constraint method, which rapidly can become computationally prohibitive.

The computational efficiency exhibited by the proposed approach is based on the following aspects:

1. The decomposition of the original model into a bilevel optimization problem, such that combinatorial and linear aspects of the problem are treated separately. This allows the efficient solution of each subproblem (level) by appropriate solution methods.

2. The use of a MOEA for solving the multiobjective installation problem. Contrary to the ε-constraint method, the population-based algorithm solves the problem

in a collaborative way, i.e., information among the population is shared to evolve towards the true Pareto front.

3. The heuristic employed to construct the linear transportation problem. This reduces the complexity of the lower-level subproblem, as it reduces the number of potential exporting grids.

The solution strategy proposed here for the HSC design problem yet presents some limitations, thus providing potential improvement features. The first limitation regards the possibility of taking into account additional objectives: although the general structure of the hybrid algorithm can be maintained unchanged, SMS-EMOA might not be suited anymore, mainly because of the computational cost corresponding to the hypervolume calculation, which drastically increases with the number of objectives considered (Knowles and Corne, 2002). As a consequence, other paradigms for handling multiple objectives should be considered, for instance, decomposition-based MOEAs (see MOEA/D, Zhang and Li, 2007). Besides, the plant installation costs involve, in many practical contexts, nonlinear terms. Therefore, an extension of the present work to deal with nonlinear formulations constitutes a scope for improvement that will be tackled in the following chapter.

Chapter 7

Capturing spatial, temporal and technological detail in hydrogen supply chains

Contents

7.1 Introduction

In the previous chapter, a matheuristic solution method was introduced for the efficient design of sustainable hydrogen supply chains (HSC). The numerical results highlighted the efficiency of the hybrid approach in comparison with a classical approach, namely, weighted Tchebycheff function and ε-constraint method, on 9 increasing size instances. More precisely, the hybrid algorithm proposed a significant number of non-dominated solutions, and thus a more accurate approximation of the Pareto front was obtained, in reasonable computational times.

The aim of this chapter is to provide a more complete evaluation of the HSC by proposing some improvements to the previous model. The innovations of this chapter can be summarized in three main aspects, as follows:

1. The economy of scale for production and storage facilities are accounted for more appropriately through the sizing-up of a nominal facility.

2. The potential of cost reductions associated to technological learning, which is particularly important for low maturity technologies, is considered based on a recent work (Böhm et al., 2020).

3. Two alternative production technologies, namely, steam methane reforming with carbon capture, use and storage (SMR w/CCUS), and proton-exchange membrane water electrolysis (PEM) coupled with renewable electricity have been introduced in the set of technological options.

This chapter is organized as follows. The next section introduces the key innovations of this chapter. Section 7.3 presents the sizing-cost relationships used in the model. These changes to the base model are now analyzed from a global optimization perspective, identifying the sources of nonlinearities in Subsections 7.3.1 and 7.3.2. In Section 7.4, the solution of the proposed model by the matheuristic approach is briefly discussed, from the viewpoint of the efficient handling of nonlinearities by the MOEA. This solution methodology is validated in Section 7.5 through the study of eight increasing-size instances. The obtained results are analyzed and discussed in detail in Section 7.5.3 and, finally, conclusions are drawn in Section 7.6.

7.2 Key innovations

7.2.1 Economy of scale

The model considered so far (Almansoori and Shah, 2009; Almaraz et al., 2014a) only consider a range of discrete values for the sizing of production units, represented by index j in the former model. In that model, each one of the different sizes available has its corresponding capital and operating cost (CAPEX and OPEX, respectively). This formulation has the advantage of being of MILP type; however, the main drawback is that unit sizes are chosen from a set of possibilities, which can be restrictive (the $|J|$ available options). The facility sizing can be modeled in a simple and classical way following the power relationships typically used in chemical engineering (Peters et al., 2003). As a consequence, the size of the production and storage facilities are to be considered as continuous variables.

7.2.2 Learning rate

With respect to the cost reduction associated to the experience acquired over time by technology manufacturers, i.e., the learning rate, the previous model does not consider it as a technology-dependent factor, but only as a time-varying parameter. Based on the work of Böhm et al. (2020), in this chapter different learning rates for different production technologies are introduced. For example, the learning rate used for electrolysis-based technologies should allow for a more important cost reduction over time than that for steam reforming technology, which is an already mature technology. Moreover, the cost of energy feedstock is modeled in a similar manner: the levelized cost of energy (LCOE) for renewable energy sources, like wind onshore or solar photo-voltaic, is a time-varying parameter, considering that these renewable energies are expected to grow under the studied horizon time (Haeusler et al., 2020).

7.2.3 Additional process alternatives

Additionally, two additional low-carbon hydrogen production technologies, namely, SMR w/CCUS and PEM, which represent relevant production pathways from both economic and environmental perspectives, will be considered as process alternatives. The hydrogen produced by SMR w/CCUS consists in a low-carbon pathway because CO_2 can be captured before or after it is emitted (directly from the air), followed by a permanent geological storage or uses of CO_2 (IEA, 2019). This hydrogen, produced from fossil fuels with CO_2 emissions reduced, is commonly labelled as "blue" hydrogen. Concerning PEM electrolyzers, they are based on solid polymer membranes technologies. The main advantages they present, compared to alkaline electrolyzers (AEL), include faster cold start and higher flexibility, thus representing a better option for intermittent systems (Götz et al., 2016). Even if this technology is currently more expensive than AEL systems, forecasts estimate that the PEM technology will overtake AEL in a near future, then becoming the cheapest solution (Böhm et al., 2020).

7.2.4 Impact on problem formulation

From an optimization perspective, the extended methodology will now involve additional decision variables representing the plant and storage capacities and equations correlating sizing and cost of these facilities. Typically, these sizing-cost relations use a scale factor as an exponent of the ratio between a given size equipment and the questioned size equipment. Therefore, the resulting HSC problem becomes of MINLP type and its solution is more complex than the base MILP model. The matheuristic approach introduced in the previous chapter is particularly relevant for solving this type of problems, mainly because the changes made to the base model only involve upper-level variables treated by the evolutionary algorithm, that is well suited for managing nonlinearities, whereas the lower-level problem (production and transportation problem) will remain almost of the same complexity.

7.3 Modeling of scaling and learning effects

According to Peters et al. (2003, p. 242) the equipment cost of a particular size or capacity can be estimated by using the relationship known as the *six-tenths factor rule*, providing that the size and cost data of a similar equipment is available. According to this rule, the cost of a new equipment item is approximated as the cost of a known equipment item times the ratio of their capacities with an exponent f, as formulated in the following equation:

$$C_b = C_a \left(\frac{S_b}{S_a} \right)^f \tag{7.1}$$

where C_b is the cost of the considered equipment and S_b is its respective size, while C_a and S_a represent the cost and size of the known reference, respectively.

In Böhm et al. (2020) the relationship in equation 7.1 is applied to the estimation of electrolysis-based technologies. The authors indicated that the scale factor values depend on the system size and may vary from one technology to another, for alkaline technology, for instance, f can take values ranging from 0.51-0.96, whereas for

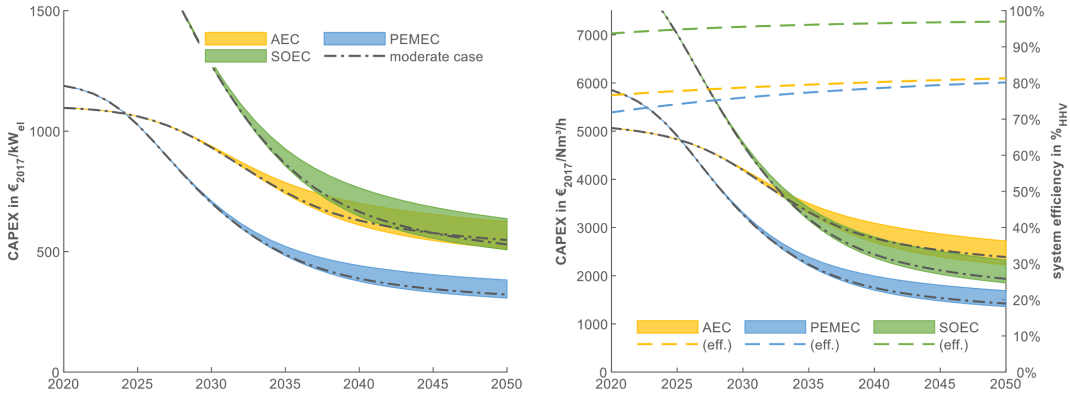


Figure 7.1: Estimated ranges for technological learning of electrolysis (left: based on electric power; right: based on hydrogen output with developing efficiencies). Taken from Böhm et al. (2020).

proton exchange technologies, f typically ranges from 0.53-0.97. Therefore, using the nomenclature used in the previous chapters, the CAPEX of establishing a production plant, PCC_{pigt}, of a given capacity, $PCap_{pigt}$, can be estimated as:

$$PCC_{pigt} = PCC_{pit}^{\text{ref}} \left(\frac{PCap_{pigt}}{PCap_{pit}^{\text{ref}}} \right)^{r_p} \qquad \forall p \in P, i \in I, \forall g \in G, \forall t \in T \tag{7.2}$$

where $PCap_{pit}^{\text{ref}}$ and PCC_{pit}^{ref} are, respectively, the production capacity of a reference plant and its associated capital cost, and r_p is the scaling parameter. In this work, the CAPEX is considered in euros (EUR), the capacity of an electrolyzer in MW (electrical input), whereas the capacity of a SMR plant is given in kg of H_2 per day.

From equation (7.2), it can be observed that different installation costs can be estimated depending on the period of time t at which a given plant is installed, with the condition that the corresponding data of a reference plant is available. In Böhm et al. (2020), the authors investigated the dynamic installation costs for electrolysis technologies considering potential demands of power-to-gas systems. The obtained results indicated significant CAPEX reductions in the studied time horizon (2020-2050) considering a nominal power plant of 5 MW_{el}, see Figure 7.1. It can be noted that the estimates show that PEM (or PEMEC in the figure) electrolysis is about to undercut AEL (AEC in the figure) in the near future (before 2025). In this respect, this work adopted the profile of the learning rates for AEL and PEM technologies proposed by (ibid.). The data related to steam reforming, in particular for production capacity and cost of a reference plant was taken from NREL (2018). For this technology, no cost reduction associated to technological learning was considered, since it is an already mature technology.

Now, considering the operational costs (OPEX) associated to a production plant of a given size, this latter is usually estimated as a function of the capital cost, more precisely, expressed as a percentage of the CAPEX per year (E&E Consultant et al., 2014). Thus, the unit production cost UPC_{pigt}, which is the cost of producing one unit (kg) of hydrogen, involves the operational cost as well as the cost associated to

the energy and material feedstock. It can be calculated as:

$$UPC_{pigt} = \beta_{pigt}\frac{PCC_{pigt}}{\alpha PCap_{pigt}} + \gamma_{pigt}EC_{pigt} \qquad \forall p \in P,\, i \in I,\, \forall g \in G,\, \forall t \in T \qquad (7.3)$$

where β_{pigt} is the OPEX parameter ranging from 0.03 for PEM electrolysis (ibid.), to 0.15 for SMR w/CCUS (NREL, 2018), α is the network operating period, in days per year, γ_{pigt} is a conversion factor representing the energy and material feedstock needed for producing a unit of hydrogen, and EC_{pigt} is the corresponding unit cost.

Regarding the installation and operation costs of storage facilities, they can be estimated using the six-tenths factor rule, similarly to the calculation for production plants. In this sense, the capital cost of establishing a storage facility, SCC_{sigt}, is computed as a function of the storage unit's capacity, $SCap_{sigt}$, considering the nominal capacity of a reference storage facility, $SCap^{\text{ref}}_{sit}$, and its related capital cost, SCC^{ref}_{sit}, as:

$$SCC_{sigt} = SCC^{\text{ref}}_{sit}\left(\frac{SCap_{sigt}}{SCap^{\text{ref}}_{sit}}\right)^{r_s} \qquad \forall p \in P,\, i \in I,\, \forall g \in G,\, \forall t \in T \qquad (7.4)$$

where adequate values for parameters $SCap^{\text{ref}}_{sit}$, SCC^{ref}_{sit} and r_s, can be found in the literature, see for example Beccali et al. (2013).

Also, the unit storage cost (USC_{sigt}) is computed as a function of the storage capacity (E&E Consultant et al., 2014), i.e., as a function of its capital cost (SCC_{sigt}), so that the bigger the storage facility, the lower the unit storage cost. It is calculated as:

$$USC_{sigt} = \beta_{sigt}\frac{SCC_{sigt}}{\alpha SCap_{sigt}} \qquad \forall s \in S,\, i \in I,\, \forall g \in G,\, \forall t \in T \qquad (7.5)$$

where β_{sigt} is the OPEX parameter.

7.3.1 Sources of nonlinearities

In the proposed model, equations (7.2-7.5) are now used to correlate installation and operation costs to production and storage capacities. In this way, the facility capacities, $PCap_{pigt}$ and $SCap_{sigt}$, are considered as continuous decision variables. Therefore, equations (7.2) and (7.4) are nonlinear and non-convex because of the exponent of the scale parameter (taking positive values inferior to 1). Further, equations (7.3) and (7.5) are also nonlinear, not only because the sizing variables appear in the denominator, but also because the variables associated to installation costs, PCC_{pigt} and SCC_{sigt}, are now considered as (dependent) decision variables. Besides, the modifications carried out give rise to nonlinearities elsewhere in the original model, namely, the facility capital cost of the entire network FCC_t, which was calculated according to equation (5.3) in page 113, and involves now bilinear terms, and equation (5.3) becomes:

$$FCC_t = \sum_{i\in I}\sum_{g\in G}\left(\sum_{p\in P} PCC_{pigt}IP_{pigt} + \sum_{s\in S} SCC_{sigt}IS_{sigt}\right) \qquad \forall t \in T \qquad (7.6)$$

Note that both the first and second terms in equation (7.6) are nonlinear and non-convex, since variables IP_{pigt} and IS_{sigt} are discrete.

Additionally, the facility operating cost (see equation 5.7 in page 114), which considers the cost associated to the operation of each production plant and storage facility, involves also product bilinear terms. Thus:

$$FOC_t = \sum_{i\in I}\sum_{g\in G}\left(\sum_{p\in P} UPC_{pigt}P_{pigt} + \sum_{s\in S} USC_{sigt}S^T_{igt}\right) \qquad \forall t \in T \tag{7.7}$$

In this case, only the first term inside the brackets represents a nonlinear term because, in the second term, S^T_{igt} is not a decision variable but a parameter.

7.3.2 Mathematical model

The novel mathematical representation of the bi-objective hydrogen supply chain design (HSC) proposed i this chapter is as follows:

$$\min_{\mathbf{x}} \quad [TDC(\mathbf{x}), GWP(\mathbf{x})]^T \tag{7.8}$$

$$\text{s.t.} \quad \sum_{p\in P}\sum_{g\in G} PCap^{\min}_{pigt} NP_{pigt} - \sum_{g\in G} D_{igt} \leq 0 \qquad \forall i \in I, \forall t \in T \tag{7.9}$$

$$\sum_{g\in G} D_{igt} - \sum_{p\in P}\sum_{g\in G} PCap^{\max}_{pigt} NP_{pigt} \leq 0 \qquad \forall i \in I, \forall t \in T \tag{7.10}$$

$$\sum_{s\in S} SCap^{\min}_{sigt} NS_{sigt} - S^T_{igt} \leq 0 \qquad \forall i \in I, \forall g \in G, \forall t \in T \tag{7.11}$$

$$S^T_{igt} - \sum_{s\in S} SCap^{\max}_{sigt} NS_{sigt} \leq 0 \qquad \forall i \in I, \forall g \in G, \forall t \in T \tag{7.12}$$

$$PCap^{\min}_{pigt} NP_{pigt} - P_{pigt} \leq 0 \qquad \forall p \in P, \forall i \in I, \forall g \in G, \forall t \in T \tag{7.13}$$

$$P_{pigt} - PCap^{\max}_{pigt} NP_{pigt} \leq 0 \qquad \forall p \in P, \forall i \in I, \forall g \in G, \forall t \in T \tag{7.14}$$

$$\sum_{p\in P} P_{pigt} - \sum_{l\in L}\sum_{g'\in G, g'\neq g} \left(Q_{ilgg't} - Q_{ilg'gt}\right) - D_{igt} = 0 \qquad \forall i \in I, \forall g \in G, \forall t \in T \tag{7.15}$$

$$PCap^{\text{lo}}_{pigt} \leq PCap_{pigt} \leq PCap^{\text{up}}_{pigt} \qquad \forall p \in P, \forall i \in I, \forall g \in G, \forall t \in T \tag{7.16}$$

$$SCap^{\text{lo}}_{sigt} \leq SCap_{sigt} \leq SCap^{\text{up}}_{sigt} \qquad \forall s \in S, \forall i \in I, \forall g \in G, \forall t \in T \tag{7.17}$$

$$NP_{pigt}, NS_{sigt} \in \mathbb{N} \qquad \forall p \in P, \forall s \in S, \forall i \in I, \forall g \in G, \forall t \in T \tag{7.18}$$

$$PCap_{pigt}, SCap_{sigt} \in \mathbb{R}_{\geq 0} \qquad \forall p \in P, \forall s \in S, \forall i \in I, \forall g \in G, \forall t \in T \tag{7.19}$$

$$P_{pigt}, Q_{ilgg'} \in \mathbb{R}_{\geq 0} \qquad \forall p \in P, \forall i \in I, \forall l \in L, \forall g \in G, \forall t \in T \tag{7.20}$$

$$\mathbf{x} = \left[PCap_{pigt}, SCap_{sigt}, NP_{pigt}, NS_{sigt}, P_{pigt}, Q_{ilgg't}\right]^T \tag{7.21}$$

where $\mathbf{x}$ in equation (7.8) represents the vector of decision variables, i.e., $\mathbf{x} = [PCap_{pigt}, SCap_{sigt}, NP_{pigt}, NS_{sigt}, P_{pigt}, Q_{ilgg't}]^T$, as stated in equation (7.21). For readability purposes, the complete model is not presented here, nevertheless, the reader can easily hint the missing equations, presented in Chapter 5. These are equations (5.1-5.14), with the exception of equations (5.3) and (5.7), and the addition of (7.2-7.7).

Also, the assumption is made that production plants installed in the same grid, in the same period of time and with the same technology type, have necessarily the same capacity $PCap_{pigt}$. In this way, for instance, the variable NP_{pigt} indicates the number of production plants with identical specifications in each grid and in each period. This reduces the problem complexity, yet constituting a realistic assumption, e.g., the installation of two facilities in the same grid and in the same period are expected not to have significant differences in sizing, since both facilities have to satisfy the same hydrogen demand.

Again, regarding sources of nonlinearities in the model, note that constraints (7.9-7.14) involve nonlinear and non-convex bi-linear product terms, because the operating conditions limits of the facilities depend on their actual sizes, $PCap_{pigt}$ and $SCap_{sigt}$. More precisely, the facility operating conditions are dictated for each given technology, for example, considering an alkaline electrolysis, it can operate at least at 20% of its nominal capacity to a maximum operating condition of 110% of its nominal capacity (E&E Consultant et al., 2014). The constraints (7.16) and (7.17) impose realistic bounds, in terms of capacity, for establishing production and storage facilities, respectively.

7.4 Solution strategy

7.4.1 Handling nonlinearities at the upper level

The idea behind the bilevel decomposition is to transform the original problem into two subproblems of tractable complexity, such that each aspect of the original problem can be tackled by appropriate solution approaches. As for the previous chapter, the HSC design problem can be decomposed into (1) the problem related to facility installation (upper level) and (2) the continuous problem associated to production and transportation rates (lower level). Here, equations (7.8-7.12) and (7.16-7.19) are involved in the upper-level problem, with decision variables $PCap_{pigt}$, $SCap_{sigt}$, NP_{pigt} and NS_{sigt}; whereas the lower-level is described by equations (7.8), (7.13-7.15) and (7.20), with decision variables P_{pigt} and $Q_{ilgg't}$. Within the matheuristic strategy, as previously, the upper-level problem is tackled by a MOEA and, with respect to the lower-level problem, it is transformed into a canonical transportation problem applying the (already described) heuristic, to be solved efficiently by an exact LP solver.

Yet, from the analysis of the sources of nonlinearities presented in Subsection 7.3.1,the decision variables involved in nonlinear equations (7.2-7.7) are $PCap_{pigt}$, $SCap_{sigt}$, NP_{pigt} and NS_{sigt}, which all belong to the upper level. With regards to the other constraints containing nonlinear terms (7.13-7.15), they are at the lower-level, but they become linear once some variables ($PCap_{pigt}$) are set to some value by the upper-level algorithm. Thus, every nonlinear function can be easily handled by the evolutionary algorithm.

7.4.2 Impact on the lower-level problem

The changes to the base model have also an impact on the transportation and production problem. As mentioned before, the lower-level problem is described by equations (7.8), (7.13-7.15) and (7.20), with decision variables P_{pigt} and $Q_{ilgg't}$. Therefore, if compared to the previous linear model (see Section 6.10), it is clear that the

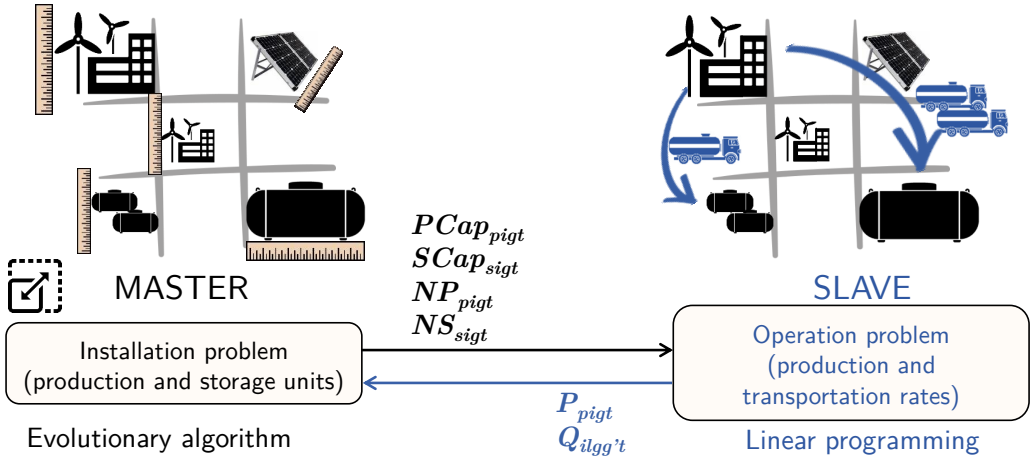


Figure 7.2: Simplified diagram of the hybrid (master-slave) approach for the improved HSC model. Upper-level variables in black and lower-level variables in blue.

mathematical representation of the lower level has not changed, i.e., equations (7.13-7.15) match almost exactly with equations (6.8-6.10).

The only difference is the subindex j associated to the size type of the production facility in variable P_{pjigt} in equations(6.8-6.10). Although this might seem an insignificant difference, it might entail important effects on computational times, in particular for instances with a high number of periods. To better explain this, let us consider the installation of production facilities in the previous MILP model. The number of production variables, of the same physical form type i, at each grid g, at each period t, can be at most $|P| \times |J|$. Then, assuming the worst case in which production plants of all types are installed in every grid, the overall number of production variables implicated in the evaluation of an upper-level partial solution is $|P| \times |J| \times |I| \times |G| \times |T|$. On the contrary, in the proposed model, the number of production variables with the same physical form type i, at each grid g, at each period t, has a nonlinear relationship with the number of periods considered. For instance, in each grid, in the first period, there are at most $|P|$ production variables (production levels are to be determined for one period); in the second period, there are at most $|P| \times 2$ production variables (production levels are to be determined for two periods); in the n-th period, there are at most $|P| \times n$ production variables (production levels are to be determined for n periods). Therefore, the overall number of production variables involved in the lower-level operation subproblem is at most $|P| \times |I| \times |G| \times |T| \times (|T| + 1)/2$.

7.5 Computational experiments

The efficiency of the proposed matheuristic methodology for solving the bi-objective HSC problem is validated through the solution of eight growing size instances of the problem. Contrary to the experiments on the previous linear formulation, in which a performance comparison of both hybrid and classical approaches was performed, no computation through exact techniques based on mathematical programming was carried in this non-linear case. Indeed, in the present chapter, the aim is to apply the hybrid solution strategy developed earlier, without major changes, to the new formulation, in order to propose practical solutions for this case study. Moreover,

some preliminary experiments using GAMS solvers showed that the determination of an initial feasible solution is a difficult task, so that it did not seem justified to devote additional efforts for this class of methods that, as shown in the previous chapter, is outperformed by our matheuristic approach.

