N° d'ordre NNT : 2021LYSEI056

Thèse de doctorat de l'Université de Lyon
opérée au sein de
L'INSA Lyon

Ecole Doctorale N° 512
Mathématiques et Informatique (InfoMaths)

Spécialité / discipline de doctorat : Informatique

Soutenu publiquement le 04/10/2021 par
Alexandre Millot

Exceptional Model Mining meets Multi-Objective Optimization: Application to Plant Growth Recipes in Controlled Environments

---

Devant le jury composé de :

Bruno Crémilleux	Professeur, Université de Caen	Rapporteur
Dino Ienco	Chargé de recherche HDR, INRAE Montpellier	Rapporteur
Bart Goethals	Professeur, Université d'Anvers	Examinateur
Thomas Guyet	Maître de Conférences HDR, AgroCampus Ouest	Examinateur
Céline Robardet	Professeure, INSA-Lyon	Examinatrice
Céline Rouveirol	Professeure, Université Paris 13	Examinatrice
Jean-François Boulicaut	Professeur, INSA-Lyon	Directeur de thèse
Rémy Cazabet	Maître de Conférences, Université Claude Bernard Lyon 1	Co-directeur de thèse

Département FEDORA – INSA Lyon - Ecoles Doctorales

SIGLE	ECOLE DOCTORALE	NOM ET COORDONNEES DU RESPONSABLE
CHIMIE	**CHIMIE DE LYON** **https://www.edchimie-lyon.fr** Sec. : Renée EL MELHEM Bât. Blaise PASCAL, 3e étage secretariat@edchimie-lyon.fr	**M. Stéphane DANIELE** C2P2-CPE LYON-UMR 5265 Bâtiment F308, BP 2077 43 Boulevard du 11 novembre 1918 69616 Villeurbanne **directeur@edchimie-lyon.fr**
E.E.A.	**ÉLECTRONIQUE, ÉLECTROTECHNIQUE, AUTOMATIQUE** **https://edeea.universite-lyon.fr** Sec. : Stéphanie CAUVIN Bâtiment Direction INSA Lyon Tél : 04.72.43.71.70 secretariat.edeea@insa-lyon.fr	**M. Philippe DELACHARTRE** INSA LYON Laboratoire CREATIS Bâtiment Blaise Pascal, 7 avenue Jean Capelle 69621 Villeurbanne CEDEX Tél : 04.72.43.88.63 **philippe.delachartre@insa-lyon.fr**
E2M2	**ÉVOLUTION, ÉCOSYSTÈME, MICROBIOLOGIE, MODÉLISATION** **http://e2m2.universite-lyon.fr** Sec. : Sylvie ROBERJOT Bât. Atrium, UCB Lyon 1 Tél : 04.72.44.83.62 secretariat.e2m2@univ-lyon1.fr	**M. Philippe NORMAND** Université Claude Bernard Lyon 1 UMR 5557 Lab. d'Ecologie Microbienne Bâtiment Mendel 43, boulevard du 11 Novembre 1918 69 622 Villeurbanne CEDEX **philippe.normand@univ-lyon1.fr**
EDISS	**INTERDISCIPLINAIRE SCIENCES-SANTÉ** **http://ediss.universite-lyon.fr** Sec. : Sylvie ROBERJOT Bât. Atrium, UCB Lyon 1 Tél : 04.72.44.83.62 secretariat.ediss@univ-lyon1.fr	**Mme Sylvie RICARD-BLUM** Institut de Chimie et Biochimie Moléculaires et Supramoléculaires (ICBMS) - UMR 5246 CNRS - Université Lyon 1 Bâtiment Raulin - 2ème étage Nord 43 Boulevard du 11 novembre 1918 69622 Villeurbanne Cedex Tél : +33(0)4 72 44 82 32 **sylvie.ricard-blum@univ-lyon1.fr**
INFOMATHS	**INFORMATIQUE ET MATHÉMATIQUES** **http://edinfomaths.universite-lyon.fr** Sec. : Renée EL MELHEM Bât. Blaise PASCAL, 3e étage Tél : 04.72.43.80.46 infomaths@univ-lyon1.fr	**M. Hamamache KHEDDOUCI** Université Claude Bernard Lyon 1 Bât. Nautibus 43, Boulevard du 11 novembre 1918 69 622 Villeurbanne Cedex France Tél : 04.72.44.83.69 **hamamache.kheddouci@univ-lyon1.fr**
Matériaux	**MATÉRIAUX DE LYON** **http://ed34.universite-lyon.fr** Sec. : Yann DE ORDENANA Tél : 04.72.18.62.44 yann.de-ordenana@ec-lyon.fr	**M. Stéphane BENAYOUN** Ecole Centrale de Lyon Laboratoire LTDS 36 avenue Guy de Collongue 69134 Ecully CEDEX Tél : 04.72.18.64.37 **stephane.benayoun@ec-lyon.fr**
MEGA	**MÉCANIQUE, ÉNERGÉTIQUE, GÉNIE CIVIL, ACOUSTIQUE** **http://edmega.universite-lyon.fr** Sec. : Stéphanie CAUVIN Tél : 04.72.43.71.70 Bâtiment Direction INSA Lyon mega@insa-lyon.fr	**M. Jocelyn BONJOUR** INSA Lyon Laboratoire CETHIL Bâtiment Sadi-Carnot 9, rue de la Physique 69621 Villeurbanne CEDEX **jocelyn.bonjour@insa-lyon.fr**
ScSo	**ScSo*** **https://edsciencessociales.universite-lyon.fr** Sec. : Mélina FAVETON INSA : J.Y. TOUSSAINT Tél : 04.78.69.77.79 melina.faveton@univ-lyon2.fr	**M. Christian MONTES** Université Lumière Lyon 2 86 Rue Pasteur 69365 Lyon CEDEX 07 **christian.montes@univ-lyon2.fr**

*ScSo : Histoire, Géographie, Aménagement, Urbanisme, Archéologie, Science politique, Sociologie, Anthropologie

Abstract

In today's society, information is becoming ever more pervasive. With the advent of the digital age, collecting and storing these near-infinite quantities of data is becoming increasingly easier. In this context, designing new Pattern Discovery methods, that allow for the semi-automatic discovery of relevant information and knowledge, is crucial. We consider data made of a set of descriptive attributes, where one or several of these attributes can be considered as target label(s). When a unique target label is considered, the Subgroup Discovery task aims at discovering subsets of objects – subgroups – whose target label distribution significantly deviates from that of the overall data. Exceptional Model Mining is a generalization of Subgroup Discovery. It is a recent framework that enables the discovery of significant local deviations in complex interactions between several target labels. In a world where everything has to be optimized, Multi-objective Optimization methods, which find the optimal trade-offs between numerous competing objectives, are of the essence. Although these research fields have given an extensive literature, their cross-fertilization has been considered only sparsely.