7.5.1 Case study 3

The instances for the HSC problem considered in this work correspond to the data for the case study of Midi-Pyrénées territory in France (Almaraz et al., 2014a). In Figure 7.3, a summary of the case study data is shown. This time, four different periods of time are studied for each 8 and 22-grid instances, that is, the entire planning horizon is set to year 2050, with different periods divisions: 1 period (2050), 2 periods (2020, 2050), 4 periods (2020, 2030, 2040, 2050), and 7 periods (2020, 2025, 2030, 2035, 2040, 2045, 2050). The data used for different production technologies as well as the data employed for energy sources is provided in Appendix B.

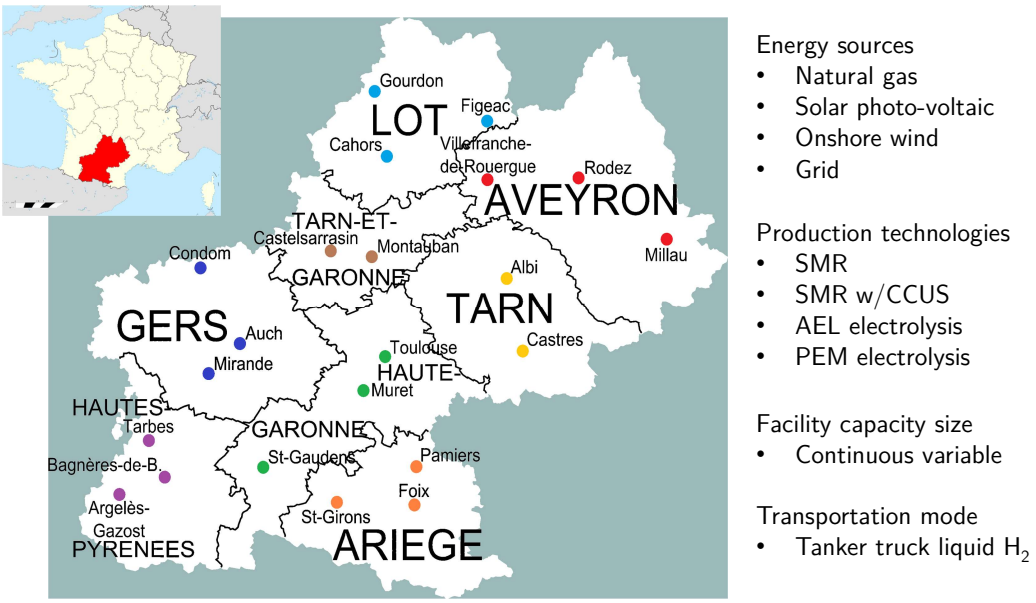


Figure 7.3: Midi-Pyrénées region case study. Names in capitals indicate departments (grids) for 8-grid instances. Color points correspond to grid cities (in lower case) for 22-grid instances.

7.5.2 Parameter settings

Regarding the operating parameters of the matheuristic used as a solution tool, the population size of the MOEA is set to 1 000 for all instances according to a preliminary sensitivity analysis; the stopping criterion (number of function evaluations) is set to 2×10^6 for all instances. The variation operators use the following (standard)

Table 7.1: Numerical results of the hybrid solution approach for case study 3.

Instance	UL variables		HV					CPU
	Discr.	Cont.	best	median	worst	mean	std	time(s)
nHSC08g01p	72	72	0.4790	0.4692	0.4606	0.4693	0.0051	642.4
nHSC08g02p	144	144	0.9226	0.9215	0.9200	0.9213	0.0019	1 397.6
nHSC08g04p	288	288	0.9354	0.9321	0.9311	0.9326	0.0014	5 542.3
nHSC08g07p	504	504	0.9724	0.9717	0.9673	0.9707	0.0022	22 640.2
nHSC22g01p	198	198	0.3854	0.3569	0.3380	0.3608	0.0163	3 324.5
nHSC22g02p	396	396	0.8746	0.8695	0.8645	0.8689	0.0029	8 552.3
nHSC22g04p	792	792	0.8825	0.8816	0.8806	0.8815	0.0013	41 527.9
nHSC22g07p	1386	1386	0.9139	0.9114	0.9097	0.9122	0.0035	164 310.1

parameters: SBX operator's probability and distribution are $p_c = 1$ and $\eta_c = 20$, respectively; polynomial mutation's probability and distribution are equal to $p_m = 1/n$ and $\eta_m = 20$, respectively, where n is the number of decision variables. The reference point for the hypervolume indicator (HV) computation is set at $[3.3, 12.1]^T$ in the objective space, for all instances.

As for the linear case, SMS-EMOA was implemented in MATLAB language (vR2019a) and the solution of the LP subproblem is performed using the CPLEX solver v12.8.0, called from MATLAB. Also, all the computational experiments were carried out with a processor Intel Xeon E3-1505M v6 at 3.00 GHz and 32 Go RAM.

7.5.3 Results and discussion

The obtained numerical results are presented in Table 7.1. It can be appreciated that, in the upper-level subproblem, the number of discrete and continuous variables are equal for each instance, because of the hypothesis made about the equal capacities of installed facilities of the same type in the same grid and in the same period. That is, these upper-level variables represent the number and size of each different production/storage technology (p/s), at each potential location grid (g), in each period (t). Also, the column presenting the CPU times (for each execution) in Table 7.1 suggests that the solution of the proposed model requires higher computational times than those required for the linear model (see Table 6.2 at page 132). This trend is particularly obvious for medium-to-large size instance, like nHSC08g07p or those containing 22 grids, because the multi-period aspect of the problem echoes in the lower-level subproblems, as discussed in Section 7.4. Indeed, for the largest instance, more than 45 hours are required for one single execution. The statistical results regarding the hypervolume indicator are provided in Table 7.1. The low values of the standard deviation indicates the robustness of the algorithm, which allows to infer that convergence has been reached.

However, apart from an evaluation of the numerical results obtained over all instances and rather than an assessment of the optimization technique performance levels, a focus is proposed in what follows on the particular solutions obtained and their practical interpretation, in the framework of the tackled case study. For this purpose, instance nHSC08g07p is analyzed here, because it provides general guidelines for the optimal design of the HSC over time (it contains seven 5-years

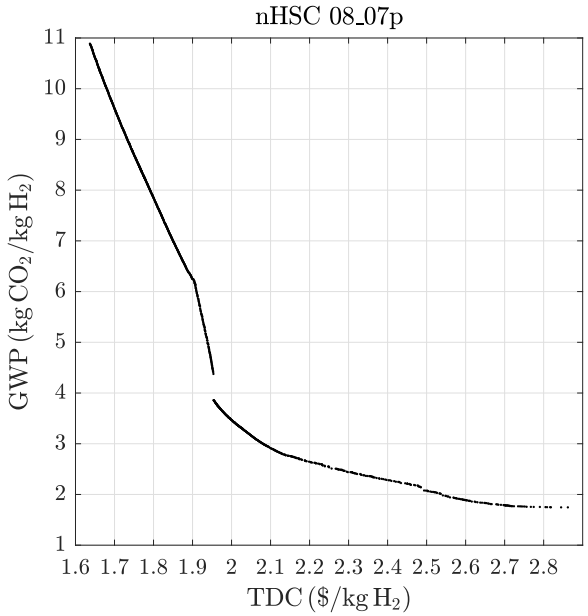


Figure 7.4: Final Pareto front approximation of nHSC08g07p instance.

time periods), but it does not constitute a large instance (only 8 grids), for which a detailed representation would be too complex to present here. Figure 7.4 presents the Pareto front approximation obtained for this instance is shown (corrsponding to the median solution set, with respect to hypervolume). It includes 1 000 non-dominated solutions, each one describing a specific HSC design and its evolution over the time horizon considered (2020-2050), which includes: the strategic decisions about size, type, location and operation of production and storage technologies, as well as the logistics features defining the transportation of hydrogen among grids, for each time period. Besides, please note that both *TDC* and *GWP* objectives are presented per unit of H_2 produced, this is done by dividing the original objectives by the total hydrogen demand. It can be observed that the non-dominated solutions found lie in the range 1.64 - 2.86 USD/kgH_2 for the TDC objective, with a global warming potential of 10.89 - 1.75 kgCO_2-eq/kgH_2, respectively. The shape of the resulting Pareto front is worth being analyzed: for low TDC values, it is composed of two linear segments with different slopes. Then, as TDC increases, a disconnection in the front is observed, followed by nonlinear segments corresponding to the lowest GWP values.

In order to analyze some candidate solutions, the Pareto front approximation is divided into five segments for further analysis. As mentioned above, the partition is carried out according to the Pareto front shape. In this way,

- segment 1 comprises non-dominated solutions with a TDC objective ranging from 1.64 - 1.90 USD/kgH_2 TDC;
- segment 2 has non-dominated solutions with a TDC objective ranging from 1.90 - 1.95 USD/kgH_2 TDC;
- segment 3 with a TDC objective ranging from 1.95 - 2.13 USD/kgH_2 TDC;
- segment 4 with a TDC objective ranging from 2.13 - 2.49 USD/kgH_2 TDC;

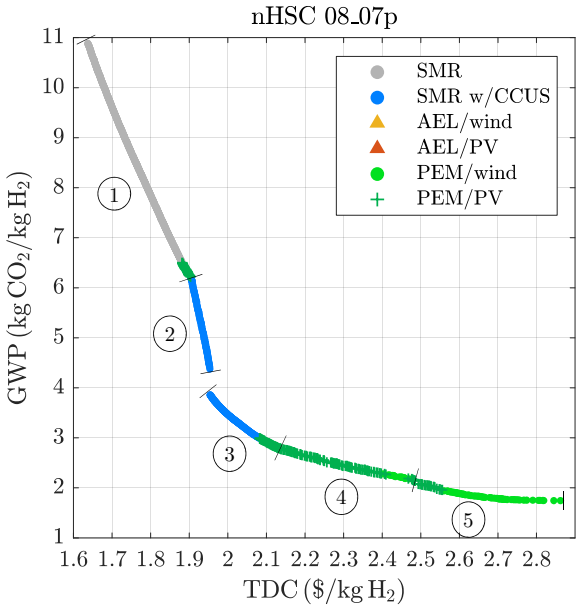


Figure 7.5: Final Pareto front approximation of nHSC08g07p instance. Colors indicate the majority of production technology used for hydrogen demand satisfaction.

- and segment 5 with a TDC objective ranging from 2.49 - 2.86 USD/kgH_2 TDC.

In Figure 7.5, this division into segments can be appreciated visually. Besides, each solution is assigned a color depending on the major production technology used for satisfying the hydrogen demand over the whole time horizon. Accordingly, the solutions in the left part of the front (low TDC values) mostly employ the SMR technology for satisfying hydrogen demand, whereas the solutions in the right part of the front (low GWP values) mainly promote the use of PEM electrolysis powered by wind energy. Now, with the aim of presenting the evolution of the HSC design in further details, one solution is selected from each one of the five segments. To this end, the AASF scalarazing function (Pescador-Rojas et al., 2017) is employed in order to determine the most relevant solution of each segment, that is, a solution with a balanced trade-off for both objectives. In this perspective, each objective is normalized within each segment and equal weights are used for each objective.

Analysis of the AASF solution from segment 1

The AASF solution shown in Figure 7.6 corresponds to a 1.76 USD/kgH_2 averaged total daily cost (i.e., the averaged cost over all periods) and it produces 8.46 kgCO_2-eq/kgH_2. Most of the hydrogen (considering all time periods) is produced from the steam methane reforming technology. In Figure 7.6, the deployment of the HSC in the studied region is displayed for every period. The arrows indicate transportation between grids, while the pie chart in each period indicates the percentage of hydrogen demand satisfied by each production technology. Also, the colored squares in each grid represent production units, depending on the employed technology; the corresponding number indicates the cumulative number (over past periods) of installed production plants. The storage facilities are not shown in the chart.

It can be observed that, for this solution, only two different technologies are used: SMR and PEM electrolysis powered by solar photo-voltaic. However, the PEM

technology does not appear before the sixth period, that is, when it constitutes a technology mature enough to constitute a good option from an economic point of view (according to the data employed). Also, it is worth mentioning that only one production plant is able to satisfy the hydrogen demand for the first five periods, because SMR plants can exhibit relatively large production capacities, in comparison to the low demand in these periods. The next plant installation is indeed carried out in the 6th period only (i.e., 35 years later). The only SMR production unit installed is located at the grid corresponding to the Haute-Garonne department (where hydrogen demand is the most important). In this way, the transportation costs are minimized.

Analysis of the AASF solution from segment 2

As observed in Figure 7.5, segment 2 contains only solutions that produce a majority of "blue" hydrogen for satisfying the network demand. The details of the solution chosen through the ASSF are shown in Figure 7.7. The corresponding objective values are 1.93 USD/kgH_2 for the averaged total daily cost and 5.36 kgCO_2-eq/kgH_2 produced for the GWP. This solution presents the singularity that plant installation only occurs in the first period: one SMR w/o CCUS and one SMR w/CCUS production plants. Besides, the proportion of the hydrogen demand satisfaction for these two plants is the same in every period, 21% for SMR and 79% for SMR w/CCUS. Note that this does not necessarily mean that the production rates for these two plants are the same in each period, because the hydrogen demand increases over time. Regarding the logistics features, the transportation routes remain logically the same over all periods, since no plant is installed apart from the first period. Again, the installation of production plants takes place in the department with the highest population density and, thus, with the highest hydrogen demand expectation.

Analysis of the AASF solution from segment 3

With respect to the third segment, the selected solution produces mostly either "blue" or "green" hydrogen (see segment 3 in Figure 7.5). This solution is detailed in Figure 7.8, with an averaged TDC of 2.03 USD/kgH_2 and a GWP equal to 3.27 kgCO_2-eq/kgH_2. In the first period, blue hydrogen (from SMR w/CCUS) is produced, in one production site to satisfy the demand in the whole network. It can be appreciated that the plant installed in the first period has a capacity size sufficient to satisfy the demand in the next four periods. In the sixth period, PEM water electrolysis then represents the major technology, producing 63% of the total hydrogen demand in the network. Contrary to SMR production plants, which have the potential to produce large amounts of hydrogen, electrolysis is constrained by the technology in itself and also by the intermittent availability of the energy source, described by the charge factor. Thus, in the sixth period, 11 electrolyzers are installed to satisfy the hydrogen demand and, in the last period, one more is finally installed. It is important to highlight that, as mentioned before, the installation and operational costs for electrolyzers are time-varying parameters. This explains in part the installation of electrolyzers in later periods, when the technology has become competitive, or even more, the cheapest option for hydrogen production.

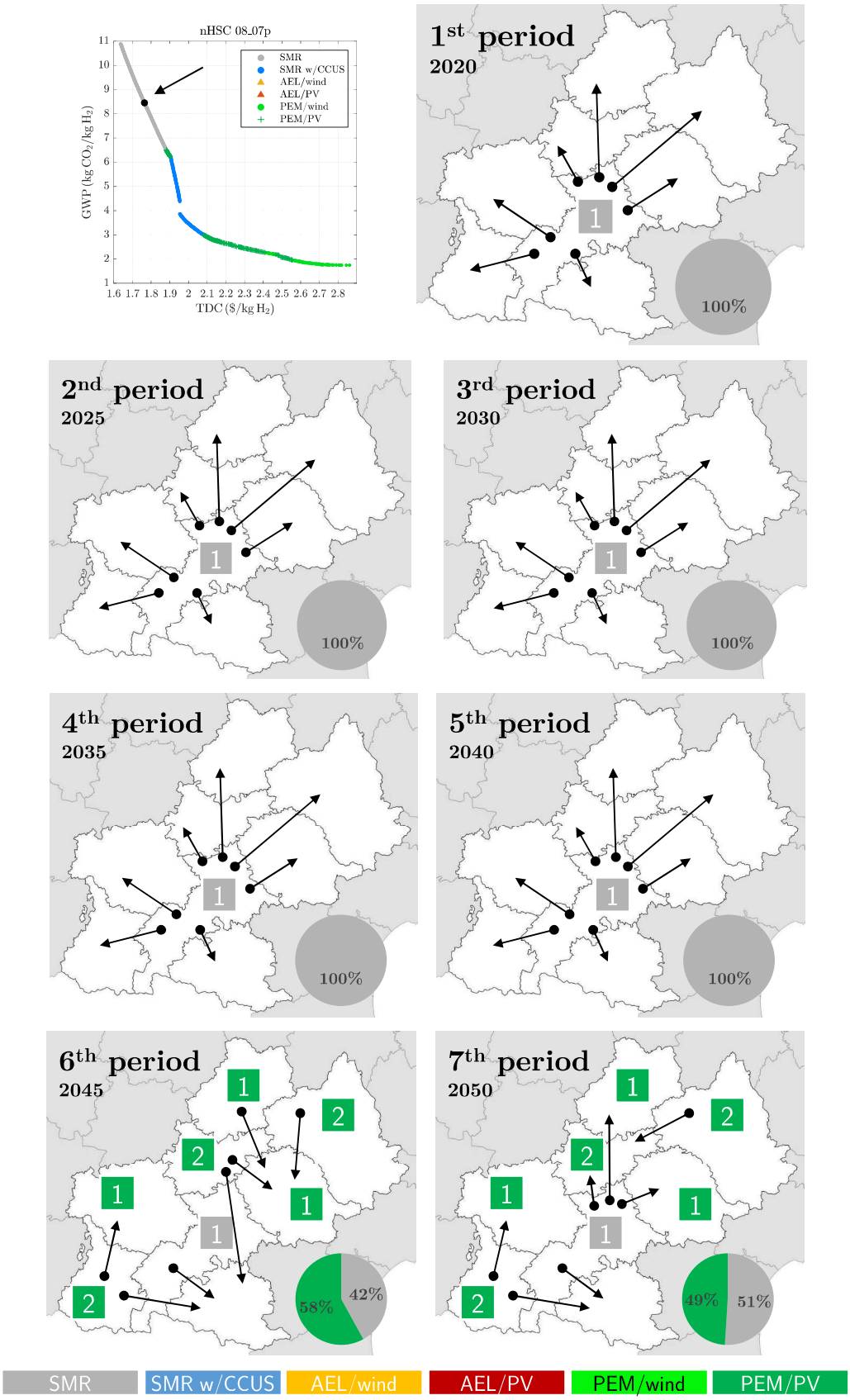


Figure 7.6: Detailed chosen solution on segment 1 for nHSC08g07p instance. Squares indicate the cumulative number of production units; arrows between grids indicate transport flows; pie charts indicate demand satisfaction by production type.

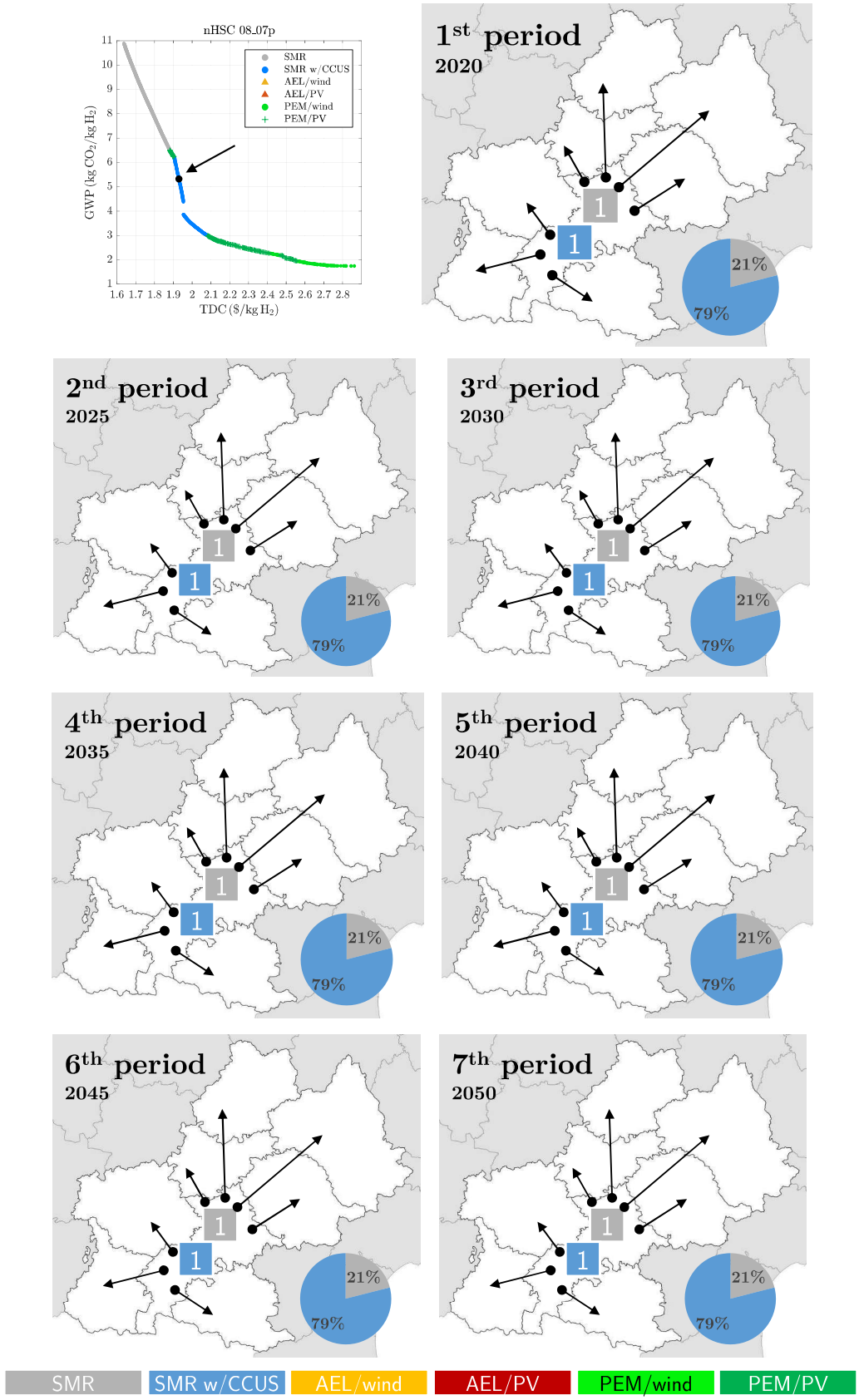


Figure 7.7: Detailed chosen solution on segment 2 for nHSC08g07p instance. Squares indicate the cumulative number of production units; arrows between grids indicate transport flows; pie charts indicate demand satisfaction by production type.

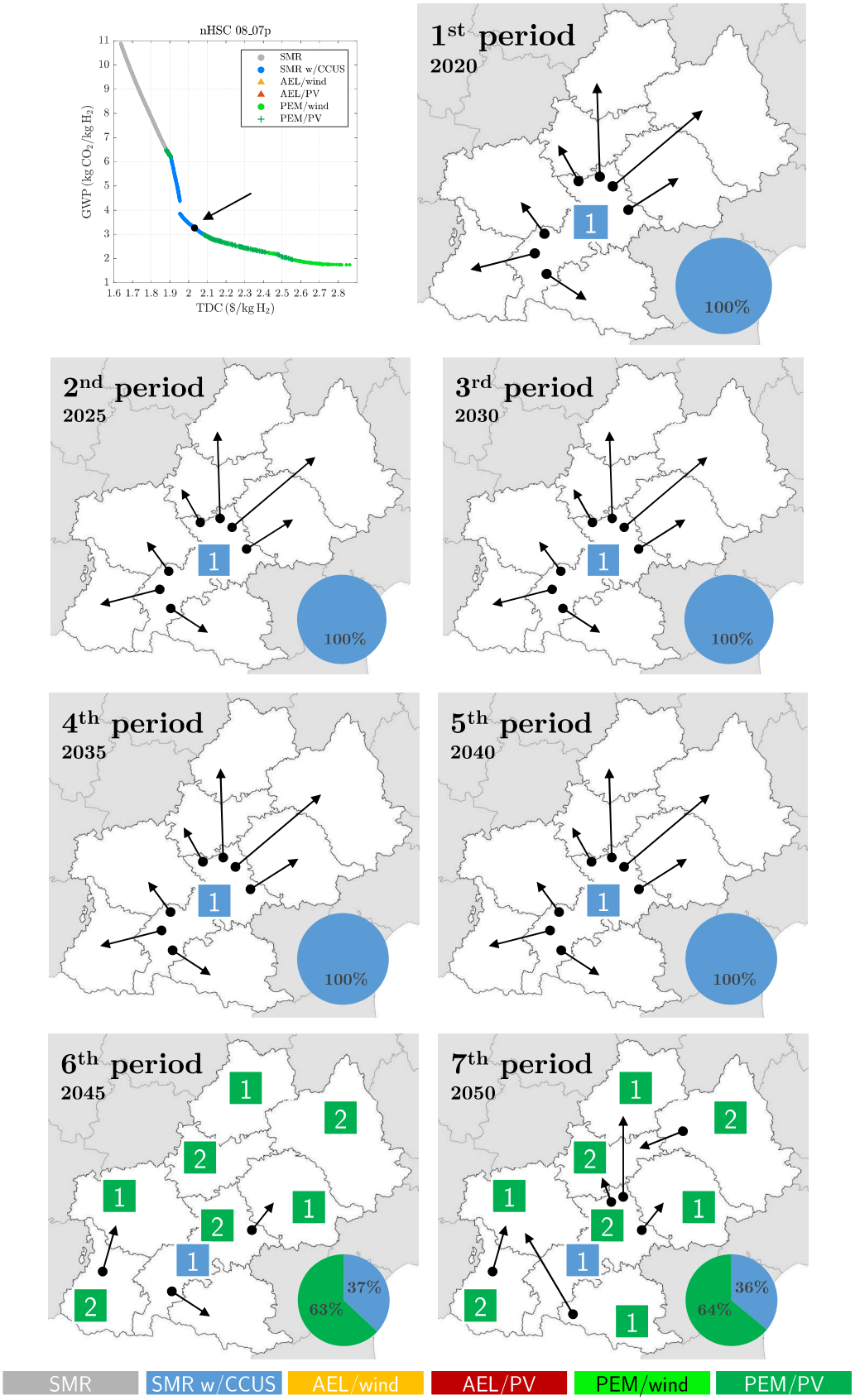


Figure 7.8: Detailed chosen solution on segment 3 for nHSC08g07p instance. Squares indicate the cumulative number of production units; arrows between grids indicate transport flows; pie charts indicate demand satisfaction by production type.

Analysis of the AASF solution from segment 4

The solution selected in segment 4 is detailed in Figure 7.5. This solution produces the majority of hydrogen by PEM electrolysis employing electricity generated by solar photo-voltaic systems. The averaged TDC is of 2.30 USD/kgH_2 and entails 2.44 kgCO_2-eq/kgH_2. In the first two periods (first ten years), blue hydrogen is produced in one single production site, located at the grid corresponding to the Haute-Garonne department. In the third period, an electrolysis of type PEM, powered by wind energy, is installed in the grid corresponding to the Haute-Pyrénées department and produces 17% of the overall hydrogen demand in the network. From that production site, hydrogen is transported towards two other grids (Auch and Ariège). In the fourth period, three additional PEM/wind electrolyzers are installed, with hydrogen distribution to Aveyron, Tarn et Haute-Garonne departments. In that period PEM/wind technology becomes the most important production technology, covering 61% of the network demand. In the fifth period, three different production technologies interact, namely, SMR w/CCUS, PEM/wind and PEM/PV. It is important to recall that, according to the data used, from period 3, electricity generated by solar PV technology is less expensive than wind onshore, but has a higher GWP than wind energy, according to a LCA analysis (Mehmeti et al., 2018; Muteri et al., 2020). This explains why PEM/PV does not appear before period 5: in periods 3 and 4, the cost of electricity from wind energy is slightly higher than that from PV energy, but the benefits in the environmental objective are more important than those of the economic criterion. In contrast, this trend begins to change in period 5. Periods 6 and 7 show a significant installation of PEM/PV electrolysis, satisfying more than 85% of the network demand. Besides, it is observed that, contrary to the detailed solutions from segments 1-3, the installation of electrolyzers is carried out progressively, accounting for the fact that low capital costs are attained in late periods of time.

Analysis of the AASF solution from segment 5

Segment 5 comprises non-dominated solutions that only employ water electrolysis technology to satisfy the growing hydrogen demand. On the one hand, these solutions can be considered as the most expensive alternatives for deploying the hydrogen supply chain, and, on the other hand, they involve the best green roadmaps for the HSC. The AASF solution from this part of the approximation front is displayed in Figure 7.10, with an averaged cost of 2.62 USD/kgH_2 and producing 1.86 kgCO_2-eq/kgH_2. This time, alkaline technology is employed, in particular that using electricity from wind energy. In the first period, 33% and 67% of the H_2 demand is fulfilled by AEL/wind and PEM/wind, respectively. Note that the AEL production drops from 33% to only 6% (of the total demand), from the first to the second period. This is because the membrane technology turns out to be less expensive than alkaline from the second period, and thus, the former becomes the most appropriate production technology. Also, since the mathematical model imposes constraints on the production rate, the three AEL electrolyzers installed in the first period continue working in the subsequent periods at their lower bound capacities. In periods five and six, the network shows that the installation of PEM electrolysis coupled with photo-voltaic systems. Nevertheless, the majority of hydrogen is still produced by PEM/wind technology.

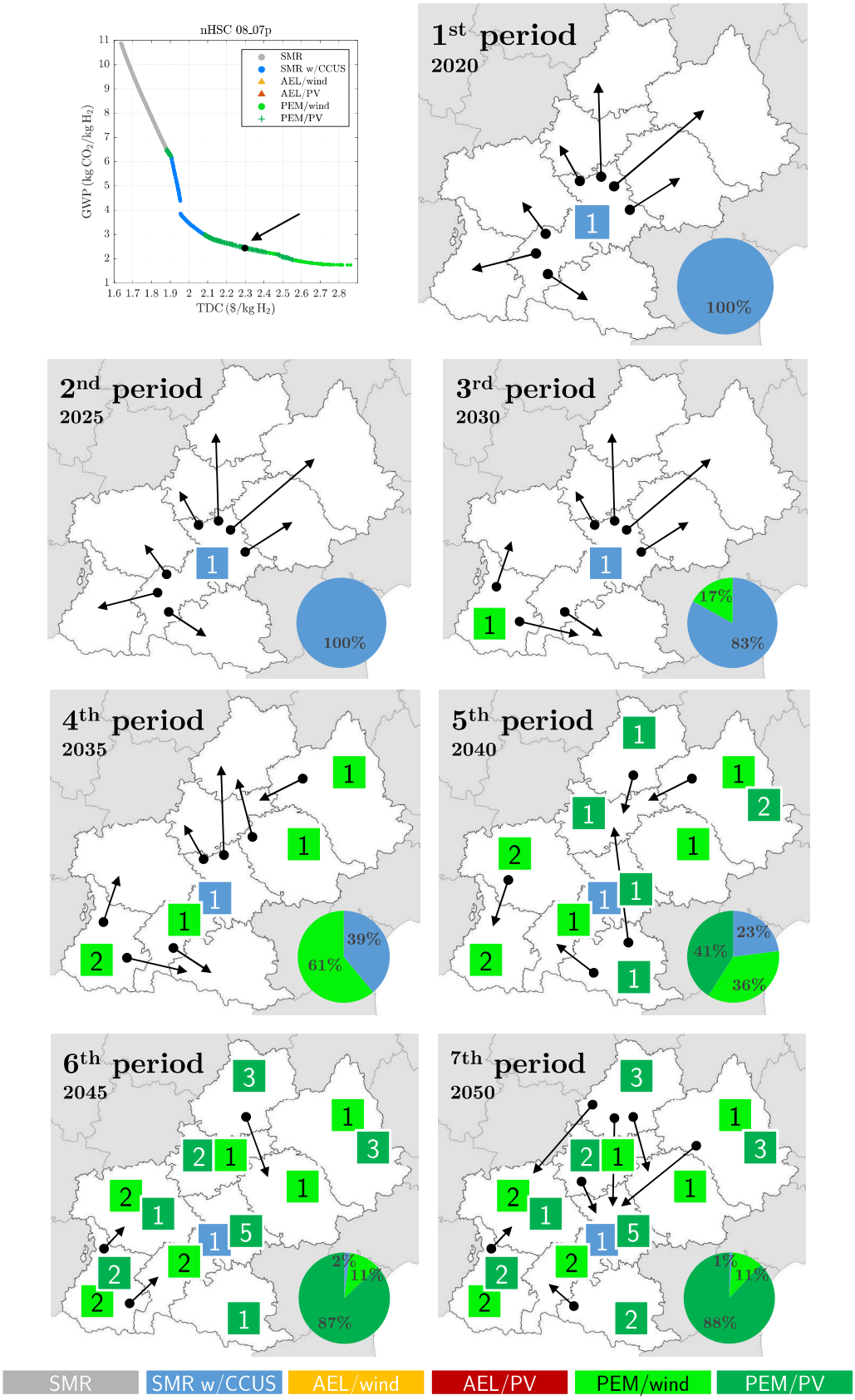


Figure 7.9: Detailed chosen solution on segment 4 for nHSC08g07p instance. Squares indicate the cumulative number of production units; arrows between grids indicate transport flows; pie charts indicate demand satisfaction by production type.

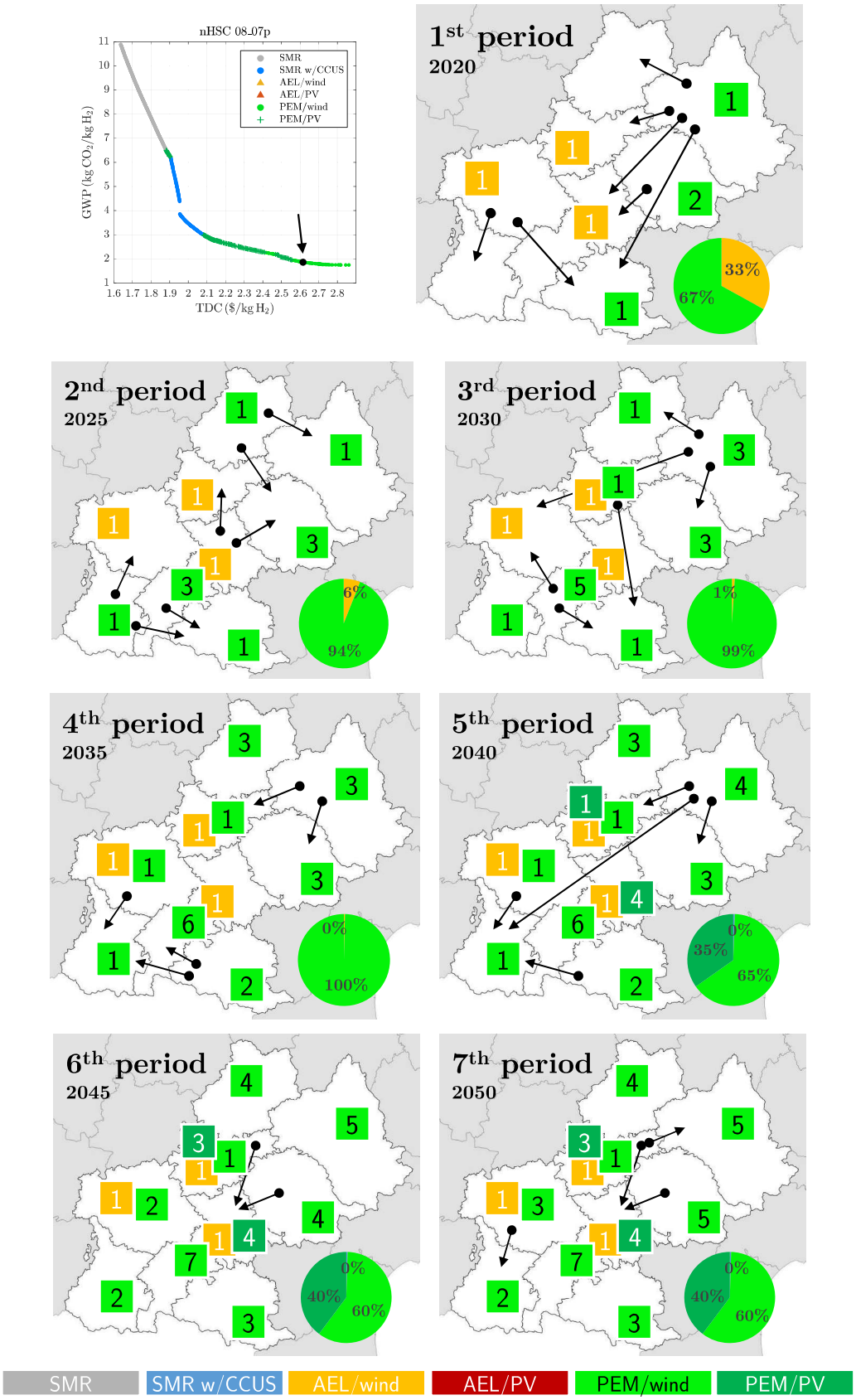


Figure 7.10: Detailed chosen solution on segment 5 for nHSC08g07p instance. Squares indicate the cumulative number of production units; arrows between grids indicate transport flows; pie charts indicate demand satisfaction by production type.

7.6 Conclusions

An innovative formulation of the HSC problem has been proposed and solved in this chapter, using sizing-cost relationships in order to better account for the scaling and learning effects of production and storage facilities. Consequently, the HSC design problem is now represented through a MINLP formulation. The hybrid solution approach introduced in the previous chapter was adapted and investigated to tackle this new challenging problem.

To validate this approach, the optimal design of the HSC is studied for eight instances of different sizes, corresponding to the Midi-Pyrénées territory in France. The obtained numerical results showed the robustness of the matheuristic algorithm, able to handle nonlinear terms and to produce an approximation of the Pareto front in controllable computational times. The results obtained provide valuable insights for the optimal deployment of the HSC. The solutions involving a relatively low hydrogen cost suggest the necessity of fossil-based production technologies in early periods of time and, subsequently, electrolysis technologies are gradually being installed. Further, the results obtained are in accordance with those of the devoted literature (Hydrogen Council, 2017; IEA, 2019), which indicate the necessity of different production technologies at different time periods, for example, a production based on SMR w/CCUS on the first years aiming to launch the hydrogen economy, and then, a progressive transition pathway to cleaner solutions, such as water electrolysis.

Finally, the proposed approach is particularly well-suited to capture the complexity of hydrogen supply chains through their technological and spatial integration. Their modeling and the optimization of their territorial integration should make them resilient. The multi-objective issue was addressed here from a bi-objective perspective, cost and greenhouse gas emissions. Its extension to the development of a multi-criteria approach including other environmental criteria as well as territorial specificities (existence of hydrogen producers, hydrogen consumers in industry and in mobility, volumes concerned, dispersion of these consumption points), socio-economic and regulatory objectives is an interesting perspective to explore. This framework seems attractive to consider the prospective dimension for hydrogen deployment, since the hydrogen infrastructure develops the needs for hydrogen evolve and the technological maturity of the different bricks and the deployment of local renewable energies in line with the local production of decarbonized hydrogen.

Conclusions and perspectives for future work

The main motivation of this research was to explore alternative solution methods for addressing complex multiobjective optimization problems related to the Process Systems Engineering area. In particular, advantage is taken from the recent advances in Evolutionary Computation, which is only used marginally in the framework of PSE applications formulated as multiobjective optimization problems. However, the adaptation of metaheuristic-based tools to this class of problems is not straightforward, mainly because these problems typically involve many equality and inequality constraints, which represent a source of difficulties, emphasized in a multiobjective optimization context. Therefore, this work was divided into two main parts. The first one consisted in an empirical study, performed in the perspective of comparing different algorithmic configurations and selecting the best one for providing accurate approximations of Pareto sets. In a second part, the lessons learned from the first numerical part were re-used to design a novel matheuristic solution tool, adapted to the problem of hydrogen supply chains (HSC) design. In the following, the main contributions of this thesis are described in more details.

First, a systematic comparison of five representative constraint-handling techniques in evolutionary algorithms was carried out in a single-objective framework, studying 14 test problems related to PSE applications. This chapter explored the performance of constraint-handling schemes that have not been employed in this area, which in general considers only penalty functions, feasibility rules or problem reformulations. The numerical experiments highlighted the superiority of a repair method based on constraint gradients, which enables the algorithm to evolve the population across severely constrained search spaces.

In a second step, the performance of this repair technique was investigated in the context of multiobjective optimization, notably for the first time in the literature. To this end, six classical constraint-handling techniques were embedded within two state-of-the-art multiobjective evolutionary algorithms (MOEAs), namely NSGA-II and MOEA/D. Numerical experiments, performed over a benchmark including academic test problems with inequality (CF, LIR-CMOP) and equality constraints (EQC, Eq-DTLZ and Eq-IDTLZ), demonstrated that the gradient-based repair technique outperforms all its contenders, in particular for the solution of problems containing equality constraints. This technique renders the MOEA capable of exploring constrained search spaces as it uses additional information to push infeasible individuals towards feasible search spaces without affecting population diversity, which proved to be a condition for obtaining a satisfactory description of the whole Pareto fronts.

In a second part, the promising results obtained on benchmark functions were thus explored to the optimal design of hydrogen supply chains, considered as a representative problem related to PSE. This optimization problem is of particular importance since much of the future expansion of hydrogen utilization depends on the HSC deployment. Its mathematical representation, implying mixed-integer variables, a considerable number of constraints and two performance criteria (the total cost and the environmental impact), entails difficulties to both exact and stochastic methods. In a first attempt, ε-MOEA/D coupled with the gradient-based repair method was

investigated as a solution approach to this problem. The results indicated a poor performance of this algorithm for finding solutions along the true Pareto front. Indeed, due to the number of mass balances involving multiple decision variables, the MOEA experienced convergence difficulties, in spite of the repair of infeasible solutions.

Nevertheless, these first experiments paved the way to the design of a novel matheuristic methodology. Within this strategy, a bilevel reformulation is adopted, allowing for the decomposition of the original problem into master and slave subproblems, each one being treated by appropriate solution methods. In this way, on the one hand, the combinatorial and multiobjective features of the master subproblem, which addresses the HSC structure design, are effectively handled using a MOEA. On the other hand, the slave subproblem, which deals with the HSC operation (transportation and production) subproblem, concentrates all the problem equality constraints. This subproblem is efficiently tackled by a LP solver, using a scalarizing function to handle the two objectives accounted for. Performance comparisons with a classical approach are carried out on two real case studies of HSC deployment (from UK and from the Midi Pyrénées region in France), resulting in nine instances of increasing size. The results obtained suggested that the matheuristic approach is an efficient solution method for the optimal design of the HSC, able to provide decision-makers with an accurate approximation of the Pareto front.

Finally, a modification to the problem formulation was proposed in order to deal with a more realistic representation of the HSC, accounting for sizing-cost relationships and, thus, for the proper modeling of scaling effects. Besides, different learning rates for different production technologies were taken into account. The resulting model involved additional decision variables (accounting for facility sizing) and nonlinear terms, making the model a MINLP problem. The obtained results provided spatial, temporal and technological details of the HSC, evidencing the necessity of different hydrogen production pathways at different periods, closely related to the different levels of maturity of each different technology. Concerning the solution approach, the designed matheuristic showed robustness for handling nonlinear terms and providing solutions in reasonable computational times.

Answers to the research questions

This dissertation investigated four research questions. After carrying out the related substantial research, it is now possible to answer them directly.

1. *What is the best solution approach to tackle current PSE problems, typically formulated as multiobjective optimization problems?*

   Depending on the problem formulation, especially the number of objectives and the type and number of constraints, mathematical programming methods can be efficient for solving optimization problems related to PSE. Nevertheless, these methods may require intractable computational times in problems where multiple criteria and combinatorial aspects are present, mainly because they require a transformation of the multiobjective problem into a set of scalar subproblems. In particular, tuning the scalarizing parameters (for instance, weight vectors or ε-levels) turns to be a harsh task for producing efficiently a set of solutions uniformly distributed along the Pareto front. Regarding metaheuristics, and particularly multiobjective evolutionary algorithms,

they are alternatives to MP techniques, able to construct an approximation of the Pareto front in a collaborative way in one single run. The handling of equality constraints, however, constitutes their main drawback, even more if these constraints involve many continuous variables, since the stochastic variation operators employed are likely to produce infeasible solutions.
Therefore, hybrid algorithms represent promising alternative techniques, proposed to face the drawbacks encountered for each solution paradigm, and simultaneously to take advantage of their strengths. In this work, a matheuristic has been successfully designed to study the optimal design of hydrogen supply chains and, according to the obtained results, it outperforms both exact and stochastic approaches for this problem. Even though the No Free Lunch theorem (Wolpert and Macready, 1997) does not allow to extend directly these conclusions to any other problem, this hybridization strategy seems however adapted to similar production and distribution problems from the PSE area.

2. *Considering the recent advances in the metaheuristics field, in particular those related to constraint-handling, do they constitute efficient techniques for treating highly-constrained problems as those related to PSE?*

   In this work, only some representative (by their working mode) constraint-handling methods were explored. Regarding the repair mechanism based on constraints' gradient information, it is noteworthy that it demonstrated its efficacy and efficiency for the solution of academic benchmark problems (studied in chapters 2 and 3), but obtained poor results when addressing intricate and large-size PSE applications. Indeed, the MOEA employed in Chapter 5 as a first attempt to solve the HSC design problem had convergence troubles and did not show a satisfying robustness enough for this difficult problem containing equality constraints involving multiple decision variables. Thus, this difference in result quality observed when treating academic test functions and real-world applications suggests that a significant research effort is needed for the design of relevant test functions representing a stronger challenge to metaheuristic techniques and reflecting the sources of complexity appearing in engineering problems.

3. *Can a matheuristic approach be designed for the solution of the HSC design problem, so that it outperforms classical exact methods, thus providing solutions in tractable computational times?*

   Regarding the satisfying results obtained by the hybrid approach developed and validated in Chapter 6, the answer is clearly "yes". The strategy proposed in this work follows a bilevel decomposition that promotes the design and use of matheuristic algorithms. In particular, a MOEA was in charge of the Pareto front construction and of the combinatorial aspect of the problem (by handling the integer variables), and worked on a search space constrained only by some inequality constraints that are easy to fulfill. Besides, equality constraints were efficiently handled through a linear programming solver, for each partial solution proposed by the MOEA. Besides, this approach allowed the solution of a more realistic model for the design of HSC, which captures additional features like the scaling and learning effects of each different

production pathway and now involves nonlinear terms, also controlled by the MOEA.

4. *And finally, can a matheuristic approach be designed as a general solution tool for a broad range of PSE problems?*

   The design of a matheuristic approach is problem dependent, meaning that particular encoding is needed for each specific problem. Nevertheless, the principles employed in this work for the matheuristic design can be replicated to other PSE problems. Thanks to the present work, some general guidelines for such design can be stated. First, a preliminary analysis of the different features included in the treated system representation is required to identify substantial aspects that explain the problem complexity, e.g., number of objectives, types/number of constraints, types/number of decision variables, etc. Typically, it might be appropriate to use a metaheuristic to address the combinatorial aspects of the problem, in addition to the handling of the multiobjective aspect, for which different algorithmic paradigms may be considered depending on the number of objectives involved. In addition, since exact methods, due to their working mode, efficiently deal with constraint satisfaction issues, constraints that are difficult to satisfy by metaheuristics (particularly, equality constraints) can be left to a MP technique. Moreover, considering the efficiency of linear programming solvers, it might be useful to construct a linear programming problem and to handle nonlinear terms with the metaheuristic search engine.

Perspectives for future work

Some perspectives for future work have been identified. They encompass five different axes described as follows:

1. To investigate the proposed hybrid methodology for the optimal design of hydrogen supply chains with more than two objectives, e.g., accounting for social or risk objectives, or even several environmental criteria, different from the global warming potential considered in this work. Apart from the modeling of these criteria, this consideration of more objectives is not straightforward to carry out using the hybrid solution approach developed here. Indeed, the SMS-EMOA algorithm should be replaced by another MOEA for the Pareto front construction in the upper level, for instance, MOEA/D, because the computation of the hypervolume contribution for more than two objectives requires considerable computational resources. Regarding the lower-level subproblem, the scalarizing function used within the LP formulation is still valid when working with more than two objectives, but the choice of the weight vector parametrizing the obtained scalar problem may become a critical issue (see also point 4 at this respect).

2. There is an interest to explore the proposed matheuristic algorithm as a solution tool for the study of the HSC design optimization problem from the game theory perspective. In this context, each actor of the supply chain (retailers, factories, distributors, etc.) appears at different decision levels, and the problem can indeed be formulated as a bilevel optimization problem, with different

objectives at each level. Since the proposed decomposition approach performs a problem reformulation using a bilevel decomposition, it seems particularly suited to address this problem. This approach is currently under the scope of the thesis of Manuel Flores Pérez, developed at the Laboratoire de Génie Chimique at Toulouse.

3. Expanding the scope of the matheuristic solution approach proposed here to other PSE case studies also constitutes a promising perspective. In this way, it may be possible to propose a general solution tool for a family of problems, i.e., problems presenting similar characteristics. For instance, process integration optimization problems (like heat exchanger or waste water networks design, eco-industrial parks, etc.) are typically of high complexity, of multiobjective nature, involve mixed-integer variables and nonlinear and non-convex terms, thus requiring global optimization procedures that might be time-consuming.

4. The fourth axis for future work is the improvement of the matheuristic algorithm, in order to consider more appropriately the multiobjective nature of the lower-level subproblem. In this thesis, a single (randomly chosen) weight vector is used for the solution of the lower-level (operation subproblem). However, it is possible to imagine that several solutions, corresponding to the same upper-level partial solution (HSC structure design) but obtained with different weight vectors, participate to the Pareto optimal front. Formally, the solution of the lower-level problem is actually a sub-front, which may be generated by using a set of weight vectors within the scalarizing function. As a consequence, the upper-level algorithm should be able to handle not a single solution, but rather a set of solutions corresponding to the above-mentioned sub-front. So, the design of a novel evolutionary algorithm would be necessary to adapt its working mode. It is worth mentioning that such an investigation is the subject of the M.Sc. thesis of Rubén Arguedas, currently in development at the Laboratoire de Génie Chimique at Toulouse.