Given collected data about a process of interest, we investigate the design of methods for the discovery of relevant parameter values driving the its optimization. Our first contribution is OSMIND, a Subgroup Discovery algorithm that returns an optimal pattern in purely numerical data. OSMIND leverages advanced techniques for search space reduction that guarantee the optimality of the discovery. Our second contribution consists of a generic iterative framework that leverages the actionability of Subgroup Discovery to solve optimization problems. Our third and main contribution is Exceptional Pareto Front Mining, a new class of models for Exceptional Model Mining that involves cross-fertilization between Pattern Discovery and Multi-objective Optimization. In-depth empirical studies have been carried out on each contribution to illustrate their relevance. Our methods are generic and can be applied to many application domains.

To assess the actionability of our contributions in real life, we consider the problem of plant growth recipe optimization in controlled environments such as urban farms, the application scenario that has motivated our work. It is an intrinsic Multi-objective Optimization problem. We want to apply our pattern discovery methods to discover parameter values that lead to an optimized growth. Indeed, finding optimal settings could have tremendous repercussions on the profitability of urban farms. On synthetic and real-life data, we show that our methods allow for the discovery of parameter values that optimize the yield-cost trade-off of growth recipes.

Keywords: Subgroup Discovery, Exceptional Model Mining, Multi-objective Optimization, Urban Farms, Plant Growth Recipes

Résumé

Dans la société actuelle, l'information devient de plus en plus pervasive. Avec l'avènement de l'ère du numérique, collecter et stocker ces quantités presque infinies d'informations devient de plus en plus accessible. Dans ce contexte, la conception de méthodes de découverte de motifs permettant la découverte semi-automatique d'informations pertinentes ou de connaissances est cruciale. Nous considérons des données mettant en jeu un ensemble d'attributs descriptifs, avec un ou plusieurs de ces attributs qui peut (peuvent) être considéré(s) comme variable(s) cible(s). Quand on a un seul attribut cible, la découverte de sous-groupes vise à découvrir des sous-ensembles d'objets – des sous-groupes – dont la distribution de l'étiquette cible dévie significativement de celle de l'ensemble des données. La fouille de modèles exceptionnels est une généralisation de la découverte de sous-groupes. C'est un cadre récent permettant la découverte de déviations locales significatives dans des interactions complexes entre plusieurs variables cibles. Dans un monde où tout doit être optimisé, les méthodes d'optimisation multi-objectifs, qui trouvent les compromis optimaux entre plusieurs variables concurrentes, sont essentielles. Bien que ces différents domaines de recherche possèdent une littérature riche, leur fertilisation croisée n'a été que peu étudiée.

Avec la disponibilité de données collectées sur un processus d'intérêt, nous nous intéressons à la conception de méthodes permettant la découverte de valeurs de paramètres pertinentes pour son optimisation. Notre première contribution est OSMIND, un algorithme de découverte de sous-groupes qui retourne un motif optimal dans des données purement numériques. OSMIND exploite des techniques avancées de réduction de l'espace de recherche garantissant l'optimalité de la découverte. Notre seconde contribution consiste en un framework itératif générique qui met à profit l'exploitabilité de la découverte de sous-groupes pour résoudre des problèmes d'optimisation. Notre troisième et principale contribution est la fouille de frontières de Pareto exceptionnelles, une nouvelle classe de modèles pour la fouille de modèles exceptionnels, qui implique une fertilisation croisée entre la découverte de motifs et l'optimisation multi-objectifs. La pertinence de chacune de nos contributions a été confirmée à travers des études empiriques approfondies. Nos méthodes sont génériques et peuvent être utilisées dans de nombreux domaines d'application.

Pour évaluer l'exploitabilité de nos contributions en situation réelle, nous considérons le problème d'optimisation de recettes de pousse de plantes en environnements contrôlés tels que les fermes urbaines, le scénario d'application qui a motivé nos travaux. Améliorer la pousse des plantes est un problème intrinsèquement multi-objectifs. Nous souhaitons appliquer nos méthodes de découverte de motifs pour découvrir les valeurs de paramètres menant à une pousse optimisée. En effet, découvrir ces réglages optimaux pourrait avoir des répercussions importantes sur la rentabilité des fermes urbaines. À partir de données synthétiques et réelles, nous démontrons que nos méthodes permettent la découverte de valeurs de paramètres optimisant le compromis rendement/coûts de recettes de pousses.

Mots clés: Découverte de Sous-Groupes, Fouille de Modèles Exceptionnels, Optimisation Multi-objectifs, Fermes Urbaines, Recettes de Pousse de Plantes

Remerciements

Je tiens à remercier Dino Ienco, Chargé de Recherche HDR à l'INRAE Montpellier et Bruno Crémilleux, Professeur à l'Université de Caen, d'avoir accepté le rôle de rapporteur de ma thèse, ainsi que pour leur travail de lecture approfondi.

Je remercie également Céline Rouveirol, Professeur à l'Université Paris 13, Bart Goethals, Professeur à l'Université d'Anvers, Thomas Guyet, Maître de Conférences HDR à AgroCampus Ouest, et Céline Robardet, Professeur à l'INSA de Lyon, d'avoir accepté de participer au jury de ma thèse.

Je remercie l'entreprise INSAVALOR, responsable du financement de ma thèse et m'ayant suivi et appuyé au cours des trois dernières années. Je remercie les différents acteurs des entreprises FUL et Atos avec qui j'ai pu intéragir tout au long de ma thèse, et plus particulièrement Lauriane Charles et Paul Agnus pour l'aide apportée concernant l'application de mes travaux de recherche sur des cas d'études concrets.

Je tiens à remercier mes encadrants de thèse, Jean-François Boulicaut, Professeur à L'INSA de Lyon, et Rémy Cazabet, Maître de Conférences à l'Université Claude Bernard Lyon 1, qui m'ont guidé et offert leur savoir pendant ces trois années de thèse. Je remercie également les doctorants de l'équipe DM2L Romain et Aimene pour leur aide précieuse dans des moments de doute.

Merci également à mes amis proches, PqiNNN, Pip et Alex, dont le soutien et l'humour quotidiens auront rendu ces trois années beaucoup plus agréables.

Un grand merci bien évidemment à ma famille qui m'a toujours supporté dans mes projets.

Enfin, le plus important, je remercie du fond du coeur ma femme, Enza, pour son soutien sans faille et son amour inconditionnel, je n'en serais pas là aujourd'hui sans elle.