5. We would finally note that this optimization framework approach is sufficiently general to be extended to more complex hydrogen energy supply chains (currently under the scope of Renato Luise's thesis, Luise et al. (2021, accepted)), including power-to-gas, power-to-heat, power-to-liquids, and power-to-ammonia. Indeed, one of the most challenging tasks for energy system models is to capture the full degree of variability and complexity that exists in such systems. As more and more renewable sources such as solar and wind are introduced to contribute to decarbonization, systems are increasingly spatially distributed, technologically diverse, and time-varying. At the same time, new technologies and increased interconnectivity are enabling greater interaction between different energy sectors. To ensure that energy system models not only provide an accurate representation of these systems but also do not miss the potential of new technologies such as hydrogen-based ones, they must capture the requisite level of temporal, spatial, technological, and cross-sectoral detail, thus requiring efficient optimization strategies.

It is our hope that this research work has given valuable results in the context of mathematical optimization, with particular applicability in the Process Systems Engineering area with a focus on Energy Systems Engineering.

Conclusions et perspectives

La motivation principale de cette thèse était d'explorer des méthodes de résolution alternatives pour traiter des problèmes d'optimisation multiobjectif complexes liés au domaine des Procédés et Systèmes Industriels (PSI). En particulier, nous avons tiré parti des progrès récents en Algorithmes Évolutionnaires, qui ne sont utilisés que de façon marginale dans le cadre d'applications au génie des procédés formulées comme des problèmes d'optimisation multiobjectif. Cependant, l'adaptation des outils métaheuristiques à cette classe de problèmes n'est pas simple, principalement parce que ces problèmes impliquent typiquement de nombreuses contraintes égalité et inégalité, qui représentent une source de difficultés, d'autant plus dans un contexte d'optimisation multiobjectif. Par conséquent, ce travail a été divisé en deux parties principales. La première consiste en une étude empirique, réalisée dans la perspective de comparer différentes configurations algorithmiques et de sélectionner la meilleure pour fournir des approximations précises des ensembles de solutions Pareto-optimales. Dans une deuxième partie, les leçons tirées de la première partie numérique ont été réutilisées pour concevoir un nouvel outil de solution matheuristique, adapté au problème de la conception des chaînes d'approvisionnement en hydrogène (HSC). Dans ce qui suit, les principales contributions de cette thèse sont décrites plus en détail.

Tout d'abord, une comparaison systématique de cinq techniques représentatives de traitement des contraintes dans les algorithmes évolutionnaires a été effectuée dans un cadre mono-objectif, à travers l'étude de 14 problèmes test, liés à des applications en PSI. Ce chapitre a permis d'évaluer les performances de mécanismes de gestion des contraintes qui n'ont pas été employés dans ce domaine, qui ne prend généralement en compte que les fonctions de pénalité, les règles de faisabilité ou la reformulation des problèmes. Les expériences numériques réalisées ont mis en évidence la supériorité d'une méthode de réparation basée sur le gradient de contraintes, qui permet à l'algorithme de faire évoluer la population à travers des espaces de recherche fortement contraints.

Dans un deuxième temps, les performances de cette technique de réparation par gradient ont été étudiées dans un contexte d'optimisation multiobjectif, pour la première fois dans la littérature. À cette fin, six techniques classiques de gestion des contraintes ont été intégrées à deux algorithmes évolutionnaires multiobjectif (MOEA) de l'état-de-l'art, à savoir NSGA-II et MOEA/D. Des expériences numériques, réalisées sur un benchmark comprenant des problèmes test académiques, incluant des contraintes inégalité (CF, LIRCMOP) et égalité (EQC, EqDTLZ et EqIDTLZ), ont démontré que la technique de réparation par gradient obtient de meilleurs résultats que tous ses concurrents, en particulier pour la résolution de problèmes contenant des contraintes égalité. Cette technique rend le MOEA capable d'explorer des espaces de recherche contraints car elle utilise des informations supplémentaires pour pousser les individus infaisables vers des espaces de recherche faisables, sans affecter la diversité de la population, qui s'avère être une condition indispensable pour obtenir une description satisfaisante du front de Pareto.

Dans une deuxième partie, les résultats prometteurs obtenus sur les fonctions de référence ont ainsi été explorés pour la conception optimale des chaînes d'approvisionnement en hydrogène, considérées comme un problème représentatif lié

au domaine des PSI. Ce problème d'optimisation est d'une importance particulière, car une grande partie de l'expansion future de l'utilisation de l'hydrogène dépend du déploiement de sa chaîne d'approvisionnement. Sa représentation mathématique, qui implique des variables mixtes, un nombre considérable de contraintes et deux critères de performance (le coût total et l'impact environnemental), pose en pratique des difficultés aux méthodes exactes et stochastiques. Dans une première tentative, l'algorithme ε-MOEA/D, couplée à la méthode de réparation par gradient, est proposée pour résoudre ce problème. Les résultats ont indiqué une mauvaise performance de cette approche, incapable pour trouver des solutions sur le front de Pareto optimal. En effet, en raison du nombre de bilans massiques (contraintes égalité) impliquant de nombreuses variables de décision, le MOEA a montré des problèmes de convergence, malgré la réparation des solutions infaisables.

Néanmoins, ces premières expériences ont ouvert la voie à la conception d'une nouvelle méthodologie matheuristique. Selon cette stratégie, une reformulation bi-niveaux est adoptée, permettant la décomposition du problème original en deux sous-problèmes, maître et esclave, chacun étant traité par des méthodes de résolution appropriées. Ainsi, les caractéristiques combinatoire et multi-objectif du sous-problème maître, qui concerne la conception de la structure de la HSC, sont traitées efficacement à l'aide d'un MOEA. D'autre part, le sous-problème esclave, qui traite le sous-problème d'opération de la HSC (transport et production), concentre toutes les contraintes égalité du problème. Ce sous-problème est traité efficacement par un solveur LP, au moyen d'une fonction de scalarisation pour la gestion des deux objectifs pris en compte. Les performances de cet algorithme hybride sont comparées avec celles d'une approche classique sur deux cas réels de déploiement de la HSC au Royaume-Uni et dans la région Midi Pyrénées en France, dérivant en neuf instances de taille croissante. Les résultats obtenus suggèrent que l'approche matheuristique est une méthode de résolution efficace pour le problème de conception optimale de la HSC, capable de fournir aux décideurs une approximation précise du front de Pareto.

Enfin, une modification de la formulation du problème a été proposée afin de traiter une représentation plus réaliste de la HSC, en tenant compte des relations taille-coût et, ainsi, de la modélisation appropriée des effets d'échelle. En outre, des taux d'apprentissage différents pour les différentes technologies de production ont été pris en compte. Le modèle résultant implique des variables de décision supplémentaires (associées au dimensionnement des installations) et des termes non linéaires, transformant ainsi le modèle en un problème de type MINLP. Les résultats obtenus ont permis de fournir les détails spatiaux, temporels et technologiques du déploiement de la HSC, mettant en évidence la nécessité de différents modes de production d'hydrogène à différentes périodes, étroitement liés aux niveaux de maturité de chaque technologie. En ce qui concerne la stratégie de résolution, la matheuristique conçue a montré sa robustesse pour traiter les termes non linéaires et fournir des solutions non-dominées en des temps de calcul raisonnables.

Réponses aux questions de recherche

Cette thèse a soulevé quatre questions de recherche. Après les travaux de recherche correspondants effectués, il est maintenant possible d'y répondre directement.

1. *Quelle est la meilleure approche pour résoudre les problèmes actuels de PSI, généralement formulés comme des problèmes d'optimisation multi-objectif ?*

Selon la formulation du problème, notamment le nombre d'objectifs, ainsi que le type et le nombre de contraintes, les méthodes de programmation mathématique peuvent être efficaces pour résoudre les problèmes d'optimisation liés au domaine des PSI. Néanmoins, ces méthodes peuvent exiger des temps de calcul prohibitifs pour les problèmes impliquant plusieurs critères de performance et des aspects combinatoires, principalement parce qu'elles nécessitent une transformation du problème multiobjectif en un ensemble de sous-problèmes scalaires. En particulier, l'ajustement des paramètres de scalarisation (par exemple, les vecteurs de poids ou les niveaux ε) s'avère être une tâche difficile pour produire efficacement un ensemble de solutions uniformément distribuées le long du front de Pareto. En ce qui concerne les métaheuristiques, et en particulier les algorithmes évolutionnaires multiobjectif, ils représentent une alternative aux techniques de Programmation Mathématique (PM), capables de construire une approximation du front de Pareto de manière collaborative et en une seule exécution. Le traitement des contraintes égalité constitue cependant leur principal inconvénient, surtout si ces contraintes impliquent de nombreuses variables continues, car les opérateurs génétiques employés, étant stochastiques, sont susceptibles de produire des solutions infaisables.
Par conséquent, les algorithmes hybrides représentent des techniques alternatives prometteuses, développées pour faire face aux inconvénients rencontrés par chacun des paradigme de résolution et, simultanément, pour profiter de leurs points forts. Dans ce travail, une matheuristique a été conçue avec succès pour étudier la conception optimale des chaînes d'approvisionnement en hydrogène et, en accord avec les résultats obtenus, elle surpasse les approches exactes et stochastiques pour ce problème. Bien que le théorème No Free Lunch (Wolpert and Macready, 1997) ne permette pas d'étendre directement ces conclusions à n'importe quel autre problème, cette stratégie d'hybridation semble cependant adaptée à des problèmes de production et de distribution similaires, issus du domaine des PSI.

2. *Compte tenu des avancées récentes dans le domaine des métaheuristiques, en particulier celles liées à la gestion des contraintes, ces dernières constituent-elles des techniques efficaces pour traiter des problèmes fortement contraints comme ceux liés au domaine des PSI ?*

   Dans ce travail, seules quelques méthodes de traitement des contraintes représentatives (par leur mode de fonctionnement) ont été considérées. En ce qui concerne le mécanisme de réparation basé sur le gradient des contraintes, il convient de noter qu'il a démontré son efficacité, tant en termes de temps de calcul que de qualité des solutions, pour la résolution de problèmes de référence académiques (étudiés aux chapitres 2 et 3). Cependant, cette méthode obtient des résultats médiocres lorsqu'il s' agit de traiter des applications de PSI complexes et de grande taille.
   En effet, le MOEA employé dans le Chapitre 5 comme première tentative de résolution du problème de conception de la HSC a rencontré des problèmes de convergence et n'a pas montré une robustesse satisfaisante pour ce problème difficile, contenant des contraintes égalité impliquant un grand nombre de

variables de décision. Ainsi, cette différence observée entre la qualité des résultats lors de la résolution de fonctions test académiques et des applications du monde réel suggère qu'un effort de recherche important est nécessaire pour la conception de fonctions test pertinentes, représentant un défi plus dur pour les techniques métaheuristiques et reflétant les sources de complexité apparaissant dans les problèmes d'ingénierie.

3. *Une approche matheuristique peut-elle être conçue pour la résolution du problème de conception de la HSC, de manière à améliorer les résultats obtenus par des méthodes exactes classiques, fournissant ainsi des solutions dans des temps de calcul raisonnables ?*

   En ce qui concerne les résultats satisfaisants obtenus par l'approche hybride développée et validée au chapitre 6, la réponse est clairement "oui". La stratégie proposée dans ce travail obéit à une décomposition bi-niveaux qui favorise la conception et l'utilisation d'algorithmes matheuristiques. En particulier, un MOEA peut se charger de la construction du front de Pareto et de l'aspect combinatoire du problème (en traitant les variables entières), travaillant sur un espace de recherche limité uniquement par quelques contraintes inégalité, plus faciles à satisfaire. En outre, les contraintes égalité sont gérées efficacement par un solveur de programmation linéaire, pour chaque solution partielle proposée par le MOEA. Par ailleurs, cette approche a permis la résolution d'un modèle plus réaliste pour la conception de la HSC, qui prend en compte des caractéristiques supplémentaires comme les effets d'échelle et d'apprentissage de chaque mode de production et implique alors des termes non linéaires, également contrôlés par le MOEA.

4. *Enfin, une approche matheuristique peut-elle être conçue comme un outil de résolution général, adaptée à un large éventail de problèmes issus du domaine des PSI ?*

   La conception d'une approche matheuristique dépend largement du problème étudié, ce qui signifie qu'un codage particulier est nécessaire pour chaque problème spécifique. Néanmoins, les principes employés dans ce travail pour la conception de la matheuristique peuvent être reproduits pour d'autres problèmes en PSI. Grâce au travail réalisé dans le cadre de cette thèse, quelques lignes directrices générales pour la conception de matheuristiques peuvent être énoncées. Premièrement, une analyse préliminaire des différentes caractéristiques inclues dans la représentation du système traité est nécessaire pour identifier les aspects pertinents qui expliquent la complexité du problème, par exemple, le nombre d'objectifs, les type/nombre de contraintes, les type/nombre de variables de décision, etc. Typiquement, il pourrait être approprié d'utiliser une métaheuristique pour traiter les aspects combinatoires du problème, en plus de la gestion de l'aspect multiobjectif, pour lequel différents paradigmes algorithmiques peuvent être considérés en fonction du nombre d'objectifs impliqués. En outre, étant donné que les méthodes exactes, de par leur mode de fonctionnement, traitent efficacement les problèmes de satisfaction des contraintes, les contraintes difficiles à satisfaire par les métaheuristiques (en particulier, les contraintes égalité) peuvent être gérées par une technique de PM. Finalement, compte tenu de l'efficacité des solveurs de programmation linéaire, il pourrait être utile de construire un problème de

programmation linéaire au niveau esclave et de traiter les termes non linéaires au moyen du moteur de recherche métaheuristique.

Perspectives pour les travaux futurs

Certaines perspectives de travaux futurs ont été identifiées. Elles englobent cinq axes différents, décrits comme suit :

1. Étudier la méthodologie hybride proposée pour la conception optimale des chaînes d'approvisionnement en hydrogène avec plus de deux objectifs, par exemple, la prise en compte d'objectifs sociaux ou en relation avec le risque, ou même plusieurs critères environnementaux, différents du potentiel de réchauffement global considéré dans ce travail. Outre la modélisation de ces critères, cette prise en compte d'un plus grand nombre d'objectifs n'est pas évidente à réaliser en utilisant l'approche de résolution hybride développée ici. En effet, l'algorithme SMS-EMOA devrait être remplacé par un autre MOEA pour la construction du front de Pareto au niveau supérieur (par exemple MOEA/D), car le calcul de la contribution de l'hypervolume pour plus de deux objectifs requiert des ressources de calcul considérables. En ce qui concerne le sous-problème de niveau inférieur, la fonction de scalarisation utilisée dans la formulation LP est toujours valable lorsqu'on travaille avec plus de deux objectifs, mais le choix du vecteur de poids paramétrant le problème scalaire obtenu peut devenir un problème critique (voir également le point 4 à cet égard).

2. Il semble intéressant d'appliquer l'algorithme matheuristique proposé comme outil de résolution pour l'étude du problème d'optimisation de la conception de la chaîne d'approvisionnement, du point de vue de la théorie des jeux. Dans ce contexte, chaque acteur de la chaîne logistique (détaillants, usines, distributeurs, etc.) peut être présent à différents niveaux de décision et le problème peut être formulé comme un problème d'optimisation à deux niveaux, avec différents objectifs à chaque niveau. Étant donné que l'approche proposée est basée sur une reformulation du problème au moyen d'une décomposition bi-niveaux, elle semble particulièrement adaptée pour traiter ce problème. D'ailleurs, cette approche fait actuellement l'objet de la thèse de Manuel Flores Pérez, développée au Laboratoire de Génie Chimique de Toulouse.

3. L'extension du champ d'application de l'approche matheuristique de résolution proposée ici à d'autres études de cas de PSI constitue également une perspective prometteuse. De cette manière, il pourrait être possible de proposer un outil de résolution général pour une famille de problèmes, présentant des caractéristiques similaires. Par exemple, les problèmes d'optimisation d'intégration des procédés (comme la conception d'échangeurs de chaleur ou de réseaux d'eaux usées, les éco-parcs industriels, etc.) sont typiquement d'une grande complexité, de nature multi-objectif, ils impliquent des variables mixtes entières et des termes non linéaires et non convexes, nécessitant ainsi des procédures d'optimisation globale qui peuvent se révéler très coûteuses, sans garantie d'optimalité.

4. Le quatrième axe de perspectives de travail est l'amélioration de l'algorithme matheuristique afin de considérer de manière plus appropriée la nature multi-

objectif du sous-problème de niveau inférieur. Dans cette thèse, un seul vecteur de poids (choisi aléatoirement) est utilisé pour la résolution du PL au niveau inférieur (sous-problème d'opération). Cependant, il est possible d'imaginer que plusieurs solutions, correspondant à la même solution partielle de niveau supérieur (conception de la structure HSC) mais obtenues avec des vecteurs de poids différents, participent au front de Pareto optimal. Formellement, la solution du problème de niveau inférieur est en fait un sous-front, qui peut être généré en utilisant un ensemble de vecteurs de poids dans la fonction de scalarisation. Par conséquent, l'algorithme de niveau supérieur doit être capable de traiter non pas une solution unique, mais plutôt un ensemble de solutions correspondant au sous-front susmentionné. Ainsi, la conception d'un nouvel algorithme évolutionnaire serait nécessaire pour adapter son mode de fonctionnement. Il convient de mentionner qu'une telle investigation fait l'objet de la thèse de Master de Rubén Arguedas, actuellement en cours de développement au Laboratoire de Génie Chimique de Toulouse.

5. Notons enfin que cette approche du cadre d'optimisation est suffisamment générale pour être étendue à des chaînes d'approvisionnement en énergie à base d'hydrogène plus complexes (actuellement étudiée dans le cadre de la thèse de Renato Luise), notamment power-to-gas, power-to-heat, power-to-liquids et power-to-ammonia. En effet, l'une des tâches les plus difficiles pour les modèles de systèmes énergétiques est de saisir tout le degré de variabilité et de complexité qui existe dans les systèmes énergétiques. Étant donné que de plus en plus de sources renouvelables telles que le solaire et l'éolien sont introduites pour contribuer à la décarbonisation, les systèmes sont de plus en plus distribués dans l'espace, technologiquement diversifiés et variables dans le temps. Parallèlement, les nouvelles technologies et l'interconnexion accrue permettent une plus grande interaction entre les différents secteurs énergétiques. Pour s'assurer que les modèles de systèmes énergétiques fournissent non seulement une représentation précise de ces systèmes mais qu'ils ne passent pas à côté du potentiel des nouvelles technologies telles que celles basées sur l'hydrogène, ils doivent saisir le niveau requis de détails temporels, spatiaux, technologiques et intersectoriels qui nécessitent des stratégies d'optimisation efficaces.

Nous espérons que ce travail de recherche a donné des résultats utiles dans le contexte de l'optimisation mathématique, avec une applicabilité particulière dans le domaine de Procédés et Systèmes Industriels (PSI) et un accent sur l'ingénierie des systèmes énergétiques.

Appendix A
Single-objective test problems

This appendix describes the 14 global optimization test problems considered in Chapter 2. For all problems, the global optimum solution is reported as found in the literature. Additional information related to local optima and active constraints is also given.

Example 1. Reactor network design. Proposed in Ryoo and Sahinidis (1995), this problem involves the design of a sequence of two reactors of type CSTR, where the consecutive reactions A $\rightarrow$ B $\rightarrow$ C takes place. The objective is to maximize the concentration of product B (x_4) in the exit stream. The mathematical model is as follows:

$$\begin{aligned}
\min \quad & f(\mathbf{x}) = -x_4 \\
\text{s.t.} \quad & g_1(\mathbf{x}) = x_5^{0.5} + x_6^{0.5} - 4 \leq 0 \\
& h_1(\mathbf{x}) = x_1 + k_1 x_1 x_5 - 1 = 0 \\
& h_2(\mathbf{x}) = x_2 - x_1 + k_2 x_2 x_6 = 0 \\
& h_3(\mathbf{x}) = x_3 + x_1 + k_3 x_3 x_5 - 1 = 0 \\
& h_4(\mathbf{x}) = x_4 - x_3 + x_2 - x_1 + k_4 x_4 x_6 = 0 \\
& 0 \leq x_i \leq 1, \quad i = \{1,2,3,4\} \\
& 1\text{e}-5 \leq x_i \leq 16, \quad i = \{5,6\}
\end{aligned}$$

where $k_1 = 0.09755988$, $k_2 = 0.99k_1$, $k_3 = 0.0391908$, $k_4 = 0.9k_3$. The global optimum is at $\mathbf{x}^* = [0.771462, 0.516997, 0.204234, 0.388812, 3.036504, 5.096052]$, with $f(\mathbf{x}^*) = -0.388812$. Constraint g_1 is active. This example possesses a local minimum with an objective function value that is very close to that of the global solution. This local solution is at $\mathbf{x}$ = [1, 0.393, 0, 0.3881, 0, 16] with $f = -0.3881$. Interestingly, this solution utilizes only one of the two reactors whereas the global solution makes use of both reactors.

Example 2. Flowsheeting. This problems considers the optimization of a flow sheet example of the Williams & Otto process (Biegler et al., 1997; Pintaric and Kravanja, 2006). Reactants A and B and the recycle stream enter the continuous-flow stirred-tank reactor, where the main product P is produced together with one by-product E and the waste product G, while C is an intermediate.

$$\begin{aligned}
\text{A} + \text{B} &\rightarrow \text{C} \\
\text{C} + \text{B} &\rightarrow \text{P} + \text{E} \\
\text{P} + \text{C} &\rightarrow \text{G}
\end{aligned}$$

In the decanter, component G is entirely removed from the other components. Product P is removed from the overhead of the distillation column, but some of the product is retained in the bottom due to the formation of an azeotrope. Part of the

bottom stream is purged in order to avoid accumulation of the by-product, while most of it is recycled to the reactor. The purge stream has a substantial fuel value and can be sold on the market. The optimization variables account for the reactor volume, the reaction temperature, the purge fraction and the mass flow for each component, except for component P which is equal to 2 160 kg/h. The objective is to minimize the total annual cost. The model is formulated as:

$$\min \quad f(\mathbf{x}) = \frac{1}{0.453}\Big[168x_5 + 252x_1 + 2.22\big[x_1 + x_5 + \sum_{i=6}^{8}(1-x_4)x_i + 1.1(1-x_4)x_9\big] + 84x_{10} + 60x_2\rho\Big] + 1041.6$$

s.t.

$$h_1(\mathbf{x}) = x_5 + x_6(1-x_4) - \frac{k_1 x_6 x_7 x_2 \rho}{q_3^2} - x_6 = 0$$

$$h_2(\mathbf{x}) = x_1 + x_7(1-x_4) - \frac{(k_1 x_6 + k_2 x_8) x_7 x_2 \rho}{q_3^2} - x_7 = 0$$

$$h_3(\mathbf{x}) = x_8(1-x_4) + \frac{(2k_1 x_6 x_7 - 2k_2 x_7 x_8) x_2 \rho}{q_3^2} + \frac{(-k_3 x_8 (2160 + 0.1x_9)) x_2 \rho}{q_3^2} - x_8 = 0$$

$$h_4(\mathbf{x}) = x_9(1-x_4) + \frac{2k_2 x_7 x_8 x_2 \rho}{q_3^2} - x_9 = 0$$

$$h_5(\mathbf{x}) = \frac{x_9(1-x_4)}{10} + \frac{(k_2 x_7 - 0.5k_3(2160 + 0.1x_9)) x_8 x_2 \rho}{q_3^2} - 2160 + 0.1x_9 = 0$$

$$h_6(\mathbf{x}) = \frac{1.5k_3(2160 + 0.1x_9) x_8 x_2 \rho}{q_3^2} - x_{10} = 0$$

$$1\text{e}4 \leq x_1 \leq 1.5\text{e}4$$
$$0.85 \leq x_2 \leq 10$$
$$322 \leq x_3 \leq 378$$
$$0 \leq x_4 \leq 0.99$$
$$0 \leq x_i \leq 1e5 \quad \forall i \in \{5, \ldots, 10\}$$

where

$$q_3 = x_6 + x_7 + x_8 + 1.1x_9 + x_{10} + 2160$$
$$k_1 = 5.9755\text{e}9 \cdot \exp\left(\frac{-1.2\text{e}4}{x_3}\right)$$
$$k_2 = 2.5962\text{e}12 \cdot \exp\left(\frac{-1.5\text{e}4}{x_3}\right)$$
$$k_3 = 9.6283\text{e}15 \cdot \exp\left(\frac{-2\text{e}4}{x_3}\right)$$
$$\rho = 801$$

The optimum lies at $\mathbf{x}^*$ = [10878.60, 7.90, 342.11, 0.102, 4807.37, 11122.40, 39668.61, 2874.52, 61925.59, 1101.336] with $f(\mathbf{x}^*)$ = 9490592.6.