Life is a battle between trying to find more of yourself knowing that the real you is afraid, likes comfort, likes to be patted on the back, and doing what you need to do to get better.
David Goggins

Table of contents

Chapter 1

Introduction

1.1 Context

This thesis was completed thanks to a public grant funded by the French Single Inter-Ministry Fund (FUI AAP 24). The project, titled Digital Urban Farming 4.0 (DUF 4.0) was conducted as a collaboration between Atos, an IT services and consulting company, Ferme Urbaine Lyonnaise (FUL, Lyon Urban Farm) a startup company that is specialized in the design and selling of urban farms, and the LIRIS at INSA Lyon. The main objective of the project was to build one of the first urban and fully digital farms.
In this project, each participant was to bring its own specialized skill set and technical knowledge to the table. Atos was selected to lead the overall project and ensure its progression and completion according to the original plan. As an international giant of IT services, they were in charge of developing everything related to the digitalization of the urban farm prototype, including data access, data retrieval, and IT support. FUL, as the urban farm specialist had a central place in the project, and was supposed to design and provide a fully functional and automated prototype, but also deep knowledge of the inner workings of an urban farm, and everything related to expert agronomic knowledge. For its part, the LIRIS was tasked with developing innovative data science and artificial intelligence solutions to optimize diverse processes, like, e.g., plant growth or maintenance.
At the beginning of my PhD, back in November 2018, the urban farm of FUL was very much still a prototype, where only the growth of plants in a fully controlled environment was completed. Everything else – e.g., farm automation, data collection – was still under construction, and although several data science related ideas had been identified to optimize the future digitalized farm, the exact problems that this thesis would try to solve were still unclear. To pinpoint which problems would be most relevant and would have the biggest impact on the future of the farms if solved, we had to proceed to a deep dive into vertical urban farming.
Nowadays, conventional farming methods have to face many tough challenges like, for instance, soil erosion and groundwater depletion. The concept of vertical urban farms (see, e.g., AeroFarms, Infarm, Bowery Farming[1]) can be part of a solution. These farms allow for a significant reduction in water consumption while being able to optimize both the quantity and quality of plants. In their current form, vertical urban farms have to face a major

[1] `https://aerofarms.com/`, `https://infarm.com/`,`https://boweryfarming.com/`.

problem: operating and infrastructure costs keep them from being profitable at large scale in almost all existing cases. In this context, the development of computer-based methods allowing the optimization of urban farming processes would be a big step toward urban farms becoming successful.
Urban farms are able to produce large amounts of data thanks to numerous sensors, that can be stored locally or in the cloud such that various artificial intelligence and data mining methods can be used. New insights about the plant growth process itself but also other new services could therefore theoretically be provided to farm owners. Several scientific problems that could potentially be solved using computed-based methods had been identified. Among them, we found predictive maintenance, anomaly detection, the detection of interesting events, and plant growth optimization. From this list, it seemed like optimizing plant growth, if solved, would have the biggest impact on profitability. Furthermore, the scientific challenges behind building methods to optimize plant growth were attractive to me and to the DM2L team of the LIRIS, which I was a part of. We therefore decided to put our focus on the optimization of plant growth recipes in controlled environments.
In controlled environments such as vertical farms, the number of parameters influencing plant growth can be relatively important (e.g., temperature, hygrometry, water pH level, nutrient concentration, LED lighting intensity, CO_2 concentration). These parameters can all be supervised from the moment the plants are planted up to the day of harvest. Experts can specify a priori the expected values for these descriptive attributes, following what we will now call *plant growth recipes*. There are numerous ways of measuring the relevance of the harvested crops (e.g., cost, yield, size, flavor, or chemical properties). In other terms, we can retrieve several targets that can be used to evaluate the value of a given crop. In general, for a given type of plant, some expert knowledge exists regarding the sub-systems (e.g., to model the impact of nutrient on growth, the effect of LED lighting on photosynthesis, the energy consumption w.r.t. the temperature instruction) but we are far from a global understanding of the interactions between the various underlying phenomena. In other words, setting the optimal instructions for the diverse set of parameters given an optimization task remains an open problem.
Can we learn from available recipe records to suggest new ones that should provide better results w.r.t. the selected target attributes? Furthermore, as the urban farm of FUL was still in the prototype phase, we were aware that real-life growth data would be unavailable for most of the doctorate. Therefore, can we also design innovative solutions to assess the relevance of our developed methods, such that they can be directly implemented into working urban farms once the time comes?
We decided to address the issue by means of pattern discovery techniques, a domain of predilection of DM2L.
It is important to note that while the focus of the research was put on optimizing urban farms, we decided to develop generic methods that could be applied to other projects related to the so-called Industry 4.0 area.

1.2 Pattern Discovery and Multi-Objective Optimization

As society becomes ever more digitalized, giant improvements in computing power and data storage have been made. While only a few years ago data availability used to be a hindrance

to the development and validation of new methods, researchers now have at hand large masses of information to exploit. Extracting only relevant and actionable patterns from such data using computer-based techniques is a tangible challenge that has the potential to yield enormous benefits for many entities.

The need to discover interesting patterns in data is nothing new. Association Rule Mining, which allows for the discovery of rules describing outstanding relationships between several attributes, was introduced almost 30 years ago (Agrawal et al., 1993). Let us now imagine a dataset composed of several descriptive attributes about the eating habits of a large number of people and whether they are overweight or not. With this data at hand, we could discover rules such as:

$$soda = \text{“}every\_day\text{”} \wedge junk\_food = \text{“}often\text{”} \implies overweight = True$$

and

$$soda = \text{“}every\_day\text{”} \implies junk\_food = \text{“}often\text{”}$$

From these rules, we can extract relevant information: (1) people who drink soda every day and eat junk food often tend to be overweight, (2) people who drink soda every day are likely to often eat junk food.

Nowadays, a large part of data that is available can be defined as *labeled data*, i.e., data made of objects defined by a set of descriptive attributes and a target label. Discovering interesting knowledge in such data – known as Subgroup Discovery (SD) – is an important pattern discovery task that has captured the attention of researchers for 25 years. SD aims at discovering subsets of objects in data – subgroups – whose target label distribution significantly deviates from that of the overall data. In SD, the search space of subgroups consists of a large set of subgroup descriptions, and each description is made of a set of constraints on some attributes of the dataset.

Association Rule Mining and SD are closely related. In Association Rule Mining several attributes can exist in both the antecedent and consequent of the rules, and two given rules can have different attributes as consequent. SD, however, restricts the discovery to rules that discriminate a predefined target attribute. For example, given the previous example, the target attribute could be the binary label “overweight”. Then, we could be interested in discovering subsets of the population (subgroups) that are more likely to be overweight than the norm (the overall dataset).

The global problem that is tackled in this thesis regards the development of innovative Pattern Discovery methods to help solving optimization problems when typical existing algorithms cannot be applied. We consider a setting where there is a need to discover optimal values of descriptive attributes that lead to the optimization of one or several numerical targets. In this context, the use of SD is relevant.