Example 3. Process synthesis MINLP. This is a little process synthesis problem with only two decision variables. It was proposed by Kocis and Grossmann (1988), and also found in Ryoo and Sahinidis (1995):

$$\begin{aligned} \min \quad & f(\mathbf{x}) = 2x_1 + x_2 \\ \text{s.t.} \quad & g_1(\mathbf{x}) = 1.25 - x_1^2 - x_2 \leq 0 \\ & g_2(\mathbf{x}) = x_1 + x_2 - 1.6 \leq 0 \\ & 0 \leq x_1 \leq 1.6 \\ & x_2 = \{0, 1\} \end{aligned}$$

The global minimum is [0.5, 1] with $f = 2$. There is a local minimum at [1.118, 0] with $f = 2.236$. Constraint g_1 is active.

Example 4. MINLP. This example is taken from Kocis and Grossmann (1987):

$$\begin{aligned} \min \quad & f(\mathbf{x}) = 2x_1 + x_2 - x_3 \\ \text{s.t.} \quad & g_1(\mathbf{x}) = -x_1 + x_2 + x_3 \leq 0 \\ & h_1(\mathbf{x}) = x_1 - 2\exp(-x_2) = 0 \\ & 0.5 \leq x_1 \leq 1.4 \\ & 0 \leq x_2 \leq 2 \\ & x_3 = \{0, 1\} \end{aligned}$$

There is one local optimum at [0.853, 0.853, 0] with $f = 2.558$. The global minimum is $\{\mathbf{x}^*; f(\mathbf{x}^*)\}$ = {1.375, 0.375, 1; 2.124}. Constraint g_1 is active.

Example 5. MINLP. Problem taken from Floudas (1995):

$$\begin{aligned} \min \quad & f(\mathbf{x}) = -0.7x_3 + 5(x_1 - 0.5)^2 + 0.8 \\ \text{s.t.} \quad & g_1(\mathbf{x}) = -\exp(x_1 - 0.2) - x_2 \leq 0 \\ & g_2(\mathbf{x}) = x_2 + 1.1x_3 + 1 \leq 0 \\ & g_3(\mathbf{x}) = x_1 - 1.2x_3 - 0.2 \leq 0 \\ & 0.2 \leq x_1 \leq 1 \\ & -2.22554 \leq x_2 \leq -1 \\ & x_3 = \{0, 1\} \end{aligned}$$

The global minimum is at $[0.94194, -2.1, 1]$ where $f(\mathbf{x}^*) = 1.07654$. Constraints g_1 and g_2 are active.

Example 6. MINLP. Proposed in Kocis and Grossmann (1988), and also reported in Cardoso et al. (1997), Floudas et al. (1989), and Ryoo and Sahinidis (1995):

$$\begin{aligned}
\min \quad & f(\mathbf{x}) = 2x_1 + 3x_2 + 1.5x_3 + 2x_4 - 0.5x_5 \\
\text{s.t.} \quad & g_1(\mathbf{x}) = x_1 + x_3 - 1.6 \leq 0 \\
& g_2(\mathbf{x}) = 1.333x_2 + x_4 - 3 \leq 0 \\
& g_3(\mathbf{x}) = -x_3 - x_4 + x_5 \leq 0 \\
& h_1(\mathbf{x}) = x_1^2 + x_3 - 1.25 = 0 \\
& h_2(\mathbf{x}) = x_2^{1.5} + 1.5x_4 - 3 = 0 \\
& 0 \leq x_1 \leq 1.5 \\
& 0 \leq x_2 \leq 2.2 \\
& x_i = \{0, 1\}, \quad i = \{3, 4, 5\}
\end{aligned}$$

There are 2^3 different combinations of the binary variables, of these only one combination is infeasible because it violates the pure integer constraint. The global solution is $\mathbf{x}^* = [1.118, 1.3310, 0, 1, 1]$ with $f(\mathbf{x}^*) = 7.667$. Constraint g_3 is active.

Example 7. Reactor network design. This problem, taken from Kocis and Grossmann (1989) and also studied in Cardoso et al. (1997), is a two-reactor problem, where selection is to be made among two candidate reactors the one that minimizes the cost of producing a desired product. The MINLP formulation is given as:

$$\begin{aligned}
\min \quad & f(\mathbf{x}) = 7.5x_5 + 5.5x_6 + 7x_1 + 6x_2 + 5(x_3 + x_4) \\
\text{s.t.} \quad & g_1(\mathbf{x}) = x_1 - 10x_5 \leq 0 \\
& g_2(\mathbf{x}) = x_2 - 10x_6 \leq 0 \\
& g_3(\mathbf{x}) = x_3 - 20x_5 \leq 0 \\
& g_4(\mathbf{x}) = x_4 - 20x_6 \leq 0 \\
& h_1(\mathbf{x}) = x_5 + x_6 - 1 = 0 \\
& h_2(\mathbf{x}) = x_7 - 0.9x_3(1 - \exp(-0.5x_1)) = 0 \\
& h_3(\mathbf{x}) = x_8 - 0.8x_4(1 - \exp(-0.4x_2)) = 0 \\
& h_4(\mathbf{x}) = x_7 + x_8 - 10 = 0 \\
& x_i \geq 0, \quad i = \{1, 2, 3, 4, 7, 8\} \\
& x_i = \{0, 1\}, \quad i = \{5, 6\}
\end{aligned}$$

The global minimum is $\mathbf{x}^*$ = [3.514, 0, 13.428, 0, 1, 0, 10, 0.0001] with f = 99.238. Constraints g_2 and g_4 are active.

Example 8. Process synthesis MINLP. This example is taken from Yuan et al. (1989), and is also found in Cardoso et al. (1997), Floudas et al. (1989), Ryoo and Sahinidis (1995), and Yiqing et al. (2007):

$$\begin{aligned}
\min \quad & f(\mathbf{x}) = (x_4 - 1)^2 + (x_5 - 2)^2 + (x_6 - 1)^2 \\
& - \ln(x_7 + 1) + (x_1 - 1)^2 + (x_2 - 2)^2 + (x_3 - 3)^2
\end{aligned}$$

$$
\begin{aligned}
\text{s.t.} \quad & g_1(\mathbf{x}) = \sum_{i=1}^{6} x_i - 5 \leq 0 \\
& g_2(\mathbf{x}) = \sum_{i=1}^{4} x_i^2 - 5.5 \leq 0 \\
& g_3(\mathbf{x}) = x_4 + x_1 - 1.2 \leq 0 \\
& g_4(\mathbf{x}) = x_5 + x_2 - 1.8 \leq 0 \\
& g_5(\mathbf{x}) = x_6 + x_3 - 2.5 \leq 0 \\
& g_6(\mathbf{x}) = x_7 + x_1 - 1.2 \leq 0 \\
& g_7(\mathbf{x}) = x_5^2 + x_2^2 - 1.64 \leq 0 \\
& g_8(\mathbf{x}) = x_6^2 + x_3^2 - 4.25 \leq 0 \\
& g_9(\mathbf{x}) = x_5^2 + x_3^2 - 4.64 \leq 0 \\
& x_i \geq 0, \quad i = \{1, 2, 3\} \\
& x_i = \{0, 1\}, \quad i = \{4, 5, 6, 7\}
\end{aligned}
$$

The global minimum is $\{\mathbf{x}^*; f(\mathbf{x}^*)\}$ = {0.2, 0.8, 1.9079, 1, 1, 0, 1; 4.579582}. Constraints g_3, g_4, g_6, g_7 and g_9 are active.

Example 9. Planning problem. First introduced in Kocis and Grossmann (1988), this example represents a small planning problem, in which several alternatives are proposed for obtaining product C. The goal is to produce the profitable product C from B that is purchased from a market or produced from raw material A. There are also two paths to produce B from A. The problem is modeled as a MINLP:

$$
\begin{aligned}
\min \quad & f(\mathbf{x}) = 3.5x_1 + x_2 + 1.5x_3 + 7x_5 + x_6 + 1.2x_7 + 1.8x_8 - 11x_{11} \\
\text{s.t.} \quad & g_1(\mathbf{x}) = x_4 - 5x_1 \leq 0 \\
& g_2(\mathbf{x}) = x_9 - 5x_2 \leq 0 \\
& g_3(\mathbf{x}) = x_{10} - 5x_3 \leq 0 \\
& g_4(\mathbf{x}) = x_{11} - 1 \leq 0 \\
& h_1(\mathbf{x}) = x_6 - \ln(1 + x_9) = 0 \\
& h_2(\mathbf{x}) = x_7 - 1.2\ln(1 + x_{10}) = 0 \\
& h_3(\mathbf{x}) = x_{11} - 0.9x_4 = 0 \\
& h_4(\mathbf{x}) = -x_4 + \sum_{i=5}^{7} x_i = 0 \\
& h_5(\mathbf{x}) = -x_8 + x_9 + x_{10} = 0 \\
& x_i = \{0, 1\}, \quad i = \{1, 2, 3\} \\
& x_i \geq 0, \quad \forall i \\
& x_6 \leq 5 \\
& x_{11} \leq 1
\end{aligned}
$$

The model contains three binary variables and five continuous variables. The global minimum is $\mathbf{x}^*$ = [1, 0, 1, 1.11111081, 0, 0, 1.11111081, 1.5242038, 0, 1.5242038, 0.99999978] and $f(\mathbf{x}^*) = -1.9231$. Constraints g_2 and g_4 are active. There is a local

optimum at $\mathbf{x}$ = [1, 1, 1, 1.111, 0, 0.446744, 0.664156, 1.30208, 0.563058, 0.739121, 1] with $f(\mathbf{x}) = -1.41252645$.

Examples 10–14. Multi-product batch plant design. The multi-product batch plant consists of M processing stages in series where fixed amounts Q_i of N products have to be manufactured. The objective is to determine for each stage j the number of parallel units N_j and their sizes V_j and for each product i the corresponding batch sizes B_i and cycle times $T_{\mathrm{L}i}$. The problem data are the horizon time H, the size factors S_{ij} and processing times t_{ij} of product i in stage j, the required productions Q_{ij}, and appropriate cost functions α_j and β_j. The mathematical formulation of this problem is as follows (Kocis and Grossmann, 1988):

$$
\begin{aligned}
\min \quad & \sum_{j=1}^{M} \alpha_j N_j V_j^{\beta_j} \\
\text{s.t.} \quad & \sum_{i=1}^{N} \frac{Q_i T_{\mathrm{L}i}}{B_i} - H \leq 0 \\
& S_{ij} B_i - V_j \leq 0 \\
& t_{ij} - N_j T_{\mathrm{L}i} \leq 0 \\
& 1 \leq N_j \leq N_j^{\mathrm{u}} \\
& V_j^{\mathrm{l}} \leq V_j \leq V_j^{\mathrm{u}} \\
& T_{\mathrm{L}i}^{\mathrm{l}} \leq T_{\mathrm{L}i} \leq T_{\mathrm{L}i}^{\mathrm{u}} \\
& B_j^{\mathrm{l}} \leq B_j \leq B_j^{\mathrm{u}} \\
& N_j \text{ integer}
\end{aligned}
$$

The bounds N_j^{u}, V_j^{l}, V_j^{μ} are specified by the problem and appropriate bounds for $T_{\mathrm{L}i}$ and B_i can be determined as follows:

$$
\begin{aligned}
T_{\mathrm{L}i}^{\mathrm{l}} &= \max_j \frac{t_{ij}}{N_j^{\mathrm{u}}} \\
T_{\mathrm{L}i}^{\mathrm{u}} &= \max_j t_{ij} \\
B_i^{\mathrm{l}} &= \frac{Q_i}{H} T_{\mathrm{L}i}^{\mathrm{l}} \\
B_i^{\mathrm{u}} &= \min\left(Q_i, \min_j \frac{V_j^{\mathrm{u}}}{S_{ij}}\right)
\end{aligned}
$$

The number of inequality constraints for each problem depends on the number of products N and the number of processing stages M, according to $2MN + 1$. The data corresponding to these problems are presented in Table A.1. For all examples the parameters α_j, β_j and H are 250, 0.6 and 6000, respectively. For problems 10 and 11, the parameters V_j^{l} and V_j^{u} take values of 250 and 2500, respectively. For problems 12 to 14, the parameters V_j^{l} and V_j^{u} are 300 and 3000, respectively.

Table A.1: Input data for Examples 10–14.

Example	M	N	N_j^{u}	S_{ij}	t_{ij}	Q_i
10	3	2	3	$\begin{bmatrix} 2 & 3 & 4 \\ 4 & 6 & 3 \end{bmatrix}$	$\begin{bmatrix} 8 & 20 & 8 \\ 16 & 4 & 4 \end{bmatrix}$	$\begin{bmatrix} 40000 \\ 20000 \end{bmatrix}$
11	3	2	3	$\begin{bmatrix} 2 & 3 & 4 \\ 4 & 6 & 3 \end{bmatrix}$	$\begin{bmatrix} 8 & 20 & 8 \\ 16 & 4 & 4 \end{bmatrix}$	$\begin{bmatrix} 200000 \\ 100000 \end{bmatrix}$
12	6	5	4	$\begin{bmatrix} 7.9 & 2.0 & 5.2 & 4.9 & 6.1 & 4.2 \\ 0.7 & 0.8 & 0.9 & 3.4 & 2.1 & 2.5 \\ 0.7 & 2.6 & 1.6 & 3.6 & 3.2 & 2.9 \\ 4.7 & 2.3 & 1.6 & 2.7 & 1.2 & 2.5 \\ 1.2 & 3.6 & 2.4 & 4.5 & 1.6 & 2.1 \end{bmatrix}$	$\begin{bmatrix} 6.4 & 4.7 & 8.3 & 3.9 & 2.1 & 1.2 \\ 6.8 & 6.4 & 6.5 & 4.4 & 2.3 & 3.2 \\ 1.0 & 6.3 & 5.4 & 11.9 & 5.7 & 6.2 \\ 3.2 & 3.0 & 3.5 & 3.3 & 2.8 & 3.4 \\ 2.1 & 2.5 & 4.2 & 3.6 & 3.7 & 2.2 \end{bmatrix}$	$\begin{bmatrix} 250000 \\ 150000 \\ 180000 \\ 160000 \\ 120000 \end{bmatrix}$
13	7	6	4	$\begin{bmatrix} 7.9 & 2.0 & 5.2 & 4.9 & 6.1 & 4.2 & 3.6 \\ 0.7 & 0.8 & 0.9 & 3.4 & 2.1 & 2.5 & 0.6 \\ 0.7 & 2.6 & 1.6 & 3.6 & 3.2 & 2.9 & 3.8 \\ 4.7 & 2.3 & 1.6 & 2.7 & 1.2 & 2.5 & 3.5 \\ 1.2 & 3.6 & 2.4 & 4.5 & 1.6 & 2.1 & 3.6 \\ 5.2 & 3.0 & 1.8 & 4.2 & 4.0 & 2.4 & 1.6 \end{bmatrix}$	$\begin{bmatrix} 6.4 & 4.7 & 8.3 & 3.9 & 2.1 & 1.2 & 6.4 \\ 6.8 & 6.4 & 6.5 & 4.4 & 2.3 & 3.2 & 2.6 \\ 1.0 & 6.3 & 5.4 & 11.9 & 5.7 & 6.2 & 6.2 \\ 3.2 & 3.0 & 3.5 & 3.3 & 2.8 & 3.4 & 6.1 \\ 2.1 & 2.5 & 4.2 & 3.6 & 3.7 & 2.2 & 1.8 \\ 2.6 & 4.2 & 3.8 & 4.1 & 5.8 & 3.8 & 6.9 \end{bmatrix}$	$\begin{bmatrix} 250000 \\ 150000 \\ 180000 \\ 160000 \\ 120000 \\ 200000 \end{bmatrix}$
14	8	6	4	$\begin{bmatrix} 7.9 & 2.0 & 5.2 & 4.9 & 6.1 & 4.2 & 3.6 & 2.4 \\ 0.7 & 0.8 & 0.9 & 3.4 & 2.1 & 2.5 & 0.6 & 2.0 \\ 0.7 & 2.6 & 1.6 & 3.6 & 3.2 & 2.9 & 3.8 & 1.4 \\ 4.7 & 2.3 & 1.6 & 2.7 & 1.2 & 2.5 & 3.5 & 2.3 \\ 1.2 & 3.6 & 2.4 & 4.5 & 1.6 & 2.1 & 3.6 & 2.7 \\ 5.2 & 3.0 & 1.8 & 4.2 & 4.0 & 2.4 & 1.6 & 6.2 \end{bmatrix}$	$\begin{bmatrix} 6.4 & 4.7 & 8.3 & 3.9 & 2.1 & 1.2 & 6.4 & 5.2 \\ 6.8 & 6.4 & 6.5 & 4.4 & 2.3 & 3.2 & 2.6 & 8.0 \\ 1.0 & 6.3 & 5.4 & 11.9 & 5.7 & 6.2 & 6.2 & 7.1 \\ 3.2 & 3.0 & 3.5 & 3.3 & 2.8 & 3.4 & 6.1 & 8.2 \\ 2.1 & 2.5 & 4.2 & 3.6 & 3.7 & 2.2 & 1.8 & 1.4 \\ 2.6 & 4.2 & 3.8 & 4.1 & 5.8 & 3.8 & 6.9 & 4.6 \end{bmatrix}$	$\begin{bmatrix} 250000 \\ 150000 \\ 180000 \\ 160000 \\ 120000 \\ 200000 \end{bmatrix}$

Appendix B
Data for HSC instances

This appendix describes the data for the HSC instances and the related nomenclature.

B.1 Model notation

Sets/Indices

E/e	Primary energy sources
G/g	Grid squares
I/i	Product physical form
J/j	Size of production plants and storage facilities
L/l	Type of transportation modes
P/p	Plant type with different production technologies
S/s	Storage facility type with different storage facilities
T/t	Time period of the planning horizon

Parameters

A^0_{eg}	Initial average availability of primary energy sources e in grid g, unit d^{-1}.
CCF	Capital charge factor – payback period of capital investment, y.
D^T_{igt}	Total demand for product form i in grid g during time period t, kg d^{-1}.
DW_l	Driver wage of transportation mode l, \$ h^{-1}.
FE_l	Fuel economy of transportation mode l, km L^{-1}.
FP_l	Fuel price of transportation mode l, \$ L^{-1}.
GE_l	General expenses of transportation mode l, \$ L^{-1}.
$L_{lgg'}$	Average delivery distance between grids g and g' by transportation mode l, km.
LR	Learning rate – cost reduction as technology manufacturers accumulate experience, %.
LUT_l	Load/unload time of product for transportation mode l, h.
ME_l	Maintenance expenses of transportation mode l, \$ km^{-1}.
$PCap^{\min}_{pji}$	Minimum production capacity of plant type p and size j for product form i, kg d^{-1}.
$PCap^{\max}_{pji}$	Maximum production capacity of plant type p and size j for product form i, kg d^{-1}.
PCC_{pji}	Capital cost of establishing plant type p and size j producing product form i, \$.
$Q^{\min}_{il}$	Minimum flow rate of product form i by transportation mode l, kg d^{-1}.
$Q^{\max}_{il}$	Maximum flow rate of product form i by transportation mode l, kg d^{-1}.
$SCap^{\min}_{sji}$	Minimum storage capacity of storage type s and size j for product form i, kg.

$SCap_{pji}^{\max}$ Maximum storage capacity of storage type s and size j for product form i, kg.

SCC_{sji} Capital cost of establishing storage type s and size j storing product form i, \$.

SP_l Average speed of transportation mode l, km h^{-1}.

SSF Safety stock factor of primary energy sources within a grid, %.

$TCap_{il}$ Capacity of transportation mode l transporting product form i, kg $mode^{-1}$.

TMA_l Availability of transportation mode l, h d^{-1}.

TMC_{il} Cost of establishing transportation mode l transporting product form i, \$ $mode^{-1}$.

UIC_e Unit cost of importing primary energy source e from overseas, \$ $unit^{-1}$.

UPC_{pji} Unit production cost for product form i by plant type p and size j, \$ kg^{-1}.

USC_{sji} Unit storage cost for product form i at storage type s and size j, \$ kg^{-1} d^{-1}.

α Network operating period, d y^{-1}.

β Storage holding period – average number of days' worth of stock, d.

γ Rate of utilization of primary energy source e by plant type p and size j, unit resource per unit product.

Decision variables

IP_{pjigt} Investment of plants of type p, size j, producing product form i in grid g, during period t.

IS_{sjigt} Investment of storage facilities of type s, size j, for product form i in grid g, during period t.

P_{pjigt} Production rate of product form i produced by plant type p of size j in grid g, during period t, kg d^{-1}.

$Q_{ilgg't}$ Flow rate of product form i by transportation mode l between grids g and g', during period t, kg d^{-1}.

Dependent variables

A_{egt} Average availability of primary energy source e in grid g during time period t, unit d^{-1}.

D_{igt}^{L} Demand for product form i in grid g satisfied by local production during time period t, kg d^{-1}.

D_{igt}^{I} Imported demand of product form i to grid g during time period t, kg d^{-1}.

ESC Primary energy source cost, \$ d^{-1}.

FC Fuel cost, \$ d^{-1}.

FCC Facility capital cost, \$.

FOC Facility operating cost, \$ d^{-1}.

GC General cost, \$ d^{-1}.

I_{egt} Import of primary energy source e to grid g from overseas during time period t, unit d^{-1}.

L_{gt}^{ave} Average delivery distance within grid g during time period t, km.

LC Labor cost, \$ d^{-1}.
LTC_{gt} Local transportation cost within grid g during time period t, \$.
MC Maintenance cost, \$ d^{-1}.
P^T_{igt} Total production rate of product form i in grid g during time period t, kg d^{-1}.
NP_{pjigt} Number of plants of type p, size j, producing product form i in grid g, during period t.
NS_{sjigt} Number of storage facilities of type s, size j, for product form i in grid g, during period t.
NTU Number of transport units.
$QE_{egg't}$ Flow rate of primary energy source e between grids g and g' during time period t, unit d^{-1}.
S_{sjigt} Average inventory of product form i stored in storage type s and size j in grid g during time period t, kg.
S^T_{igt} Total average inventory of product form i in grid g during time period t, kg.
TCC Transportation capital cost, \$.
TDC Total daily cost of the network, \$ d^{-1}.
TOC Transportation operating cost, \$ d^{-1}.

B.2 Data instances

Table B.1: Costs associated with primary energy sources.

Primary energy source, e	Natural gas	PV-elect	Wind-elect	Hydro-elect	Nuclear elect
Unit cost of energy source, UEC_e (\$/unit)	0.12	0.53	0.05	0.05	0.05
Unit import cost of energy source, UIC_e (\$/unit)	0.012	0.005	0.005	0.005	0.005

Table B.2: Production capacities and costs of hydrogen plants.

Plant type, p	Steam methane reforming			Centralized electrolysis			Distributed electrolysis	
Plant size, j	Small	Medium	Large	Small	Medium	Large	Small	Medium
$PCap^{\min}_{pji}$(t/d)	0.3	10	200	0.3	1.05	10	0.05	0.45
$PCap^{\max}_{pji}$(t/d)	9.5	150	960	2.5	9.5	150	0.4	1
$\gamma_{\ epj}$	4.02	3.34	3.16	52.49	52.49	52.49	52.49	52.49
PCC_{pji} (\$ $\times 10^6$)	29	224	903	20.198	61	663	4.03	9.02
UPC_{pji} (\$/kg)	3.36	1.74	1.43	4.94	4.69	4.59	6.24	5.38

Table B.3: Costs and characteristics of transportation modes.

Transportation mode, l	Tanker truck
Transport unit capacity, $TCap_{il}$ (t/mode)	3.5
Fuel economy between grid, FE_l^L(km/L)	2.30
Average speed between grid, SP_l^L(km/hr)	66.8
Mode availability between grids, TMA_l^R(hr/d)	18
Load/unload time, LUT_l (hr)	2
Driver wage, DW_l (\$/hr)	14.57
Fuel price, FP_l (\$/L)	1.5
Maintenance expenses, ME_l (\$/km)	0.126
General expenses, GE_l (\$/d)	8.22
Transport mode cost, TMC_{il} (\$/mode)	500,000

Table B.4: Storage capacities and costs of liquid hydrogen storage facilities.

Storage type, s	Liquid hydrogen storage			
Storage size, j	Mini	Small	Medium	Large
Minimum storage capacity, $SCap_{sji}^{\min}$(t)	0.05	0.5	10	200
Maximum storage capacity, $SCap_{sji}^{\max}$(t)	0.45	9.5	150	540
Storage capital cost, SCC_{sji} (million \$)	0.802	5	33	122
Unit storage cost, USC_{sji} (\$/kg/d)	0.064	0.032	0.010	0.005

Table B.5: Global warming potential.

	Type	Value
GWP due to transportation (g CO2 per tonne-km)	Tankertruck	62
GWP due to storage (g CO2 eq per kg H2)	Liquid hydrogen	704
GWP due to production (g CO2 eq per kg H2)	SMR	10100
	PV-elect	6206
	Wind-elect	1034
	Hydro-elect	2068
	Nuclear-elect	3100

Table B.6: Local and regional delivery distances for 8 grid instances.