For example, let us imagine a scenario where we have at hand several attributes describing isolation properties of houses and a target label that defines, for each house, its energy consumption. In this setting, discovering a subgroup of houses that optimize energy consumption is extremely relevant. Indeed, the description of the subgroup would detail interesting isolation properties that lead to reduced energy consumption. The information provided by the description of the subgroup is therefore directly actionable to optimize the process: it can easily be exploited to build new houses with better isolation and reduced energy consumption.

When only one target label is considered, SD can be applied. However, it is inherently limited to a single target, and there is a need for a framework that allows discovering the same kind of interesting information when several target labels have to be optimized at the same time. Exceptional Model Mining (EMM) is the task that allows this. It is a pattern discovery framework introduced more recently (Duivesteijn et al., 2016) as a generalization of SD. EMM is able to handle data where two or more targets exist, enabling the discovery of more complex interactions between variables. In EMM, we consider models instead of simple distributions on the target labels, and we look for subgroups whose models deviate significantly from the same model fitted on the entire dataset. Using the same scenario as for SD but now with both the energy consumption and the cost of the house as target labels, we could be interested in discovering informative subgroups of houses that optimize the energy/cost trade-off. Exploiting the descriptions of these subgroups would help in designing new houses with optimized energy/cost trade-offs. This is a difficult task: discovering optimal trade-offs between several variables is the subject of an entire field of research, namely Multi-objective Optimization (MOO).

Having access to generic methods that can solve any given optimization problem is essential to the development and proper functioning of numerous complex processes. Multi-objective optimization (Deb, 2014) is a sub-field of Multi-criteria Decision Making that is focused on finding globally optimal solutions for real-life problems that involve a set of usually conflicting objectives. For simple problems, we can use methods that transform the multi-objective optimization problems into single-objective ones and discover a single globally optimal solution. When dealing with more complex scenarios – such as plant growth optimization – scalarization techniques lead to sub-optimal results and using proper MOO methods that yield not one, but a set of optimal solutions is needed.
Inspired by nature and based on concepts from the theory of evolution (Eiben and Smith, 2015), evolutionary algorithms, and more precisely genetic algorithms represent by far the most widely used methods in MOO. As global optimization techniques, genetic algorithms are driven to converge toward global solutions, rather than local ones. Therefore, in the MOO setting, genetic algorithms aim at discovering the set of globally optimal solutions, i.e., the globally optimal trade-offs between the considered objectives.
Cross-fertilization between MOO methods and Pattern Mining techniques has unfortunately received little attention thus far, and even more so when the focus is being put on SD and EMM. In this thesis, we consider several complex problems that appear when we want to couple Pattern Discovery and MOO to solve real-life problems.

1.3 Problems Addressed in this Thesis

We want to design innovative Pattern Discovery methods to solve a particular set of Multi-objective Optimization problems, i.e., settings where there is a need to discover optimal parameter values when several numerical objectives are optimized at the same time. The application scenario at the heart of this research is the design of better plant growth recipes in controlled environments. Indeed, plant growth optimization is an intrinsic MOO problem where several competing objectives – such as yield and energy cost – need to be optimized concurrently. Therefore, optimizing plant growth means finding the best trade-offs between

these objectives. This is a crucially complex task: when optimizing recipes, the underlying model is unknown and experiments are severely limited due to time and cost constraints, making it impossible to exploit existing MOO approaches. We therefore need to devise methods that support the discovery of relevant and exploitable information in such MOO settings. To answer these limitations, let us now identify 3 important and open problems that need to be solved.

- ***Problem #1**: How can we exploit Pattern Discovery to discover relevant parameter values for Single-objective Optimization problems?*

While numerous optimization problems are multi-objective by nature, others only involve one objective to optimize. In the absence of knowledge about the underlying models that govern these processes, we need to be able to provide actionable information about the ideal parameters that lead to an optimized objective. For example, in the absence of knowledge about the cost of growth recipes, finding the ideal growing conditions that lead to an optimized yield is relevant. We therefore need to explore the development of pattern discovery methods that will help solve this problem.

- ***Problem #2:** How can we leverage Pattern Discovery to discover relevant parameter values for Multi-objective Optimization problems?*

As most real-life optimization problems involve multiple competing objectives, a large part of our work needs to focus on devising methods that can enable the discovery of relevant information about the parameter values that lead to optimal trade-offs between these objectives. For example, plant growth optimization can involve not only maximizing the yield but also minimizing the cost of the recipes at the same time. In this setting, finding the environment parameter values that optimize both objectives simultaneously is crucial.
However, answering these two crucial problems would be of limited importance, if their relevance could not be confirmed in a real-life situation. This leads us to introduce our third and last problem.

- ***Problem #3:** In the absence of real data, how can we assess the performance of our contributions?*

Since obtaining access to real-life plant growth data is difficult, we need to find ways to assess the relevance and actionability of our methods on comparable problems and/or data. Notice however that producing synthetic data can be hard when qualitative aspects are to be assessed.

1.4 Contributions

Having now introduced the main problems that this thesis tackles, we detail our contributions.

1.4.1 OSMIND: A New Algorithm for Optimal Subgroup Discovery in Purely Numerical Data

Subgroup Discovery is a local pattern detection technique that aims at discovering subsets of objects in a dataset – subgroups – whose target label distribution significantly deviates from

that of the whole dataset. Mining subgroups in purely numerical data has unfortunately received little attention thus far. The few proposed methods usually involve the use of discretization methods on the numerical attributes. It is however well-known for inducing loss of information, suboptimal results, and irrelevant patterns.
To solve these issues, we propose OSMIND, an SD algorithm that enables the discovery of an optimal pattern in purely numerical data when the q^a_{mean} family of quality measures is considered. To ensure the optimality of the search, we consider the search space of interval patterns as defined in (Kaytoue et al., 2011). OSMIND leverages the concept of closure on the positives adapted to a numerical setting to compress the size of the search space. Furthermore, we introduce a new tight optimistic estimate and exploit advanced techniques that allow for the pruning of irrelevant patterns efficiently. Finally, we demonstrate the relevance of OSMIND against the state of the art algorithm SD-MAP$\star$ (Atzmueller and Lemmerich, 2009) in a thorough empirical evaluation.