Grid, g	**01**	**02**	**03**	**04**	**05**	**06**	**07**	**08**
01	0	111.1	105.5	58.3	133.6	220.2	110.7	194
02	111.1	0	71.8	126.9	214.8	287.7	146.5	228.7
03	105.5	71.8	0	75.1	152.1	225	74.5	156.7
04	58.3	126.9	75.1	0	88	160.9	51	135.5
05	133.6	214.8	152.1	88	0	73.8	79.6	137.9
06	220.2	287.7	225	160.9	73.8	0	152.8	156.9
07	110.7	146.5	74.5	51	79.6	152.8	0	84.6
08	194	228.7	156.7	135.5	137.9	156.9	84.6	0

Table B.7: Local and regional delivery distances for 22 grid instances.

Grid, g	**01**	**02**	**03**	**04**	**05**	**06**	**07**	**08**	**09**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**
01	0	63	46	97	128	197	198	175	102	139	148	191	222	306	338	297	243	174	155	222	246	240
02	63	0	70	34	65	134	146	103	133	170	220	226	293	341	373	332	278	205	186	253	277	242
03	46	70	0	59	112	173	156	133	60	97	127	148	180	264	295	254	201	132	113	180	204	198
04	97	34	59	0	53	114	111	69	76	113	192	165	197	280	312	271	218	148	131	197	220	232
05	128	65	112	53	0	70	115	71	131	208	287	230	262	309	341	300	247	177	151	216	249	234
06	197	134	173	114	70	0	135	104	186	261	339	266	300	345	377	336	289	216	187	312	292	330
07	198	146	156	111	115	135	0	43	99	140	214	157	189	226	258	217	164	94	71	124	180	141
08	175	103	133	69	71	104	43	0	75	133	187	155	186	234	266	225	172	102	75	141	174	158
09	102	133	60	76	131	186	99	75	0	23	117	88	112	206	238	197	143	73	55	123	146	140
10	139	170	97	113	208	261	140	133	23	0	83	75	100	218	250	209	156	87	68	135	159	152
11	148	220	127	192	287	339	214	187	117	83	0	43	54	102	137	125	141	135	146	188	169	205
12	191	226	148	165	230	266	157	155	88	75	43	0	24	73	108	91	86	83	80	147	124	136
13	222	293	180	197	262	300	189	186	112	100	54	24	0	49	84	68	72	92	104	135	138	145
14	306	341	264	280	309	345	226	234	206	218	102	73	49	0	35	23	66	135	158	163	111	173
15	338	373	295	312	341	377	258	266	238	250	137	108	84	35	0	35	99	168	191	196	144	206
16	297	332	254	271	300	336	217	225	197	209	125	91	68	23	35	0	58	134	150	155	103	165
17	243	278	201	218	247	289	164	172	143	156	141	86	72	66	99	58	0	72	93	100	49	102
18	174	205	132	148	177	216	94	102	73	87	135	83	92	135	168	134	72	0	23	51	75	102
19	155	186	113	131	151	187	71	75	55	68	146	80	104	158	191	150	93	23	0	70	94	87
20	222	253	180	197	216	312	124	141	123	135	188	147	135	163	196	155	100	51	70	0	55	20
21	246	277	204	220	249	292	180	174	146	159	169	124	138	111	144	103	49	75	94	55	0	44
22	240	242	198	232	234	330	141	158	140	152	205	136	145	173	206	165	110	102	87	20	44	0

Table B.8: Global warming potential for different production technologies.

Production type	Energy source	GWP (gCO_2-eq/kgH_2)	References
SMR	Natural gas	10 100	Strømman and Hertwich (2004)
SMR w/CCUS	Natural gas	3 070	Reiter and Lindorfer (2015)
AEL/PEM electrolysis	Onshore wind	1 034	Bhandari et al. (2014)
AEL/PEM electrolysis	Solar phto-voltaic	1 400	Bhandari et al. (2014)
AEL/PEM electrolysis	Grid	3 100	Utgikar and Thiesen (2006)

Data for instance HSC08g01p

Table B.9: Hydrogen demand of each grid and time period (kg/d) for instance HSC08g01p.

Grid, g	Time period, t t_1(2050)
01	12610
02	21100
03	24770
04	17710
05	14610
06	16170
07	80620
08	10580

Table B.10: Initial availability of energy sources (unit/d) for instance HSC08g01p.

Grid, g	Primary energy source, e				
	Natural gas	PV-elect	Wind-elect	Hydro-elect	Nuclear-elect
01	0	661344	0	557061	0
02	0	674199	3626113	285325	0
03	0	423414	2917903	550197	0
04	0	430540	1813654	1112723	51210000
05	0	718353	1813654	0	0
06	0	0	0	3281233	0
07	0	493130	1230321	1914367	0
08	0	26575	0	1654163	0

Data for instance HSC08g04p

Table B.11: Hydrogen demand of each grid and time period (kg/d) for instance HSC08g04p.

Grid, g	Time period, t			
	t_1 (2020)	t_2 (2021–2030)	t_3 (2031–2040)	t_4 (2041–2050)
01	502	3780	8850	12610
02	843	6320	14750	21100
03	977	7410	17330	24770
04	709	5320	12400	17710
05	570	4420	10260	14610
06	639	4850	11310	16170
07	3221	24180	56470	80620
08	437	3150	7420	10580

Table B.12: Initial availability of energy sources (unit/d) for instance HSC08g04p.

Time period, t	Grid, g	Primary energy source, e				
		Natural gas	PV-elect	Wind-elect	Hydro-elect	Nuclear-elect
1	01	0	471278	0	557061	0
	02	0	483634	2457909	285325	0
	03	0	297382	2119665	550197	0
	04	0	304231	1058296	1112723	51210000
	05	0	526073	1058296	0	0
	06	0	0	0	3281233	0
	07	0	364574	840080	1914367	0
	08	0	26575	0	1654163	0
2	01	0	635663	0	557061	0
	02	0	477847	2922190	124022	0
	03	0	406972	2804597	550197	0
	04	0	413821	1743228	1112723	51210000
	05	0	690458	1743228	0	0
	06	0	0	0	3281233	0
	07	0	474164	1182546	1914367	0
	08	0	26575	0	1654163	0
3	01	0	648377	0	557061	0
	02	0	660980	3555013	285325	0
	03	0	415112	2860689	550197	0
	04	0	422098	1778092	1112723	51210000
	05	0	704268	1778092	0	0
	06	0	0	0	3281233	0
	07	0	483553	1206197	1914367	0
	08	0	26575	0	1654163	0
4	01	0	661344	0	557061	0
	02	0	674199	3626113	285325	0
	03	0	423414	2917903	550197	0
	04	0	430540	1813654	1112723	51210000
	05	0	718353	1813654	0	0
	06	0	0	0	3281233	0
	07	0	493130	1230321	1914367	0
	08	0	26575	0	1654163	0

Data for instance HSC08g07p

Table B.13: Hydrogen demand of each grid and time period (kg/d) for instance HSC08g07p.

Grid, g	Time period, t						
	t_1 (2020)	t_2 (2021–2025)	t_3 (2026–2030)	t_4 (2031–2035)	t_5 2036–2040)	t_6 (2041–2045)	t_7 (2045–2050)
01	502	2141	3780	6315	8850	10730	12610
02	843	3581.5	6320	10535	14750	17925	21100
03	977	4193.5	7410	12370	17330	21050	24770
04	709	3014.5	5320	8860	12400	15055	17710
05	570	2495	4420	7340	10260	12435	14610
06	639	2744.5	4850	8080	11310	13740	16170
07	3221	13701	24180	40325	56470	68545	80620
08	437	1793.5	3150	5285	7420	9000	10580

Table B.14: Initial availability of energy sources (unit/d) for instance HSC08g07p.

Time period, t	Grid, g	Primary energy source, e				
		Natural gas	PV-elect	Wind-elect	Hydro-elect	Nuclear-elect
1	01	0	471278	0	557061	0
	02	0	483634	2457909	285325	0
	03	0	297382	2119665	550197	0
	04	0	304231	1058296	1112723	51210000
	05	0	526073	1058296	0	0
	06	0	0	0	3281233	0
	07	0	364574	840080	1914367	0
	08	0	26575	0	1654163	0
2	01	0	553470	0	557060	0
	02	0	480740	2690000	204670	0
	03	0	352180	2462100	550200	0
	04	0	359030	1400800	1112700	51210000
	05	0	608270	1400800	0	0
	06	0	0	0	3281200	0
	07	0	419370	1011300	1914400	0
	08	0	26575	0	1654200	0
3	01	0	635663	0	557061	0
	02	0	477847	2922190	124022	0
	03	0	406972	2804597	550197	0
	04	0	413821	1743228	1112723	51210000
	05	0	690458	1743228	0	0
	06	0	0	0	3281233	0
	07	0	474164	1182546	1914367	0
	08	0	26575	0	1654163	0
4	01	0	642020	0	557060	0
	02	0	569410	3238600	204670	0
	03	0	411040	2832600	550200	0
	04	0	417960	1760700	1112700	51210000
	05	0	697360	1760700	0	0
	06	0	0	0	3281200	0
	07	0	478860	1194400	1914400	0
	08	0	26575	0	1654200	0

Table B.14 continued from previous page

5	01	0	648377	0	557061	0
	02	0	660980	3555013	285325	0
	03	0	415112	2860689	550197	0
	04	0	422098	1778092	1112723	51210000
	05	0	704268	1778092	0	0
	06	0	0	0	3281233	0
	07	0	483553	1206197	1914367	0
	08	0	26575	0	1654163	0
6	01	0	654860	0	557060	0
	02	0	667590	3590600	285330	0
	03	0	419260	2889300	550200	0
	04	0	426320	1795900	1112700	51210000
	05	0	711310	1795900	0	0
	06	0	0	0	3281200	0
	07	0	488340	1218300	1914400	0
	08	0	26575	0	1654200	0
7	01	0	661344	0	557061	0
	02	0	674199	3626113	285325	0
	03	0	423414	2917903	550197	0
	04	0	430540	1813654	1112723	51210000
	05	0	718353	1813654	0	0
	06	0	0	0	3281233	0
	07	0	493130	1230321	1914367	0
	08	0	26575	0	1654163	0

Data for instance HSC22g01p

Table B.15: Hydrogen demand of each grid and time period (kg/d) for instance HSC22g01p.

Grid, g	Time period, t t_1 (2050)
01	3050
02	3990
03	5570
04	4790
05	10810
06	5500
07	12820
08	11950
09	12170
10	5540
11	5210
12	6330
13	3070
14	10000
15	2850
16	3320
17	5010
18	12990
19	62620
20	4920
21	1990
22	3670

Table B.16: Initial availability of energy sources (unit/d) for instance HSC22g01p.

Grid, g	Primary energy source, e				
	Natural gas	PV-elect	Wind-elect	Hydro-elect	Nuclear-elect
1	0	211707	0	0	0
2	0	211707	0	177275	0
3	0	237930	0	379786	0
4	0	242919	906827	189923	0
5	0	219573	1069870	95402	0
6	0	211707	1649416	0	0
7	0	211707	1859149	0	0
8	0	211707	1058754	550197	0
9	0	218833	906827	92937	0
10	0	211707	906827	1019786	51210000
11	0	294939	906827	0	0
12	0	211707	906827	0	0
13	0	211707	0	0	0
14	0	0	0	0	0
15	0	0	0	2281507	0
16	0	0	0	999726	0
17	0	4726	0	1019178	0
18	0	225817	0	775129	0
19	0	262587	1230321	120060	0
20	0	26575	0	86575	0
21	0	0	0	500164	0
22	0	0	0	1067424	0

Data for instance HSC22g04p

Table B.17: Hydrogen demand of each grid and time period (kg/d) for instance HSC22g04p.

Grid, g	Time period, t			
	t_1 (2050)	t_2 (2021–2030)	t_3 (2031–2040)	t_4 (2041–2050)
1	124	910	2140	3050
2	157	1200	2800	3990
3	221	1670	3910	5570
4	196	1430	3340	4790
5	428	3230	7570	10810
6	219	1660	3840	5500
7	509	3840	8970	12820
8	468	3570	8360	11950
9	480	3660	8520	12170
10	229	1660	3880	5540
11	211	1570	3660	5210
12	243	1910	4440	6330
13	116	940	2160	3070
14	398	3000	7000	10000
15	115	850	1990	2850
16	126	1000	2320	3320
17	196	1500	3520	5010
18	518	3900	9110	12990
19	2507	18780	43840	62620
20	208	1470	3450	4920
21	93	590	1400	1990
22	136	1090	2570	3670

Table B.18: Initial availability of energy sources (unit/d) for instance HSC22g04p.

Time period, t	Grid, g	Primary energy source, e				
		Natural gas	PV-elect	Wind-elect	Hydro-elect	Nuclear-elect
1	01	0	148691	0	0	0
	02	0	148691	0	177275	0
	03	0	173896	0	379786	0
	04	0	178691	529148	189923	0
	05	0	156252	685860	95402	0
	06	0	148691	1242901	0	0
	07	0	148691	1444490	0	0
	08	0	148691	675175	550197	0
	09	0	155540	529148	92937	0
	10	0	148691	529148	1019786	51210000
	11	0	228691	529148	0	0
	12	0	148691	529148	0	0
	13	0	148691	0	0	0
	14	0	0	0	0	0
	15	0	0	0	2281507	0
	16	0	0	0	999726	0
	17	0	4726	0	1019178	0
	18	0	162253	0	775129	0

Table B.18 continued from previous page

	19	0	197595	840080	120060	0
	20	0	26575	0	86575	0
	21	0	0	0	500164	0
	22	0	0	0	1067424	0
2	01	0	203486	0	0	0
	02	0	203486	0	177275	0
	03	0	228691	0	379786	0
	04	0	211047	1028326	95402	0
	05	0	63314	308497	28620	0
	06	0	203486	1585367	0	0
	07	0	203486	1786956	0	0
	08	0	203486	1017641	550197	0
	09	0	210335	871614	92937	0
	10	0	203486	871614	1019786	51210000
	11	0	283486	871614	0	0
	12	0	203486	871614	0	0
	13	0	203486	0	0	0
	14	0	0	0	0	0
	15	0	0	0	2281507	0
	16	0	0	0	999726	0
	17	0	4726	0	1019178	0
	18	0	217048	0	775129	0
	19	0	252390	1182546	120060	0
	20	0	26575	0	86575	0
	21	0	0	0	500164	0
	22	0	0	0	1067424	0
3	01	0	207556	0	0	0
	02	0	207556	0	177275	0
	03	0	233265	0	379786	0
	04	0	238156	889046	189923	0
	05	0	215268	1048893	95402	0
	06	0	207556	1617074	0	0
	07	0	207556	1822695	0	0
	08	0	207556	1037994	550197	0
	09	0	214542	889046	92937	0
	10	0	207556	889046	1019786	51210000
	11	0	289156	889046	0	0
	12	0	207556	889046	0	0
	13	0	207556	0	0	0
	14	0	0	0	0	0
	15	0	0	0	2281507	0
	16	0	0	0	999726	0
	17	0	4726	0	1019178	0
	18	0	221389	0	775129	0
	19	0	257438	1206197	120060	0
	20	0	26575	0	86575	0
	21	0	0	0	500164	0
	22	0	0	0	1067424	0
4	01	0	211707	0	0	0
	02	0	211707	0	177275	0
	03	0	237930	0	379786	0
	04	0	242919	906827	189923	0
	05	0	219573	1069870	95402	0
	06	0	211707	1649416	0	0
	07	0	211707	1859149	0	0
	08	0	211707	1058754	550197	0
	09	0	218833	906827	92937	0

Table B.18 continued from previous page

10	0	211707	906827	1019786	51210000
11	0	294939	906827	0	0
12	0	211707	906827	0	0
13	0	211707	0	0	0
14	0	0	0	0	0
15	0	0	0	2281507	0
16	0	0	0	999726	0
17	0	4726	0	1019178	0
18	0	225817	0	775129	0
19	0	262587	1230321	120060	0
20	0	26575	0	86575	0
21	0	0	0	500164	0
22	0	0	0	1067424	0

Data for instance HSC22g07p

Table B.20: Initial availability of energy sources (unit/d) for instance HSC22g07p.

Time period, t	Grid, g	Primary energy source, e				
		Natural gas	PV-elect	Wind-elect	Hydro-elect	Nuclear-elect
1	01	0	148691	0	0	0
	02	0	148691	0	177275	0
	03	0	173896	0	379786	0
	04	0	178691	529148	189923	0
	05	0	156252	685860	95402	0
	06	0	148691	1242901	0	0
	07	0	148691	1444490	0	0
	08	0	148691	675175	550197	0
	09	0	155540	529148	92937	0
	10	0	148691	529148	1019786	51210000
	11	0	228691	529148	0	0
	12	0	148691	529148	0	0
	13	0	148691	0	0	0
	14	0	0	0	0	0
	15	0	0	0	2281507	0
	16	0	0	0	999726	0
	17	0	4726	0	1019178	0
	18	0	162253	0	775129	0
	19	0	197595	840080	120060	0
	20	0	26575	0	86575	0
	21	0	0	0	500164	0
	22	0	0	0	1067424	0
2	01	0	176088.5	0	0	0
	02	0	176088.5	0	177275	0
	03	0	201293.5	0	379786	0
	04	0	194869	778737	142662.5	0
	05	0	109783	497178.5	62011	0
	06	0	176088.5	1414134	0	0
	07	0	176088.5	1615723	0	0
	08	0	176088.5	846408	550197	0
	09	0	182937.5	700381	92937	0
	10	0	176088.5	700381	1019786	51210000
	11	0	256088.5	700381	0	0
	12	0	176088.5	700381	0	0
	13	0	176088.5	0	0	0

Table B.20 continued from previous page

	14	0	0	0	0	0
	15	0	0	0	2281507	0
	16	0	0	0	999726	0
	17	0	4726	0	1019178	0
	18	0	189650.5	0	775129	0
	19	0	224992.5	1011313	120060	0
	20	0	26575	0	86575	0
	21	0	0	0	500164	0
	22	0	0	0	1067424	0
3	01	0	203486	0	0	0
	02	0	203486	0	177275	0
	03	0	228691	0	379786	0
	04	0	211047	1028326	95402	0
	05	0	63314	308497	28620	0
	06	0	203486	1585367	0	0
	07	0	203486	1786956	0	0
	08	0	203486	1017641	550197	0
	09	0	210335	871614	92937	0
	10	0	203486	871614	1019786	51210000
	11	0	283486	871614	0	0
	12	0	203486	871614	0	0
	13	0	203486	0	0	0
	14	0	0	0	0	0
	15	0	0	0	2281507	0
	16	0	0	0	999726	0
	17	0	4726	0	1019178	0
	18	0	217048	0	775129	0
	19	0	252390	1182546	120060	0
	20	0	26575	0	86575	0
	21	0	0	0	500164	0
	22	0	0	0	1067424	0
4	01	0	205521	0	0	0
	02	0	205521	0	177275	0
	03	0	230978	0	379786	0
	04	0	224601.5	958686	142662.5	0
	05	0	139291	678695	62011	0
	06	0	205521	1601220.5	0	0
	07	0	205521	1804825.5	0	0
	08	0	205521	1027817.5	550197	0
	09	0	212438.5	880330	92937	0
	10	0	205521	880330	1019786	51210000
	11	0	286321	880330	0	0
	12	0	205521	880330	0	0
	13	0	205521	0	0	0
	14	0	0	0	0	0
	15	0	0	0	2281507	0
	16	0	0	0	999726	0
	17	0	4726	0	1019178	0
	18	0	219218.5	0	775129	0
	19	0	254914	1194371.5	120060	0
	20	0	26575	0	86575	0
	21	0	0	0	500164	0
	22	0	0	0	1067424	0
5	01	0	207556	0	0	0
	02	0	207556	0	177275	0
	03	0	233265	0	379786	0
	04	0	238156	889046	189923	0

Table B.20 continued from previous page

	05	0	215268	1048893	95402	0
	06	0	207556	1617074	0	0
	07	0	207556	1822695	0	0
	08	0	207556	1037994	550197	0
	09	0	214542	889046	92937	0
	10	0	207556	889046	1019786	51210000
	11	0	289156	889046	0	0
	12	0	207556	889046	0	0
	13	0	207556	0	0	0
	14	0	0	0	0	0
	15	0	0	0	2281507	0
	16	0	0	0	999726	0
	17	0	4726	0	1019178	0
	18	0	221389	0	775129	0
	19	0	257438	1206197	120060	0
	20	0	26575	0	86575	0
	21	0	0	0	500164	0
	22	0	0	0	1067424	0
6	01	0	209631.5	0	0	0
	02	0	209631.5	0	177275	0
	03	0	235597.5	0	379786	0
	04	0	240537.5	897936.5	189923	0
	05	0	217420.5	1059381.5	95402	0
	06	0	209631.5	1633245	0	0
	07	0	209631.5	1840922	0	0
	08	0	209631.5	1048374	550197	0
	09	0	216687.5	897936.5	92937	0
	10	0	209631.5	897936.5	1019786	51210000
	11	0	292047.5	897936.5	0	0
	12	0	209631.5	897936.5	0	0
	13	0	209631.5	0	0	0
	14	0	0	0	0	0
	15	0	0	0	2281507	0
	16	0	0	0	999726	0
	17	0	4726	0	1019178	0
	18	0	223603	0	775129	0
	19	0	260012.5	1218259	120060	0
	20	0	26575	0	86575	0
	21	0	0	0	500164	0
	22	0	0	0	1067424	0
7	01	0	211707	0	0	0
	02	0	211707	0	177275	0
	03	0	237930	0	379786	0
	04	0	242919	906827	189923	0
	05	0	219573	1069870	95402	0
	06	0	211707	1649416	0	0
	07	0	211707	1859149	0	0
	08	0	211707	1058754	550197	0
	09	0	218833	906827	92937	0
	10	0	211707	906827	1019786	51210000
	11	0	294939	906827	0	0
	12	0	211707	906827	0	0
	13	0	211707	0	0	0
	14	0	0	0	0	0
	15	0	0	0	2281507	0
	16	0	0	0	999726	0
	17	0	4726	0	1019178	0
	18	0	225817	0	775129	0

Table B.20 continued from previous page

19	0	262587	1230321	120060	0
20	0	26575	0	86575	0
21	0	0	0	500164	0
22	0	0	0	1067424	0

Table B.19: Hydrogen demand of each grid and time period (kg/d) for instance HSC22g07p.

Grid, g	Time period, t						
	t_1 (2020)	t_2 (2021–2025)	t_3 (2026–2030)	t_4 (2031–2035)	t_5 2036–2040)	t_6 (2041–2045)	t_7 (2045–2050)
1	124	517	910	1525	2140	2595	3050
2	157	678.5	1200	2000	2800	3395	3990
3	221	945.5	1670	2790	3910	4740	5570
4	196	813	1430	2385	3340	4065	4790
5	428	1829	3230	5400	7570	9190	10810
6	219	939.5	1660	2750	3840	4670	5500
7	509	2174.5	3840	6405	8970	10895	12820
8	468	2019	3570	5965	8360	10155	11950
9	480	2070	3660	6090	8520	10345	12170
10	229	944.5	1660	2770	3880	4710	5540
11	211	890.5	1570	2615	3660	4435	5210
12	243	1076.5	1910	3175	4440	5385	6330
13	116	528	940	1550	2160	2615	3070
14	398	1699	3000	5000	7000	8500	10000
15	115	482.5	850	1420	1990	2420	2850
16	126	563	1000	1660	2320	2820	3320
17	196	848	1500	2510	3520	4265	5010
18	518	2209	3900	6505	9110	11050	12990
19	2507	10643.5	18780	31310	43840	53230	62620
20	208	839	1470	2460	3450	4185	4920
21	93	341.5	590	995	1400	1695	1990
22	136	613	1090	1830	2570	3120	3670

Data instances: Nonlinear model

For steam methane reforming technology, the data for nominal design capacity, as well as for the capital and operational expenditure, scaling factor and operating capacity factor are taken from NREL (2018). This is also the case for SMR w/CCUS technology. The data is summarized in tables B.21 and B.22.

Table B.21: Data for steam reforming technology.

SMR	
Reference design capacity (kgH_2/day)	379 387
Operating capacity factor (%)	90
Power scaling factor	0.6
CAPEX in EUR/(kgH_2/day) for ref. capacity	333.73
OPEX (%CAPEX/year)	5

Table B.22: Data for steam reforming with CCUS technology.

SMR w/CCUS	
Reference design capacity (kgH_2/day)	379 387
Operating capacity factor (%)	90
Power scaling factor	0.6
CAPEX in EUR/(kgH_2/day) for ref. capacity	764.24
OPEX (%CAPEX/year)	15

Regarding the data used for alkaline electrolysis, this is presented in Table B.23 (black color indicates available data, whereas gray color refers to estimated data). For 7-period instances, the data was obtained through numerical interpolations.

Table B.23: Data for alkaline electrolysis.