1.4.2 Exceptional Pareto Front Mining: A New EMM Model Class to Support Multi-Objective Optimization.

While OSMIND is a good first step toward discovering relevant parameter values (i.e., the description of the optimal subgroup) driving the optimization of a process, it is by essence limited to single-objective problems. However, in reality, most processes involve various competing objectives that need to be optimized concurrently. We therefore consider Exceptional Model Mining, a framework that generalizes SD and is able to deal with problems where several objectives are involved and complex interactions between them have to be (better) understood. While the literature on Pareto-based MOO is well-supplied, existing approaches cannot be used when the underlying model is unknown and/or experiments are limited due to time and cost constraints. There is a need for methods that would support the discovery of relevant and exploitable information in such settings.
We design a new class of models for EMM, namely Exceptional Pareto Front Mining (EPFM), and introduce two methods that fit the class: Exceptional Pareto Front Deviation Mining (EPFDM) and Exceptional Pareto Front Approximation Mining (EPFAM). EPFDM discovers exceptional deviations between the shape of the Pareto front left by the absence of a subgroup of objects and the shape of the Pareto front of the overall dataset. EPFAM enables the discovery of models that approximate exceptionally well the true Pareto front. To reframe these contributions in our MOO setting: EPFDM can serve as a data analysis tool to discover interesting knowledge regarding MOO problems, while EPFAM enables the generation of Pareto optimal solutions with a higher probability by exploiting the description of the best subgroup (i.e., the best approximation). The relevance and effectiveness of both approaches are confirmed through a thorough empirical study that includes a use case to hyperparameter optimization in Machine Learning.

1.4.3 Optimizing Plant Growth Recipes in Controlled Environments.

The relevance of both our SD and EMM approaches has been validated to support the discovery of relevant sets of parameter values for single and multi-objective optimization processes. We investigate their actionability for plant growth recipe optimization in controlled environments.

We first find a way to exploit an existing crop simulator to generate synthetic recipes that replicate a controlled environment. Using synthetic recipes, we then investigate the optimization of plant growth in a controlled environment when a single objective is considered. Since existing methods fall short of real-life constraints, we propose a new iterative optimization framework – based on a virtuous circle principle – that exploits the actionability of subgroup descriptions to generate better growth recipes. Indeed, at each iteration, the description of the optimal subgroup of recipes is directly used to sample the recipes of the next iteration. Next, we show how EPFM can be used to support recipe optimization in a multi-objective setting. In particular, we propose a simple iterative process that exploits EPFAM and the descriptions of subgroups to iteratively optimize the yield/cost trade-off of recipes.
Finally, we apply both our SD and EMM methods to optimize the growth recipes of basil thanks to a temporary access during summer 2020 to a real-life FUL operational urban farm. Preliminary results confirm the potential of our methods to optimize recipes, both in single-objective and multi-objective optimization settings.

1.5 Structure of the Thesis

The remainder of this thesis is organized as follows:

- In Chapter 2, we first propose an overview of the SD task, its different components, and the various contributions which have been introduced since its inception. We propose a more detailed review of the contributions for SD in numerical domains, i.e., when numerical attributes and/or numerical targets are considered. We then investigate optimal SD for each type of dataset that is commonly encountered in SD. Finally, we consider the overall literature of EMM, the fairly recent generalization of SD.

- In Chapter 3, we first detail the reasons why studying the cross-fertilization between Pattern Discovery and MOO is relevant. We then propose an overview of MOO, that provides important information regarding the relevance and actionability of existing methods to help solve our problems. We first review both classical and Pareto-based approaches to MOO. Next, we consider the literature on quality evaluation and benchmark functions for MOO. Then, we detail existing tools and application cases of MOO. Finally, we propose a detailed review of cross-fertilization between Pattern Discovery and MOO.

- Chapter 4 is dedicated to our first contribution, the `OSMIND` algorithm for an optimal SD in purely numerical data. We leverage the concept of closed interval patterns and advanced enumeration and pruning techniques. The relevance of our algorithm is studied empirically and its added value with regard to the state of the art is illustrated. This contribution has been published in the Proceedings of the 2020 Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD) (Millot et al., 2020a), and the Proceedings of the 2020 conférence Extraction et Gestion des Connaissances (EGC) (Millot et al., 2020b).

- In Chapter 5, we investigate methods that exploit Exceptional Pareto Front Mining (EPFM), a new model class for EMM. Two approaches, Exceptional Pareto Front Deviation Mining (EPFDM) and Exceptional Pareto Front Approximation Mining (EP-

FAM) are detailed. Then, an in-depth empirical evaluation, as well as an application scenario to hyperparameter optimization in Machine Learning, confirm the relevance of these methods. This contribution has been partially published in the Proceedings of the 2021 SIAM International Conference on Data Mining (SDM) (Millot et al., 2021). An extended version is currently under review for publication in the Data Mining and Knowledge Discovery (DAMI) journal (submitted in March 2021).

- Chapter 6 investigates the actionability of our contributions for plant growth recipe optimization in controlled environments like urban farms, the real-life setting that has motivated our research. Furthermore, an iterative optimization framework based on the actionability of SD is introduced. The relevance of our methods to optimize plant growth recipes in both single and multi-objective optimization settings is validated on synthetic and real-life data. Part of this chapter has been published in the Proceedings of the 2020 International Symposium on Intelligent Data Analysis (IDA) (Millot et al., 2020c).

- Chapter 7 concludes and details perspectives for future work.

1.6 List of Publications

Peer-reviewed French national conferences:

- **Alexandre Millot**, Rémy Cazabet, and Jean-François Boulicaut. *Découverte d'un sous-groupe optimal dans des données purement numériques.* In Extraction et Gestion des Connaissances : EGC 2020, Bruxelles, Belgique, January 27-31, 2020, pages 25-36. Best academic paper award.

Peer-reviewed international conferences with proceedings:

- **Alexandre Millot**, Rémy Cazabet, and Jean-François Boulicaut. *Exceptional Model Mining meets Multi-objective Optimization.* In Proceedings of the 2021 SIAM International Conference on Data Mining, SDM 2021, Alexandria, Virginia, U.S, Apr 29, 2021 – May 1, 2021, pages 378-386.

- **Alexandre Millot**, Rémy Cazabet, and Jean-François Boulicaut. *Optimal subgroup discovery in purely numerical data.* In Pacific-Asia Conference on Knowledge Discovery and Data Mining, PAKDD 2020, Singapour, Singapour, May 11-14, 2020, pages 112-124.

- **Alexandre Millot**, Romain Mathonat, Rémy Cazabet, and Jean-François Boulicaut. *Actionable subgroup discovery and urban farm optimization.* In International Symposium on Intelligent Data Analysis, IDA 2020, Konstanz, Germany, April 27-29, 2020, pages 339-351.

The following work is currently undergoing the **reviewing process**:

International journals:

- **Alexandre Millot**, Rémy Cazabet, and Jean-François Boulicaut. *Exceptional Model Mining to support Multi-objective Optimization.* In Data Mining and Knowledge Discovery, 38 pages. (Submitted in March 2021).

Chapter 2

Subgroup Discovery and Exceptional Model Mining

In this chapter, we have several goals in mind: (i) to propose a detailed overview of both Subgroup Discovery and Exceptional Model Mining, (ii) to introduce an in-depth review of both Subgroup Discovery in numerical domains and Optimal Subgroup Discovery, as well as their current limitations, (iii) to introduce and formalize the concepts necessary to the understanding of the rest of the thesis. We first propose an overview of the Subgroup Discovery process, its components, and the numerous contributions which have been made to the field since its inception. We propose a more detailed review of the contributions proposed for Subgroup Discovery in numerical domains, i.e., with numerical attributes and/or numerical targets. We investigate existing approaches that enable the discovery of the optimal subgroup, for each type of dataset that is commonly encountered in Subgroup Discovery. Finally, we consider the overall literature that deals with Exceptional Model Mining, the generalization of Subgroup Discovery.

2.1 Introduction

Over the past 25 years, the digital revolution led to giant improvements in computing power, data storage, and information recording. While in the past data availability used to be a problem for researchers, they now have to deal with large masses of information to exploit and information overload is a real growing concern. Extracting interesting and actionable patterns from such data using computer-based techniques is a real challenge that could yield enormous benefits for researchers, companies, and society as a whole. The need to discover patterns in data is nothing new. Association rule mining, which allows for the discovery of rules that describe interesting relations between attributes of a dataset, was introduced almost 30 years ago (Agrawal et al., 1993). Using data mining methods for knowledge discovery (Piatetsky-Shapiro et al., 1996) then became an important area of research, that lead to the introduction of numerous new contributions throughout the years. Although the development of new techniques, algorithms, and methodologies for efficient pattern discovery in diverse scenarios has been a focus of most researchers in the past, the focus has more recently turned to the discovery of the right patterns (see, e.g., (Bringmann and Zimmermann, 2009)). Indeed, when dealing with large search spaces, a gigantic number of patterns can be discovered, of which only a few might actually be relevant and exploitable. Nowadays, a large chunk of data that is being processed could be defined as *labeled data*, i.e., data made of objects defined by a set of descriptive attributes and one or several labels. Discovering relevant patterns in labeled data, e.g., thanks to Subgroup Discovery (SD) is an important data mining subarea that has captivated the attention of numerous researchers.
The remaining of this chapter is organized as follows. Section 2.2 contains an overview of SD and a formalization of the discovery task. In Section 2.3, we review SD with numerical attributes. We investigate SD with numerical targets in Section 2.4. Optimal SD is considered in Section 2.5. In Section 2.6, we review contributions made to EMM. Finally, Section 2.7 concludes.

2.2 Overview of Subgroup Discovery

2.2.1 Definition

Subgroup Discovery is a local pattern detection technique (Morik et al., 2005) whose birth is attributed to (Klösgen, 1996, Wrobel, 1997), although the idea of discovering *"interesting subgroups in a database"* was already pointed out in (Siebes, 1995). It aims at discovering subsets of objects in a dataset – subgroups – whose target label distribution statistically deviates from that of the overall dataset. To measure the significance of that deviation, a quality measure that takes into account both the generalization power of the subgroup and its deviation from the norm is usually used. Interestingly, subgroups are defined by means of patterns that are also called descriptions. These patterns are by construction understandable by humans and can help in discovering interesting knowledge. Following its introduction, numerous contributions (Atzmueller, 2015, Herrera et al., 2011) which investigate the different facets of SD have been proposed. It is interesting to note the close relationships that SD holds with *association rule mining* (Agrawal et al., 1996) and *frequent pattern mining* (Han et al., 2000), but also *emerging patterns* (Dong and Li, 1999) and *contrast sets* (Bay and Pazzani, 2001).

In Subgroup Discovery, a dataset (G, M, T) is a set of objects G, a set of attributes M, and a single target T. In a given dataset, the set of attributes M and the target T can contain real or nominal values. Table 2.1 depicts an example of a dataset structure that can be used in SD. It is made of 5 descriptive attributes, of which 4 are nominal and 1 is numerical. It has a unique binary target.
SD has been mainly concerned with nominal attributes and binary target labels. To deal with numerical data, prior discretization of the attributes (Fayyad and Irani, 1993, Garcia et al., 2013) is usually required. Numerical target labels can also be discretized (Moreland and Truemper, 2009). However, discretization generally involves loss of information such that the optimality of the returned subgroups w.r.t. a given measure cannot be guaranteed.

Table 2.1: Toy example of a dataset.

	gender	age	status	occupation	smokes	**Cancer**
g_1	M	25	Married	Nurse	Yes	No
g_2	F	59	Single	Engineer	Yes	Yes
g_3	M	64	Divorced	Researcher	No	No
g_4	M	33	Single	Driver	No	No
g_5	F	61	Married	Teacher	Yes	Yes

The search space of subgroups consists of a very large set of subgroup descriptions (or intents), and each description is made of a set of constraints on some attributes of the dataset. Those constraints are linked to each other to form a proper subgroup description using boolean operators. The type of constraints considered on the attributes (e.g., $<$,$>$,$\leqslant$,$\geqslant$,$=$,$\neq$,*interval membership*), as well as the type of boolean operators (e.g., AND, OR, NOT) define what we call the *pattern language*.
Considering the standard $=$ constraint for nominal attributes, *intervals* membership for numerical attributes, and *conjunctions* (AND) of attributes to build the descriptions, an example of subgroup description that respects the defined pattern language using the toy dataset of Table 2.1 would be $\langle$ age $\in$ [59,64] AND status = 'Married' $\rangle$.

Given a pattern language and a dataset, subgroups can be enumerated by applying a refinement operator either on their description (intent) or on their coverage (extent). We now give proper definitions of the intent and extent of a subgroup.

Definition 1. *The intent of a subgroup p is given by $p_d = \langle \varphi_1, \ldots, \varphi_{|M|} \rangle$ where each φ_i is a restriction on the domain value of $m_i \in M$.*

Definition 2. *The extent of a given subgroup p, denoted $ext(p) \subseteq G$, is the set of objects of G that satisfy the restrictions of p_d.*

For example, given the toy dataset in Table 2.1, we could consider a subgroup whose intent is $\langle$ gender = 'M' AND age $\in$ [25,64] $\rangle$ and whose extent is $\{g_1, g_3, g_4\}$.
Although most SD settings consider a comparison between subgroups and the overall dataset, comparing subgroups and their complements can be considered, e.g., in (Pieters et al., 2010).

Definition 3. *The complement of a subgroup p, denoted $\overline{p}$, consists in the set of objects of the dataset that are not in ext(p).*

For example, the complement of the subgroup defined in the previous example and supported by $\langle$ gender = 'M' AND age $\in$ [25,64] $\rangle$ is $\{g_2, g_5\}$.
When a refinement operator is applied to the description of a subgroup p (i.e., when a new constraint is added to the description), it produces a specialization of p.

Definition 4. *Let p_d and p'_d be the descriptions of subgroups p and p'. p'_d is said to be a specialization of p_d if and only if $p_d \subset p'_d$.*

For example, a specialization of $\langle$ gender = 'M' AND age $\in$ [25,64] $\rangle$ is $\langle$ gender = 'M' AND age $\in$ [25,64] AND smokes = 'No' $\rangle$, and its extent is $\{g_3, g_4\}$.