AEL	2020	2030	2040	2050	References
Max. nominal power consuption (MW)	10	50	80	100	IEA, 2019
Electrical efficiency (%)	75	78	80	82	Böhm et al., 2020
Operating conditions (% nom. cap.)	20-110	5-110	5-110	5-110	E&E Consultant et al.; Götz et al.
Power scaling factor	0.69	0.69	0.69	0.69	Böhm et al., 2020
CAPEX in EUR/kW_{el} for 5 MW_{el} ref.	1097	932	733	511	Böhm et al., 2020
OPEX (%CAPEX/year)	5	5	5	5	E&E Consultant et al., 2014

The data used for proton-exchange electrolysis are presented in Table B.24, with the same color code as for Table B.23.

Table B.25 describes the data for the renewable energy sources considered for electrolysis, that is, wind onshore and solar photo-voltaic. Note that the available data for the Levelized Cost of Energy (LCOE) present a cost range and not a single value. So, in this study, the mean value for each time period was considered.

Table B.24: Data for proton exchange membrane electrolysis.

PEM	2020	2030	2040	2050	References
Max. nominal power consuption (MW)	10	50	80	100	IEA, 2019
Electrical efficiency (%)	72	76	78	80	Böhm et al., 2020
Operating conditions (% nom. cap.)	5-150	5-150	5-150	5-150	E&E Consultant et al.; Götz et al.
Power scaling factor	0.72	0.72	0.72	0.72	Böhm et al., 2020
CAPEX in EUR/kW_{el} for 5 MW_{el} ref.	1188	701	445	308	Böhm et al., 2020
OPEX (%CAPEX/year)	3	3	3	3	E&E Consultant et al., 2014

Table B.25: Data concerning different energy feedstock for electrolysis.

	2020	2030	2040	2050	References
Charge factor for wind onshore (%)	25.5	25.5	25.5	25.5	RTE, 2020
Charge factor for solar PV (%)	14.5	14.5	14.5	14.5	RTE, 2020
LCOE for wind onshore (EUR/MWh)	50-71	32-58	26-50	24-46	Haeusler et al., 2020
LCOE for solar PV (EUR/MWh)	57-71	35-47	26-36	23-32	Haeusler et al., 2020

Bibliography

Ali, Layak, Samrat L Sabat, and Siba K Udgata (2012). "Particle swarm optimisation with stochastic ranking for constrained numerical and engineering benchmark problems". In: *International Journal of Bio-Inspired Computation* 4.3, pp. 155–166 (cit. on p. 45).

Almansoori, A and A Betancourt-Torcat (2016). "Design of optimization model for a hydrogen supply chain under emission constraints - A case study of Germany". In: *Energy* 111, pp. 414–429 (cit. on p. 109).

Almansoori, A. and N. Shah (2006). "Design and operation of a future hydrogen supply chain: Snapshot model". In: *Chemical Engineering Research and Design* 84.6, pp. 423 –438. DOI: https://doi.org/10.1205/cherd.05193 (cit. on pp. 108, 117).

Almansoori, A and N Shah (2009). "Design and operation of a future hydrogen supply chain: multi-period model". In: *International Journal of Hydrogen Energy* 34.19, pp. 7883–7897 (cit. on pp. 109, 111, 123, 140).

Almansoori, A and N Shah (2012). "Design and operation of a stochastic hydrogen supply chain network under demand uncertainty". In: *International Journal of Hydrogen Energy* 37.5, pp. 3965–3977 (cit. on pp. 5, 16, 109).

Almaraz, Sofía De-León, Catherine Azzaro-Pantel, Ludovic Montastruc, Luc Pibouleau, and Oscar Baez Senties (2013). "Assessment of mono and multi-objective optimization to design a hydrogen supply chain". In: *International Journal of Hydrogen Energy* 38.33, pp. 14121 –14145. DOI: https://doi.org/10.1016/j.ijhydene.2013.07.059 (cit. on p. 99).

Almaraz, Sofia De-Leon, Catherine Azzaro-Pantel, Ludovic Montastruc, and Marianne Boix (2015). "Deployment of a hydrogen supply chain by multi-objective/multi-period optimisation at regional and national scales". In: *Chemical Engineering Research and Design* 104, pp. 11–31 (cit. on pp. 5, 16, 40, 99, 104, 109, 111, 112, 123).

Almaraz, Sofía De-León, Catherine Azzaro-Pantel, Ludovic Montastruc, and Serge Domenech (2014a). "Hydrogen supply chain optimization for deployment scenarios in the Midi-Pyrénées region, France". In: *International Journal of Hydrogen Energy* 39.23, pp. 11831–11845 (cit. on pp. 100, 123, 130, 140, 147).

Almaraz, Sofía De-León, Marianne Boix, Catherine Azzaro-Pantel, Ludovic Montastruc, and Serge Domenech (2014b). "Spatial-based approach of the hydrogen supply chain in the Midi-Pyrénées region, France". In: *Computer Aided Chemical Engineering*. Vol. 33. Elsevier, pp. 307–312 (cit. on pp. 109, 111, 112).

Arnold, Dirk V and Jeremy Porter (2015). "Towards an augmented Lagrangian constraint handling approach for the (1+1)-ES". In: *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*, pp. 249–256 (cit. on p. 41).

Asafuddoula, Md, Tapabrata Ray, Ruhul Sarker, and Khairul Alam (2012). "An adaptive constraint handling approach embedded MOEA/D". In: *2012 IEEE Congress on Evolutionary Computation*. IEEE, pp. 1–8 (cit. on pp. 65, 67).

Atamna, Asma, Anne Auger, and Nikolaus Hansen (2016). "Augmented Lagrangian constraint handling for CMA-ES—case of a single linear constraint". In: *International Conference on Parallel Problem Solving from Nature*. Springer, pp. 181–191 (cit. on p. 41).

Ayala, Helon Vicente Hultmann, Patrick Keller, Márcia de Fátima Morais, Viviana Cocco Mariani, Leandro dos Santos Coelho, and Ravipudi Venkata Rao (2016). "Design of heat exchangers using a novel multiobjective free search differential evolution paradigm". In: *Applied Thermal Engineering* 94, pp. 170–177 (cit. on p. 40).

Babu, BV and Rakesh Angira (2006). "Modified differential evolution (MDE) for optimization of non-linear chemical processes". In: *Computers & Chemical Engineering* 30.6-7, pp. 989–1002 (cit. on pp. 43, 51).

Ball, Michael O (2011). "Heuristics based on mathematical programming". In: *Surveys in Operations Research and Management Science* 16.1, pp. 21–38 (cit. on p. 6).

Bandyopadhyay, S., S. Saha, U. Maulik, and K. Deb (June 2008). "A simulated annealing-based multiobjective optimization algorithm: AMOSA". In: *IEEE Transactions on Evolutionary Computation* 12.3, pp. 269–283. DOI: 10.1109/TEVC.2007.900837 (cit. on pp. 3, 13).

Barbosa, Helio JC, Afonso CC Lemonge, and Heder S Bernardino (2015). "A critical review of adaptive penalty techniques in evolutionary computation". In: *Evolutionary Constrained Optimization*. Springer, pp. 1–27 (cit. on p. 45).

Bäck, Thomas (1996). *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford University Press (cit. on p. 30).

Beccali, M, S Brunone, P Finocchiaro, and JM Galletto (2013). "Method for size optimisation of large wind–hydrogen systems with high penetration on power grids". In: *Applied Energy* 102, pp. 534–544 (cit. on p. 143).

Bhandari, Ramchandra, Clemens A Trudewind, and Petra Zapp (2014). "Life cycle assessment of hydrogen production via electrolysis–a review". In: *Journal of Cleaner Production* 85, pp. 151–163. DOI: http://dx.doi.org/10.1016/j.jclepro.2013.07.048 (cit. on pp. 103, 183).

Biegler, Lorenz T, Ignacio E Grossmann, and Arthur W Westerberg (1997). *Systematic methods for chemical process design*. Prentice Hall, Old Tappan, NJ (United States) (cit. on p. 171).

Böhm, Hans, Andreas Zauner, Daniel C Rosenfeld, and Robert Tichler (2020). "Projecting cost development for future large-scale power-to-gas implementations by scaling effects". In: *Applied Energy* 264, p. 114780 (cit. on pp. 140–142, 196, 197).

Bosman, Peter AN and Dirk Thierens (2003). "The balance between proximity and diversity in multiobjective evolutionary algorithms". In: *IEEE Transactions on Evolutionary Computation* 7.2, pp. 174–188 (cit. on p. 69).

Bozorg-Haddad, Omid, Mohammad Solgi, and Hugo A Loaiciga (2017). *Meta-heuristic and evolutionary algorithms for engineering optimization*. Vol. 294. John Wiley & Sons (cit. on p. 41).

Brest, Janez, Sao Greiner, Borko Boskovic, Marjan Mernik, and Viljem Zumer (2006). "Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems". In: *IEEE Transactions on Evolutionary Computation* 10.6, pp. 646–657 (cit. on p. 43).

Brey, JJ (2020). "Use of hydrogen as a seasonal energy storage system to manage renewable power deployment in Spain by 2030". In: *International Journal of Hydrogen Energy* (cit. on pp. 4, 15, 98).

Brooke, Anthony, David Kendrick, Alexander Meeraus, Ramesh Raman, and U America (1998). "The general algebraic modeling system". In: *GAMS Development Corporation* 1050 (cit. on p. 108).

Câmara, Diego, Tânia Pinto-Varela, and Ana Paula Barbósa-Povoa (2019). "Multi-objective optimization approach to design and planning hydrogen supply chain under uncertainty: A Portugal study case". In: *Computer Aided Chemical Engineering*. Vol. 46. Elsevier, pp. 1309–1314 (cit. on pp. 5, 16, 109).

Campbell, Stephen L and Carl D Meyer (2009). *Generalized inverses of linear transformations*. SIAM (cit. on p. 48).

Cantú, Victor H, Antonin Ponsich, and Catherine Azzaro-Pantel (Apr. 2021). "Constraint Handling in Metaheuristics and Applications". In: ed. by Anand J Kulkarni, Efrén Mezura-Montes, Yong Wang, Amir H Gandomi, and Ganesh Krishnasamy. 1st ed. Springer Singapore. Chap. On the use of gradient-based repair method for solving constrained multiobjective optimization problems – A comparative study, pp. 119–149. DOI: 10.1007/978-981-33-6710-4 (cit. on pp. 8, 19, 63).

Cantú, Victor H, Catherine Azzaro-Pantel, and Antonin Ponsich (2020a). "Multi-objective evolutionary algorithm based on decomposition (MOEA/D) for optimal design of hydrogen supply chains". In: *Computer Aided Chemical Engineering*. Ed. by Sauro Pierucci, Flavio Manenti, Giulia Luisa Bozzano, and Davide Manca. Vol. 48. Elsevier, pp. 883–888. DOI: https://doi.org/10.1016/B978-0-12-823377-1.50148-8 (cit. on pp. 8, 19).

— (2020b). "Optimal design of hydrogen supply chains by a multiobjective evolutionary algorithm based on decomposition (MOEA/D)". In: *EasyChair ROADEF2021* (cit. on pp. 8, 19).
— (2021). "Constraint-handling techniques within differential evolution for solving process engineering problems". In: *Applied Soft Computing* 108, p. 107442. DOI: https://doi.org/10.1016/j.asoc.2021.107442 (cit. on pp. 8, 19, 39).
— (accepted). "A novel matheuristic based on bi-level optimization for the multi-objective design of hydrogen supply chains". In: *Computers & Chemical Engineering*, p. 107370. DOI: https://doi.org/10.1016/j.compchemeng.2021.107370 (cit. on pp. 8, 19, 121).
Cardoso, MF, RL Salcedo, S Feyo de Azevedo, and D Barbosa (1997). "A simulated annealing approach to the solution of MINLP problems". In: *Computers & Chemical Engineering* 21.12, pp. 1349–1364 (cit. on pp. 42, 54, 174).
Carrera Guilarte, Eduardo and Catherine Azzaro-Pantel (2020). "A methodological design framework for hybrid "power-to-methane" and "power-to-hydrogen" supply chains: application to Occitania region, France". In: *Computer Aided Chemical Engineering*. Vol. 48. Elsevier, pp. 679–684 (cit. on pp. 4, 15, 101).
Chen, Xu, Wenli Du, and Feng Qian (2016). "Solving chemical dynamic optimization problems with ranking-based differential evolution algorithms". In: *Chinese Journal of Chemical Engineering* 24.11, pp. 1600–1608 (cit. on p. 43).
Chootinan, Piya and Anthony Chen (2006). "Constraint handling in genetic algorithms using a gradient-based repair method". In: *Computers & operations research* 33.8, pp. 2263–2281 (cit. on pp. 42, 43, 48).
Coello Coello, Carlos A (2000). "Use of a self-adaptive penalty approach for engineering optimization problems". In: *Computers in Industry* 41.2, pp. 113–127 (cit. on p. 45).
— (2016). "Constraint-handling techniques used with evolutionary algorithms". In: *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference Companion*. ACM, pp. 563–587 (cit. on p. 44).
Coello Coello, Carlos A, Gary B Lamont, and David A Van Veldhuizen (2007). *Evolutionary algorithms for solving multi-objective problems*. Ed. by David E. Goldberg and John R. Koza. Second. Vol. 5. Springer (cit. on pp. 2, 12, 26).
Costa, Lino and Pedro Oliveira (2001). "Evolutionary algorithms approach to the solution of mixed integer non-linear programming problems". In: *Computers & Chemical Engineering* 25.2-3, pp. 257–266 (cit. on pp. 43, 54).
Cox, Brian, Christian Bauer, Angelica Mendoza Beltran, Detlef P van Vuuren, and Christopher L Mutel (2020). "Life cycle environmental and cost comparison of current and future passenger cars under different energy scenarios". In: *Applied Energy* 269, p. 115021 (cit. on p. 99).
Cuate, Oliver, Antonin Ponsich, Lourdes Uribe, Saúl Zapotecas-Martínez, Adriana Lara, and Oliver Schütze (2020a). "A New Hybrid Evolutionary Algorithm for the Treatment of Equality Constrained MOPs". In: *Mathematics* 8.1, p. 7 (cit. on pp. 69, 85, 91).
Cuate, Oliver, Lourdes Uribe, Adriana Lara, and Oliver Schütze (2020b). "A benchmark for equality constrained multi-objective optimization". In: *Swarm and Evolutionary Computation* 52, p. 100619. DOI: https://doi.org/10.1016/j.swevo.2019.100619 (cit. on pp. 69, 91).
Cummins Inc. (n.d.). *Electrolysis*. https://www.cummins.com/new-power/applications/about-hydrogen/electrolysis. Accessed: 2021-05-23 (cit. on p. 102).
Das, Indraneel and John E Dennis (1998). "Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems". In: *SIAM journal on optimization* 8.3, pp. 631–657 (cit. on pp. 28, 34, 69, 85).
Davies, J., F. Dolci, D. Klassek-Bajorek, R. OrtizCebolla, and E. Weidner (2020). *Current status of Chemical Energy Storage Technologies*. Tech. rep. European Commission. DOI: doi:10.2760/280873 (cit. on p. 98).
Deb, K., A. Pratap, S. Agarwal, and T. Meyarivan (Apr. 2002). "A fast and elitist multiobjective genetic algorithm: NSGA-II". In: *IEEE Transactions on Evolutionary Computation* 6.2, pp. 182–197. DOI: 10.1109/4235.996017 (cit. on pp. 3, 13, 30, 32, 33, 64).

Deb, Kalyanmoy (2000). "An efficient constraint handling method for genetic algorithms". In: *Computer Methods in Applied Mechanics and Engineering* 186.2-4, pp. 311–338 (cit. on pp. 42, 43, 46, 66).

Deb, Kalyanmoy and Ram Bhushan Agrawal (1995). "Simulated binary crossover for continuous search space". In: *Complex Systems* 9.2, pp. 115–148 (cit. on pp. 30, 31).

Deb, Kalyanmoy and Himanshu Jain (2013). "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints". In: *IEEE transactions on evolutionary computation* 18.4, pp. 577–601 (cit. on p. 36).

Deb, Kalyanmoy, Lothar Thiele, Marco Laumanns, and Eckart Zitzler (2005). "Scalable test problems for evolutionary multiobjective optimization". In: *Evolutionary Multiobjective Optimization*. Springer, pp. 105–145 (cit. on p. 91).

Dodds, P E (2015). "Economics of hydrogen production". In: *Compendium of hydrogen energy*. Elsevier, pp. 63–79 (cit. on p. 101).

Dowling, Alexander W and Lorenz T Biegler (2015). "A framework for efficient large scale equation-oriented flowsheet optimization". In: *Computers & Chemical Engineering* 72, pp. 3–20 (cit. on p. 40).

E&E Consultant, Hespul, and Solagro (2014). *Etude portant sur l'hydrogène et la méthanation comme procédé de valorisation de l'électricité excédentaire*. Tech. rep. ADEME, GRTgaz, and GrDF (cit. on pp. 142, 143, 145, 196, 197).

Eichman, Josh and Francisco Flores-Espino (2016). *California power-to-gas and power-to-hydrogen near-term business case evaluation*. Tech. rep. NREL (cit. on p. 103).

Electrolysers (n.d.). https://ucellucl.com/electrolysers/. Accessed: 2021-05-23 (cit. on p. 101).

Emmerich, Michael, Nicola Beume, and Boris Naujoks (2005). "An EMO algorithm using the hypervolume measure as selection criterion". In: *International Conference on Evolutionary Multi-Criterion Optimization*. Springer, pp. 62–76 (cit. on pp. 3, 13, 30, 36, 125).

Emmerich, Michael TM and André H Deutz (2018). "A tutorial on multiobjective optimization: fundamentals and evolutionary methods". In: *Natural Computing* 17.3, pp. 585–609. DOI: https://doi.org/10.1007/s11047-018-9685-y (cit. on pp. 30, 35).

European Environment Agency (2018). *Greenhouse Gas Emissions from Transport.* Tech. rep. European Environment Agency (cit. on p. 99).

Fan, Zhun, Yi Fang, Wenji Li, Xinye Cai, Caimin Wei, and Erik Goodman (2019a). "MOEA/D with angle-based constrained dominance principle for constrained multi-objective optimization problems". In: *Applied Soft Computing* 74, pp. 621–633 (cit. on p. 65).

Fan, Zhun, Hui Li, Caimin Wei, Wenji Li, Han Huang, Xinye Cai, and Zhaoquan Cai (2016). "An improved epsilon constraint handling method embedded in MOEA/D for constrained multi-objective optimization problems". In: *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, pp. 1–8 (cit. on p. 48).

Fan, Zhun, Wenji Li, Xinye Cai, Han Huang, Yi Fang, Yugen You, Jiajie Mo, Caimin Wei, and Erik Goodman (2017). "An improved epsilon constraint-handling method in MOEA/D for CMOPs with large infeasible regions". In: *Soft Computing*, pp. 1–20 (cit. on p. 48).

— (2019b). "An improved epsilon constraint-handling method in MOEA/D for CMOPs with large infeasible regions". In: *Soft Computing* 23.23, pp. 12491–12510 (cit. on pp. 65, 68, 69, 77).

Fan, Zhun, Wenji Li, Xinye Cai, Hui Li, Caimin Wei, Qingfu Zhang, Kalyanmoy Deb, and Erik Goodman (2019c). "Push and pull search for solving constrained multi-objective optimization problems". In: *Swarm and Evolutionary Computation* 44, pp. 665–679 (cit. on p. 65).

Fan, Zhun, Jinchao Liu, Torben Sorensen, and Pan Wang (2009). "Improved differential evolution based on stochastic ranking for robust layout synthesis of MEMS components". In: *IEEE Transactions on Industrial Electronics* 56.4, pp. 937–948 (cit. on p. 45).

Fazlollahi, Samira and François Maréchal (2013). "Multi-objective, multi-period optimization of biomass conversion technologies using evolutionary algorithms and mixed integer linear programming (MILP)". In: *Applied Thermal Engineering* 50.2. Combined Special

Issues: ECP 2011 and IMPRES 2010, pp. 1504 –1513. DOI: https://doi.org/10.1016/j.applthermaleng.2011.11.035 (cit. on pp. 110, 122).

Floudas, CA, A Aggarwal, and AR Ciric (1989). "Global optimum search for nonconvex NLP and MINLP problems". In: *Computers & Chemical Engineering* 13.10, pp. 1117–1132 (cit. on p. 174).

Floudas, Christodoulos A (1995). *Nonlinear and mixed-integer optimization: fundamentals and applications*. Oxford University Press (cit. on p. 173).

Floudas, Christodoulos A and Chrysanthos E Gounaris (2009). "A review of recent advances in global optimization". In: *Journal of Global Optimization* 45.1, pp. 3–38 (cit. on pp. 2, 12, 40).

Garcia, Daniel J and Fengqi You (Oct. 2015). "Supply chain design and optimization: Challenges and opportunities". In: *Computers & Chemical Engineering* 81, pp. 153–170. DOI: https://doi.org/10.1016/j.compchemeng.2015.03.015 (cit. on p. 99).

Götz, Manuel, Jonathan Lefebvre, Friedemann Mörs, Amy McDaniel Koch, Frank Graf, Siegfried Bajohr, Rainer Reimert, and Thomas Kolb (2016). "Renewable Power-to-Gas: A technological and economic review". In: *Renewable energy* 85, pp. 1371–1390 (cit. on pp. 141, 196, 197).

Guillén-Gosálbez, Gonzalo, Fernando D. Mele, and Ignacio E. Grossmann (2010). "A bi-criterion optimization approach for the design and planning of hydrogen supply chains for vehicle use". In: *AIChE Journal* 56.3, pp. 650–667. DOI: 10.1002/aic.12024 (cit. on pp. 5, 16, 109, 122).

Haeseldonckx, Dries and William D'haeseleer (2011). "Concrete transition issues towards a fully-fledged use of hydrogen as an energy carrier: methodology and modelling". In: *International Journal of Hydrogen Energy* 36.8, pp. 4636–4652 (cit. on p. 104).

Haeusler, Laurence, Gérard Gié, Débora Moreira, Thierry Badouard, and Morgan Crenes (2020). *Coûts des énergies renouvelables et de récupération en France*. Tech. rep. ADEME (cit. on pp. 140, 197).

Hanke-Rauschenbach, R, B Bensmann, and P Millet (2015). "Hydrogen production using high-pressure electrolyzers". In: *Compendium of Hydrogen Energy*. Elsevier, pp. 179–224 (cit. on p. 102).

Heever, Susara A van den and Ignacio E Grossmann (2003). "A strategy for the integration of production planning and reactive scheduling in the optimization of a hydrogen supply network". In: *Computers & Chemical Engineering* 27.12, pp. 1813–1839 (cit. on p. 108).

Hernández Gómez, Raquel and Carlos A Coello Coello (2015). "Improved metaheuristic based on the R2 indicator for many-objective optimization". In: *Proceedings of the 2015 annual conference on Genetic and Evolutionary Computation*, pp. 679–686 (cit. on p. 29).

Hernández-Pérez, Luis Germán, Luis Fernando Lira-Barragán, and José María Ponce-Ortega (2020). "Hybrid multiobjective optimization using deterministic and metaheuristic techniques for flowback water reusing in hydraulic fracturing processes". In: *Industrial & Engineering Chemistry Research* 59.34, pp. 15298–15308 (cit. on pp. 3, 14).

Hugo, Andre, Paul Rutter, Stratos Pistikopoulos, Angelo Amorelli, and Giorgio Zoia (2005). "Hydrogen infrastructure strategic planning using multi-objective optimization". In: *International Journal of Hydrogen Energy* 30.15, pp. 1523–1534 (cit. on pp. 5, 16, 99, 104, 108).

Hydrogen Council (Jan. 2017). *Hydrogen scaling up, A sustainable pathway for the global energy transition*. Tech. rep. McKinsey & Company (cit. on pp. 98, 99, 104, 158).

— (Jan. 2020). *Path to hydrogen competitiveness. A cost perspective*. Tech. rep. Hydrogen Council (cit. on p. 104).

IEA (2019). *The Future of Hydrogen*. Tech. rep. International Energy Agency (cit. on pp. 4, 15, 98, 100, 101, 103, 104, 141, 158, 196, 197).

IEAGHG, C (2017). "Migration in the Overburden". In: *IEAGHG Technical Reports* (cit. on pp. 99–101).

Ishibuchi, Hisao, Takefumi Fukase, Naoki Masuyama, and Yusuke Nojima (2018). "Dual-grid model of moea/d for evolutionary constrained multiobjective optimization". In:

Proceedings of the Genetic and Evolutionary Computation Conference, pp. 665–672 (cit. on p. 65).

Jan, Muhammad Asif and Rashida Adeeb Khanum (2013). "A study of two penalty-parameterless constraint handling techniques in the framework of MOEA/D". In: *Applied Soft Computing* 13.1, pp. 128–148 (cit. on pp. 64, 67).

Jan, Muhammad Asif and Qingfu Zhang (2010). "MOEA/D for constrained multiobjective optimization: Some preliminary experimental results". In: *2010 UK Workshop on Computational Intelligence (UKCI)*. IEEE, pp. 1–6 (cit. on p. 66).