2.2.2 Search Space Exploration

Exploring the search space efficiently is among the most critical problems of SD since the size of the search space grows exponentially in the number of attributes. Many enumeration strategies for the subgroup discovery process have been studied. They can be grouped into 2 main categories: heuristic and exhaustive.
Heuristic approaches are employed when the search space is too large to be handled exhaustively. Using such strategies, the guarantee to discover optimal patterns is lost, at the benefit of tractability and running time. The goal is then to develop a strategy that enables the discovery of high-quality patterns without neglecting diversity. Although several strategies have been introduced (Lavrac et al., 2004, Luna et al., 2013, Mampaey et al., 2015), the search space is most commonly explored using breadth-first search, as in the well-known *beam search* algorithm. (Van Leeuwen and Ukkonen, 2013) and (Proença et al., 2021) contain examples of algorithms that exploit beam search to discover high-quality non-redundant subgroups.
Sampling-based methods (Boley et al., 2011, 2012), although less common and heuristic by nature, have also been used sparsely for subgroup discovery. Their main advantage is the discovery of high-quality patterns in a very low amount of time. In this strategy, a statistical distribution based on the optimization of quality criteria is usually designed, such that patterns that optimize those criteria have a much higher probability to be generated.
Exhaustive algorithms, that trade-off execution time – and possibly feasibility – for the guarantee of the optimality of the discovery, are popular in SD. Since the search space is usually too large to enumerate the patterns exhaustively, diverse techniques can be used to render the search tractable. Compressing the search space through the use of equivalence classes and closure systems (Boley and Grosskreutz, 2009, Grosskreutz and Paurat, 2011, Lemmerich et al., 2010) are common approaches. Pruning the number of candidates using anti-monotone constraints (Kavšek et al., 2003) and optimistic estimates on the quality of the specializations of subgroups (Belfodil et al., 2019a, Grosskreutz et al., 2008, Lemmerich et al., 2016a, Zimmermann and De Raedt, 2009) is also widely used for exhaustive SD. Finally, special data structures that improve the efficiency of the search can be used, such as in `SD-Map` (Atzmueller and Puppe, 2006a) which exploits the well-known FP-trees (Han et al., 2000), and (Lemmerich et al., 2016a) that considers a modified bitset-based data structure.
Finally, the use of anytime algorithms that allow for the retrieval of the best set of patterns at any given moment during the search has been investigated, although sparsely too. `MCTS4DM`

(Bosc et al., 2018) and `Refine&Mine` (Belfodil et al., 2018) are both anytime algorithms. Anytime subgroup discovery combines some of the strength of the previous strategies: (i) it provides subgroups instantly if needed, (ii) a set of high quality and highly diverse subgroups can be retrieved at anytime, (iii) the quality of subgroups increases as time goes on, (iv) the discovery goes from heuristic to exhaustive if the search is left to run until complete, though it is not possible in most of the real cases. The use of an anytime algorithm for subgroup discovery in labeled sequential data was also investigated in (Mathonat et al., 2019, 2021).

2.2.3 Relevance of Subgroups

When applying subgroup discovery methods, we have to deal with a huge number of patterns, of which only a few will be of interest in a given context. There is therefore a need to define criteria that will allow us to differentiate between relevant and irrelevant subgroups, hopefully during the search and not in a post-processing step. The idea that (i) user-defined constraints can specify a priori desired properties for patterns and (ii) enumeration techniques can exploit (efficiently) these constraints to avoid the computation of irrelevant patterns has given rise to the prolific research domain of constraint-based data mining. This has been the core algorithmic contribution to the so-called inductive database framework (Boulicaut et al., 2005, Dzeroski et al., 2010).

Primitive constraint can refer or not to data. Many useful primitive constraints make use of interestingness measures (see surveys on such measures in for instance (Freitas, 1999, Geng and Hamilton, 2006)). Researchers have studied smart properties of useful constraints to be able to perform an efficient search of a priori relevant descriptive patterns like, for instance, frequent and valid association rules. Constraint-based Subgroup Discovery has been studied as well (Lavrač and Gamberger, 2006). For instance, we can consider syntactic constraints on the descriptive attributes like a maximum number of conditions in the intent of the subgroups, or a proper range of values for each attribute. We can also exploit constraints on the size of the extent of the subgroup, i.e., a minimum (maximum) number of objects.

We may use primitive constraints that specify the search for the Top-K best patterns with respect to a given quality measure. Indeed, considering a given dataset, the interestingness of each subgroup can be measured by a numerical value. Usually, the value quantifies the discrepancy between the target label distribution of the subgroup and that of the overall dataset (i.e., its discriminative power). Since important discrepancies can easily be achieved with small subsets of objects, a factor that takes into account the coverage of the subgroup in some form (i.e., its generalization power) is usually introduced in the quality measure. In SD, new quality measures can be designed or adapted for each given context. Therefore, we find a large panel of indicators in the literature, such as measures for binary targets (Herrera et al., 2011, Li et al., 2014), for numerical targets (Boley et al., 2017, Lemmerich, 2014), and also for multi-class nominal targets (Abudawood and Flach, 2009).

For binary targets, the most commonly used measure is the Weighted Relative Accuracy (WRAcc) (Lavrač et al., 1999). The WRacc compares the proportion of positive objects in the extent of the subgroup to the proportion of positive objects in the overall dataset. It is given by:

$$WRAcc(p) = freq(p) \times (\delta_{ext(p)} - \delta_{ext(\varnothing)})$$

with $freq(p)$ the frequency of the subgroup in the dataset, $\delta_{ext(p)}$ the proportion of positive objects in the extent of p, and $\delta_{ext(\varnothing)}$ the proportion of positive objects in the overall dataset. The frequency serves as a generalization optimizer so that subgroups with larger coverage are favored. The WRAcc takes values in the range [-0.25,0.25].

As an example of quality measure for numerical targets, we propose to consider the popular family of quality measures based on the mean introduced in (Lemmerich et al., 2016a). Given a subgroup p, its quality is given by:

$$q^a_{mean}(p) = |ext(p)|^a \times (\mu_{ext(p)} - \mu_{ext(\varnothing)}), a \in [0, 1]$$

with $\mu_{ext(p)}$ the mean of the target label for p, $\mu_{ext(\varnothing)}$ the mean of the target label for the overall dataset, $|ext(p)|$ the cardinality of $ext(p)$ and a a parameter that controls the number of objects in the subgroups. With lower values of a, smaller subgroups are favored, while it is the opposite for larger values of a.

It is interesting to note that constraints on the minimum significance or interestingness required for the subgroups can also be used to guide the search.

Generalization-aware subgroup discovery.