Jourdan, Laetitia, Matthieu Basseur, and E-G Talbi (2009). "Hybridizing exact methods and metaheuristics: A taxonomy". In: *European Journal of Operational Research* 199.3, pp. 620–629 (cit. on pp. 3, 6, 14).

Kaiser, Nicolas M, Robert J Flassig, and Kai Sundmacher (2016). "Probabilistic reactor design in the framework of elementary process functions". In: *Computers & Chemical Engineering* 94, pp. 45–59 (cit. on p. 40).

Kaliszewski, Ignacy (1987). "A modified weighted Tchebycheff metric for multiple objective programming". In: *Computers & Operations Research* 14.4, pp. 315–323 (cit. on p. 28).

Kheawhom, Soorathep (2010). "Efficient constraint handling scheme for differential evolutionary algorithm in solving chemical engineering optimization problem". In: *Journal of Industrial and Engineering Chemistry* 16.4, pp. 620–628 (cit. on pp. 42, 43).

Kim, Jiyong and Il Moon (2008). "Strategic design of hydrogen infrastructure considering cost and safety using multiobjective optimization". In: *International Journal of Hydrogen Energy* 33.21, pp. 5887–5896 (cit. on pp. 5, 16, 109).

Knowles, Joshua and David Corne (1999). "The Pareto archived evolution strategy: A new baseline algorithm for pareto multiobjective optimisation". In: *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*. Vol. 1. IEEE, pp. 98–105 (cit. on pp. 3, 13, 32).

— (2002). "On metrics for comparing nondominated sets". In: *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600)*. Vol. 1. IEEE, pp. 711–716 (cit. on pp. 26, 37, 137).

Kocis, Gary R and Ignacio E Grossmann (1987). "Relaxation strategy for the structural optimization of process flow sheets". In: *Industrial & Engineering Chemistry Research* 26.9, pp. 1869–1880 (cit. on p. 173).

Kocis, Gary R. and Ignacio E. Grossmann (1988). "Global optimization of nonconvex mixed-integer nonlinear programming (MINLP) problems in process synthesis". In: *Industrial & Engineering Chemistry Research* 27.8, pp. 1407–1421. DOI: 10.1021/ie00080a013 (cit. on pp. 55, 173–176).

Kocis, Gary R and Ignacio E Grossmann (1989). "A modelling and decomposition strategy for the MINLP optimization of process flowsheets". In: *Computers & Chemical Engineering* 13.7, pp. 797–819 (cit. on p. 174).

Leyland, Geoffrey Basil (2002). "Multi-objective optimisation applied to industrial energy problems". PhD thesis. EPFL (cit. on p. 110).

Liberti, Leo (2008). "Introduction to global optimization". In: *Ecole Polytechnique* (cit. on pp. 2, 12, 40).

Lin, Zhenhong, Shiqi Ou, Amgad Elgowainy, Krishna Reddi, Mike Veenstra, and Laura Verduzco (2018). "A method for determining the optimal delivered hydrogen pressure for fuel cell electric vehicles". In: *Applied Energy* 216, pp. 183–194 (cit. on p. 99).

López-Flores, Francisco Javier, Luis Germán Hernández-Pérez, Luis F Lira-Barragán, Eusiel Rubio-Castro, and José M Ponce-Ortega (2021). "A hybrid metaheuristic–Deterministic optimization strategy for waste heat recovery in industrial plants". In: *Industrial & Engineering Chemistry Research* 60.9, pp. 3711–3722 (cit. on pp. 3, 14).

Luise, Renato, Annabelle Brisse, and Catherine Azzaro-Pantel (2021, accepted). "Centralised vs. decentralised production and storage: optimal design of a decarbonised hydrogen supply chain with multiple end uses". In: *Computer Aided Chemical Engineering*. Elsevier (cit. on p. 163).

Mallipeddi, Rammohan and Ponnuthurai N Suganthan (2010). "Ensemble of constraint handling techniques". In: *IEEE Transactions on Evolutionary Computation* 14.4, pp. 561–579 (cit. on p. 44).

Mavrotas, George (2009). "Effective implementation of the ε-constraint method in multi-objective mathematical programming problems". In: *Applied Mathematics and Computation* 213.2, pp. 455–465 (cit. on p. 29).

Mavrotas, George and Kostas Florios (2013). "An improved version of the augmented ε-constraint method (AUGMECON2) for finding the exact pareto set in multi-objective integer programming problems". In: *Applied Mathematics and Computation* 219.18, pp. 9652–9669 (cit. on p. 29).

Mehmeti, Andi, Athanasios Angelis-Dimakis, George Arampatzis, Stephen J McPhail, and Sergio Ulgiati (2018). "Life cycle assessment and water footprint of hydrogen production methods: from conventional to emerging technologies". In: *Environments* 5.2, p. 24 (cit. on p. 155).

Mezura-Montes, Efrén and Carlos A Coello Coello (2005). "A simple multimembered evolution strategy to solve constrained optimization problems". In: *IEEE Transactions on Evolutionary computation* 9.1, pp. 1–17 (cit. on p. 44).

— (2011). "Constraint-handling in nature-inspired numerical optimization: past, present and future". In: *Swarm and Evolutionary Computation* 1.4, pp. 173–194 (cit. on p. 44).

Michalewicz, Zbigniew and Cezary Z. Janikow (1996). "GENOCOP: A genetic algorithm for numerical optimization problems with linear constraints". In: *Communications of the ACM* 39.12es, 175–es (cit. on p. 41).

Miettinen, Kaisa (2012). *Nonlinear multiobjective optimization*. Vol. 12. Springer Science & Business Media (cit. on p. 23).

Miller, Hamish Andrew, Karel Bouzek, Jaromir Hnat, Stefan Loos, Christian Immanuel Bernäcker, Thomas Weißgärber, Lars Röntzsch, and Jochen Meier-Haack (2020). "Green hydrogen from anion exchange membrane water electrolysis: a review of recent developments in critical materials and operating conditions". In: *Sustainable Energy & Fuels* 4.5, pp. 2114–2133 (cit. on p. 102).

Muteri, Vincenzo, Maurizio Cellura, Domenico Curto, Vincenzo Franzitta, Sonia Longo, Marina Mistretta, and Maria Laura Parisi (2020). "Review on life cycle assessment of solar photovoltaic panels". In: *Energies* 13.1, p. 252 (cit. on p. 155).

Nanakorn, Pruettha and Konlakarn Meesomklin (2001). "An adaptive penalty function in genetic algorithms for structural design optimization". In: *Computers & Structures* 79.29-30, pp. 2527–2539 (cit. on p. 45).

National Academy of Engineering (2004). *The hydrogen economy: Opportunities, costs, barriers, and R&D needs*. Washington, DC: The National Academies Press. DOI: 10.17226/10922 (cit. on p. 100).

NREL (2018). *Future central hydrogen production from natural gas with and without CO2 sequestration version 3.2018*. National Renewable Energy Laboratory. URL: https://www.nrel.gov/hydrogen/h2a-production-models.html (cit. on pp. 142, 143, 196).

Ouattara, Adama (2011). "Méthodologie d'éco-conception de procédés par optimisation multiobjectif et aide à la décision multicritère". PhD thesis. INPT (cit. on p. 91).

Padhye, Nikhil, Pulkit Mittal, and Kalyanmoy Deb (2015). "Feasibility preserving constraint-handling strategies for real parameter evolutionary optimization". In: *Computational Optimization and Applications* 62.3, pp. 851–890 (cit. on p. 44).

Peng, Chaoda, Hai-Lin Liu, and Fangqing Gu (2017). "An evolutionary algorithm with directed weights for constrained multi-objective optimization". In: *Applied Soft Computing* 60, pp. 613–622 (cit. on p. 65).

Pescador-Rojas, Miriam, Raquel Hernández Gómez, Elizabeth Montero, Nicolás Rojas-Morales, María-Cristina Riff, and Carlos A Coello Coello (2017). "An overview of weighted and unconstrained scalarizing functions". In: *International Conference on Evolutionary Multi-Criterion Optimization*. Springer, pp. 499–513 (cit. on pp. 28, 29, 35, 150).

Peters, Max Stone, Klaus D Timmerhaus, Ronald Emmett West, et al. (2003). *Plant design and economics for chemical engineers*. Ed. by Eduardo D Glandt, Michael T Klein, and Thomas F Edgar. Vol. 4. McGraw-Hill New York (cit. on pp. 140, 141).

Pintaric, Z Novak and Zdravko Kravanja (2006). "Selection of the economic objective function for the optimization of process flow sheets". In: *Industrial & engineering chemistry research* 45.12, pp. 4222–4232 (cit. on pp. 91, 171).

Ponsich, Antonin, Catherine Azzaro-Pantel, Serge Domenech, and Luc Pibouleau (2007). "Mixed-integer nonlinear programming optimization strategies for batch plant design problems". In: *Industrial & Engineering Chemistry Research* 46.3, pp. 854–863 (cit. on p. 40).

Ponsich, Antonin and Carlos A Coello Coello (2011). "Differential Evolution performances for the solution of mixed-integer constrained process engineering problems". In: *Applied Soft Computing* 11.1, pp. 399–409 (cit. on p. 43).

Rangaiah, Gade Pandu (2009). *Multi-objective optimization: techniques and applications in chemical engineering*. Vol. 1. World Scientific (cit. on pp. 54, 60, 61, 69, 91).

Reiter, Gerda and Johannes Lindorfer (2015). "Global warming potential of hydrogen and methane production from renewable electricity via power-to-gas technology". In: *The International Journal of Life Cycle Assessment* 20.4, pp. 477–489 (cit. on p. 183).

RTE (Jan. 2020). *Facteur de charge et taux de couverture régionaux annuels EnR, Open Data*. Tech. rep. RTE (cit. on p. 197).

Runarsson, Thomas Philip and Xin Yao (2005). "Search biases in constrained evolutionary optimization". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 35.2, pp. 233–243 (cit. on p. 42).

Runarsson, Thomas Phillip and Xin Yao (2000). "Stochastic ranking for constrained evolutionary optimization". In: *IEEE Transactions on Evolutionary Computation* 4.3, pp. 284–294 (cit. on pp. 42, 43, 45, 67).

Ryoo, Hong S and Nikolaos V Sahinidis (1995). "Global optimization of nonconvex NLPs and MINLPs with applications in process design". In: *Computers & Chemical Engineering* 19.5, pp. 551–566 (cit. on pp. 171, 173, 174).

Sabio, Nagore, Mamdouh Gadalla, Gonzalo Guillén-Gosálbez, and Laureano Jiménez (2010). "Strategic planning with risk control of hydrogen supply chains for vehicle use under uncertainty in operating costs: a case study of Spain". In: *International Journal of Hydrogen Energy* 35.13, pp. 6836–6852 (cit. on pp. 5, 16, 109, 122).

Samanipour, Faezeh and Jasmin Jelovica (2020). "Adaptive repair method for constraint handling in multi-objective genetic algorithm based on relationship between constraints and variables". In: *Applied Soft Computing* 90, p. 106143. DOI: https://doi.org/10.1016/j.asoc.2020.106143 (cit. on p. 44).

Setak, Mostafa, Fariba Feizizadeh, Hamid Tikani, and Elham Shaker Ardakani (2019). "A bi-level stochastic optimization model for reliable supply chain in competitive environments: Hybridizing exact method and genetic algorithm". In: *Applied Mathematical Modelling* 75, pp. 310 –332. DOI: https://doi.org/10.1016/j.apm.2019.05.037 (cit. on p. 110).

Sims, Ralph, Roberto Schaeffer, F Creutzig, X Cruz-Núñez, M D'agosto, D Dimitriu, MJ Figueroa Meza, L Fulton, S Kobayashi, O Lah, et al. (2014). *Transport Climate Change 2014: Mitigation of Climate Change. Contribution of Working Group III to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change.* Tech. rep. IPCC (cit. on p. 99).

Singh, Hemant Kumar, Amitay Isaacs, Trung Thanh Nguyen, Tapabrata Ray, and Xin Yao (2009). "Performance of infeasibility driven evolutionary algorithm (IDEA) on constrained dynamic single objective optimization problems". In: *2009 IEEE Congress on Evolutionary Computation*. IEEE, pp. 3127–3134 (cit. on p. 64).

Sørensen, Bent and Giuseppe Spazzafumo (2018). *Hydrogen and fuel cells: emerging technologies and applications*. Academic Press (cit. on pp. 100, 101).

Srinivas, Mekapati and GP Rangaiah (2007). "Differential evolution with tabu list for solving nonlinear and mixed-integer nonlinear programming problems". In: *Industrial & Engineering Chemistry Research* 46.22, pp. 7126–7135 (cit. on pp. 43, 51, 54).

Storn, Rainer and Kenneth Price (1997). "Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces". In: *Journal of Global Optimization* 11.4, pp. 341–359 (cit. on p. 31).

Strømman, Anders Hammer and Edgar Hertwich (2004). "Hybrid life cycle assessment of large scale hydrogen production facilities". In: *Working Papers from Industrial Ecology Programmme* 3, pp. 1504–3681 (cit. on p. 183).

Taibi, Emanuele, Raul Miranda, Wouter Vanhoudt, Thomas Winkel, Jean-Christophe Lanoix, and Frederic Barth (Sept. 2018). *Hydrogen from renewable power: Technology outlook for the energy transition*. Tech. rep. IRENA (cit. on pp. 102–104).

Takahama, Tetsuyuki and Setsuko Sakai (2005). "Constrained optimization by ε constrained particle swarm optimizer with ε-level control". In: *Soft Computing as Transdisciplinary Science and Technology*. Ed. by Ajith Abraham, Yasuhiko Dote, Takeshi Furuhashi, Mario Köppen, Azuma Ohuchi, and Yukio Ohsawa. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 1019–1029 (cit. on pp. 42, 43, 47, 65, 68).

— (2006). "Constrained optimization by the ε constrained differential evolution with gradient-based mutation and feasible elites". In: *2006 IEEE International Conference on Evolutionary Computation*. IEEE, pp. 1–8 (cit. on pp. 42, 47, 48).

— (2010). "Constrained optimization by the ε constrained differential evolution with an archive and gradient-based mutation". In: *Evolutionary Computation (CEC), 2010 IEEE Congress on*. IEEE, pp. 1–9 (cit. on pp. 42, 48).

Tawarmalani, Mohit and Nikolaos V Sahinidis (2013). *Convexification and global optimization in continuous and mixed-integer nonlinear programming: theory, algorithms, software, and applications*. Vol. 65. Springer Science & Business Media (cit. on pp. 2, 12, 40).

Teh, Shin Yee, Kian Boon Chua, Boon Hooi Hong, Alex JW Ling, Viknesh Andiappan, Dominic CY Foo, Mimi H Hassim, and Denny KS Ng (2019). "A hybrid multi-objective optimization framework for preliminary process design based on health, safety and environmental impact". In: *Processes* 7.4, p. 200 (cit. on pp. 3, 14).

Tessema, Biruk G and Gary G Yen (2006). "A self adaptive penalty function based algorithm for constrained optimization". In: *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on*. IEEE, pp. 246–253 (cit. on p. 45).

theworldofhydrogen (n.d.). https://www.theworldofhydrogen.com/gasunie/what-is-hydrogen/. Accessed: 2021-05-23 (cit. on p. 103).

Utgikar, V and Todd Thiesen (2006). "Life cycle assessment of high temperature electrolysis for hydrogen production via nuclear energy". In: *International Journal of Hydrogen Energy* 31.7, pp. 939–944 (cit. on p. 183).

Valente, Antonio, Diego Iribarren, and Javier Dufour (2020). "Prospective carbon footprint comparison of hydrogen options". In: *Science of The Total Environment*, p. 138212 (cit. on p. 100).

Van Veldhuizen, David A and Gary B Lamont (2000). "Multiobjective evolutionary algorithms: Analyzing the state-of-the-art". In: *Evolutionary Computation* 8.2, pp. 125–147 (cit. on p. 26).

Velazquez Abad, A and PE Dodds (2017). "Production of hydrogen". In: *Encyclopedia of Sustainable Technologies, Volume 3*. DOI: http://dx.doi.org/10.1016/B978-0-12-409548-9.10117-4 (cit. on pp. 100, 102, 103).

Vesterstrom, Jakob and Rene Thomsen (2004). "A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems". In: *Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No. 04TH8753)*. Vol. 2. IEEE, pp. 1980–1987 (cit. on p. 43).

Voudouris, Vasillos T and Ignacio E Grossmann (1992). "Mixed-integer linear programming reformulations for batch process design with discrete equipment sizes". In: *Industrial & Engineering Chemistry Research* 31.5, pp. 1315–1325 (cit. on p. 40).

Woldesenbet, Yonas Gebre, Gary G Yen, and Biruk G Tessema (2009). "Constraint handling in multiobjective evolutionary optimization". In: *IEEE Transactions on Evolutionary Computation* 13.3, pp. 514–525 (cit. on p. 64).

Wolpert, David H and William G Macready (1997). "No free lunch theorems for optimization". In: *IEEE Transactions on Evolutionary Computation* 1.1, pp. 67–82 (cit. on pp. 161, 167).

Woo, Young-bin, Seolhee Cho, Jiyong Kim, and Byung Soo Kim (2016). "Optimization-based approach for strategic design and operation of a biomass-to-hydrogen supply chain". In: *International Journal of Hydrogen Energy* 41.12, pp. 5405–5418 (cit. on p. 40).

Woo, Young-Bin and Byung Soo Kim (2019). "A genetic algorithm-based matheuristic for hydrogen supply chain network problem with two transportation modes and replenishment cycles". In: *Computers & Industrial Engineering* 127, pp. 981–997 (cit. on p. 110).

Yang, Yongkuan, Jianchang Liu, and Shubin Tan (2020). "A constrained multi-objective evolutionary algorithm based on decomposition and dynamic constraint-handling mechanism". In: *Applied Soft Computing* 89, p. 106104. DOI: https://doi.org/10.1016/j.asoc.2020.106104 (cit. on pp. 44, 65).

Yang, Zhixiang, Xinye Cai, and Zhun Fan (2014). "Epsilon constrained method for constrained multiobjective optimization problems: some preliminary results". In: *Proceedings of the Companion Publication of the 2014 Annual Conference on Genetic and Evolutionary Computation*. ACM, pp. 1181–1186 (cit. on p. 48).

Yee, Terrence F and Ignacio E Grossmann (1991). "A screening and optimization approach for the retrofit of heat-exchanger networks". In: *Industrial & Engineering Chemistry Research* 30.1, pp. 146–162 (cit. on p. 40).

Yiqing, Luo, Yuan Xigang, and Liu Yongjian (2007). "An improved PSO algorithm for solving non-convex NLP/MINLP problems with equality constraints". In: *Computers & Chemical Engineering* 31.3, pp. 153–162 (cit. on pp. 43, 54, 55, 174).

Yoo, Taejong, Hyunbo Cho, and Enver Yücesan (2010). "Hybrid algorithm for discrete event simulation based supply chain optimization". In: *Expert Systems with Applications* 37.3, pp. 2354–2361 (cit. on p. 108).

Yuan, X., S. Zhang, L. Pibouleau, and S. Domenech (1989). "Une methode d'optimization nonlineaire en variables mixtes pour la conception de procedes". In: *RAIRO - Operations Research* (cit. on p. 174).

Zapotecas-Martinez, Saúl and Carlos A Coello Coello (2014). "A multi-objective evolutionary algorithm based on decomposition for constrained multi-objective optimization". In: *2014 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, pp. 429–436 (cit. on p. 65).

Zapotecas-Martínez, Saúl and Carlos A Coello Coello (2011). "A multi-objective particle swarm optimizer based on decomposition". In: *Proceedings of the 13th annual conference on Genetic and Evolutionary Computation*. ACM, pp. 69–76 (cit. on pp. 3, 13).

Zapotecas-Martínez, Saúl and Antonin Ponsich (June 2020). "Constraint handling within MOEA/D through an additional scalarizing function". In: *Proceedings of the 2020 Genetic and Evolutionary Computation Conference*, pp. 595–602. DOI: https://doi.org/10.1145/3377930.3390240 (cit. on p. 65).

Zhang, Haibo and Gade Pandu Rangaiah (2012). "An efficient constraint handling method with integrated differential evolution for numerical and engineering optimization". In: *Computers & Chemical Engineering* 37, pp. 74–88 (cit. on p. 42).

Zhang, Min, Wenjian Luo, and Xufa Wang (2008a). "Differential evolution with dynamic stochastic selection for constrained optimization". In: *Information Sciences* 178.15, pp. 3043–3074 (cit. on p. 45).

Zhang, Q. and H. Li (Dec. 2007). "MOEA/D: A multiobjective evolutionary algorithm based on decomposition". In: *IEEE Transactions on Evolutionary Computation* 11.6, pp. 712–731. DOI: 10.1109/TEVC.2007.892759 (cit. on pp. 3, 13, 30, 34, 137).

Zhang, Qingfu, Wudong Liu, and Hui Li (2009). "The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances". In: *2009 IEEE Congress on Evolutionary Computation*. IEEE, pp. 203–208 (cit. on pp. 34, 125).

Zhang, Qingfu, Aimin Zhou, Shizheng Zhao, Ponnuthurai Nagaratnam Suganthan, Wudong Liu, and Santosh Tiwari (2008b). *Multiobjective optimization test instances for the CEC 2009 special session and competition*. Tech. rep. (cit. on p. 69).

Zhu, Zhaoyou, Dongfang Xu, Xingzhen Liu, Zhen Zhang, and Yinglong Wang (2016). "Separation of acetonitrile/methanol/benzene ternary azeotrope via triple column pressure-swing distillation". In: *Separation and Purification Technology* 169, pp. 66–77 (cit. on p. 40).

Zitzler, Eckart, Kalyanmoy Deb, and Lothar Thiele (2000). "Comparison of multiobjective evolutionary algorithms: Empirical results". In: *Evolutionary Computation* 8.2, pp. 173–195 (cit. on pp. 26, 37, 85).

Zitzler, Eckart, Marco Laumanns, and Lothar Thiele (2001). "SPEA2: Improving the strength Pareto evolutionary algorithm". In: *TIK-report* 103 (cit. on pp. 3, 13, 32).

Zitzler, Eckart and Lothar Thiele (1998). "Multiobjective optimization using evolutionary algorithms—a comparative case study". In: *International Conference on Parallel Problem Solving from Nature*. Springer-Verlag, pp. 292–301 (cit. on pp. 26, 69).

Résumé — Des problèmes complexes d'optimisation apparaissent souvent dans les Procédés et Systèmes Industriels (PSI). Les algorithmes évolutionnaires multi-objectifs (MOEAs) constituent des alternatives potentielles aux méthodes d'optimisation classiques ; néanmoins, leur performance dépend fortement de la technique employée pour traiter les contraintes. L'objectif de ce travail est d'explorer l'applicabilité des avancées récentes en Calcul Évolutionnaire, afin de présenter des méthodes de solution alternatives à la communauté PSI.

Une première partie de la recherche a été consacrée à l'étude des techniques de traitement des contraintes au sein d'une métaheuristique. Les résultats obtenus sur des problèmes de référence ont indiqué une supériorité de la technique qui utilise le gradient des contraintes pour réparer les solutions infaisables.

La deuxième partie de la thèse est consacrée à la conception optimale de chaînes d'approvisionnement en hydrogène, en prenant en compte des critères économiques et environnementaux. Ce problème complexe présente des caractéristiques qui posent des difficultés aux techniques classiques et métaheuristiques. Une technique hybride (matheuristique) développée à cet effet a démontré son efficacité pour générer une approximation du front de Pareto optimal. En outre, cette approche a permis d'envisager une représentation plus réaliste de la chaine d'approvisionnement en hydrogène, prenant en compte les aspects technologiques, spatiaux et temporels de sa conception.

Mots clés : Algorithmes évolutionnaires multi-objectifs, techniques de gestion des contraintes, chaînes d'approvisionnement en hydrogène, optimisation bi-niveau, matheuristique.

Abstract — Complex optimization problems are ubiquitous in process systems engineering (PSE). Multiobjective evolutionary algorithms (MOEAs) constitute potential alternatives to classical optimization methods. Nevertheless, their performance strongly depends on the technique employed for handling constraints. The objective of this work is to explore the applicability of recent advances in Evolutionary Computation to complex optimization problems, to present alternative solution methods to the PSE community.

A first part of the research was devoted to the study of constraint-handling techniques within metaheuristics. The obtained results obtained on test functions indicated the superiority of the technique using the constraint gradient to repair infeasible solutions.

The second part of the thesis is devoted to the optimal design of sustainable hydrogen supply chains (HSCs), considering both economic and environmental criteria. This complex problem present features that entail difficulties to both classical and metaheuristic techniques. A hybrid technique (matheuristic) developed in this purpose demonstrated its efficacy for generating an approximation of the Pareto front. Besides, this approach allowed considering a more realistic representation of the HSC, accounting for technological, spatial and temporal aspects of its design.

Keywords: Multiobjective evolutionary algorithms, constraint-handling techniques, hydrogen supply chains, bilevel optimization, matheuristic.

Laboratoire de Génie Chimique, UMR 5503 CNRS/INP/UPS
4 Allée Emile Monso, 31030 Toulouse