In SD, we often discover subgroups whose target distribution is close to that of one or more of its generalizations. In this setting, the subgroups might be deemed interesting according to a quality measure, although they are not since they do not deviate from their generalizations. To remedy this problem, generalization-aware (Lemmerich and Puppe, 2011, Lemmerich et al., 2013) subgroup discovery can be exploited to prune irrelevant specializations that might make the result set less diverse. In generalization-aware subgroup discovery, the quality of a subgroup is measured by comparing it to its generalizations. To do this, a relatively simple modification on common measures can be applied, such that we compare the distribution of the target in the subgroup and the distribution of the target in its best generalization. With generalization-aware SD, a basic measure can be defined as:

$$q(p) = gen(p) \times (\psi_{ext(p)} - \max_{G \subset p} \psi_{ext(G)})$$

with $gen(p)$ the generalization power of the subgroup, $\psi_{ext(p)}$ the target label distribution in the extent of p, and $\max_{G \subset p} \psi_{ext(G)}$ the target label distribution of the generalization of p that deviates the most from the overall dataset.

Statistical significance.

A common issue with subgroup discovery methods is the lack of statistical significance behind many of the patterns described as “interesting” or “relevant”. Although several contributions have been made for the discovery of statistically significant patterns in association rule mining (Hämäläinen, 2010, Zhang et al., 2004) and the generic pattern mining framework

(Hämäläinen and Webb, 2019), few works have investigated the notion of statistical significance in the patterns produced by subgroup discovery algorithms.
The most significant work in the SD field is that of (Duivesteijn and Knobbe, 2011). It is argued that SD suffers from the multiple comparisons problem, i.e., when looking for exceptional deviations in a large search space, an algorithm is bound to discover interesting subgroups, although most of those might correspond to false discoveries. The authors therefore build a statistical model that detects false discoveries – through the generation of a random baseline model – such that the statistical significance of each subgroup can be validated by comparing its deviation from the statistical model. Going further, the authors propose the application of this method to determine the statistical significance of quality measures. This is done by measuring by how much the top subgroups found with a given quality measure deviate from the random baseline generated for that measure. By applying this method to several well-known quality measures, their statistical significance can be compared. Twelve measures are compared, and the authors conclude that the worse measures are Purity and Sensitivity, while χ^2 has the highest statistical significance.
The discovery of statistically non-redundant subgroups was investigated in (Li et al., 2014). To this end, *odds ratio* is used as a statistically sound quality measure, and the statistical significance of the subgroups is measured using the confidence intervals of odds ratios. Finally, they introduce an algorithm for the optimal subgroup discovery of statistically non-redundant subgroups using tight optimistic estimates and a pruning strategy. However, this only works when the *odds ratio* quality measure is considered.

Background knowledge and subjective interestingness.

Exploiting background knowledge (e.g., expert knowledge, domain literature, or information specific to a given setting) can be an important part of the subgroup process for many application scenarios. The concept of *Expert-guided Subgroup Discovery* was introduced in (Gamberger and Lavrac, 2002a) and (Gamberger and Lavrac, 2002b). *Expert-guided Subgroup Discovery* is an iterative discovery process that involves exploiting the input of an expert at each step of the process. In a first step, a set of apriori interesting subgroups are selected and presented to the expert, who gives directions according to his knowledge as to how to proceed for the next iteration. The expert can select a subset of more interesting subgroups, or give his input on the selection or removal of certain descriptive variables for example. Consequently, the pattern mining process exploits both expert knowledge and objective measures for the discovery of relevant and exploitable subgroups. Both contributions provide a proper methodology for the discovery of exploitable information using expert knowledge, and apply it to real-life scenarios to show its relevance.
Exploitation of background and expert knowledge in the subgroup discovery process has also been studied in-depth for *Knowledge-intensive Subgroup Discovery* (Atzmueller and Puppe, 2006b, Atzmueller et al., 2004, 2005). They propose to apply as much background knowledge as possible from the start of the discovery process, but also to add knowledge throughout the interactive iterative process. For subgroup discovery, background knowledge can take many forms. It can consist in applying constraints on the different components of the discovery process. For example, constraints can be applied to the values of the descriptive attributes,

but also constraints can be used to remove irrelevant attributes. Expert preferences can also be included in the design of the quality measure. The pattern language can also be modified to take into account relevant knowledge, and weights can be used on descriptive attributes to drive the search on certain parts of the search space. Using background knowledge can help avoiding the discovery of too many subgroups, but also discovering uninteresting subgroups. Through application scenarios in the medical field, it was shown that exploiting background knowledge can help focusing the search on already known interesting parts of the search space, leading to the discovery of subgroups of higher quality.

Avoiding redundancy.

One of the main problems of subgroup discovery and other pattern discovery methods is the sheer amount of redundant patterns that can be discovered during the discovery process. There is therefore a need for techniques that favor the discovery of diverse subgroups. The most common technique is to use *weighted covering*, which exploits a weighting scheme on the objects of the dataset (Kavsek and Lavrac, 2004a, Kavšek et al., 2003, 2004, Lavrac et al., 2004) in an iterative process. In this approach, each object of a given dataset is assigned a weight (usually 1) at the start of the search process, and a quality measure that takes into account the weight of each object in its computation is devised. Then, the given SD algorithm is executed, and the best subgroup found is retrieved. Next, the weight of each object that is part of the best subgroup is decreased following the predefined weighting scheme. The SD algorithm is repeated once again, but this time using the reweighted objects, and so on. The principle behind this method is to iteratively reduce the quality of subgroups made of objects that have already been part of the best subgroups in the previous iterations. Using this technique, the diversity of the set of discovered subgroups is usually greatly improved. The well-known `CN2-SD` and `Apriori-SD` algorithms use such a weighting scheme. This approach was inspired by the sequential covering approach introduced in the `CN2` algorithm (Clark and Niblett, 1989), although in `CN2` the objects that are part of the best pattern at each iteration are removed from the dataset instead of being reweighted. (Scholz, 2005) also introduced an iterative process using a weighted scheme for the discovery of a small diverse set of interesting subgroups. However, contrary to existing approaches, the new weighting scheme allows for the incorporation of previously discovered knowledge in the reweighting, such that already discovered knowledge should not be rediscovered in the next iterations. The incorporation of prior knowledge is made possible through the use of *Rejection Sampling*.
The concept of *subgroup set discovery* was introduced by (Van Leeuwen and Knobbe, 2012). In *subgroup set discovery*, instead of looking at individual subgroups, we are interested in the discovery of sets of high-quality non-redundant subgroups. A method to mine for such subgroup sets – called Diverse Subgroup Set Discovery (`DSSD`) – is devised for both subgroup discovery and exceptional model mining. The relevance of the approach compared to *weighted covering* is studied, and results show that `DSSD` can find comparable results in a significantly lower amount of time.
In (Belfodil et al., 2019a), a new approach for *subgroup set discovery* that incorporates both the interestingness and the diversity of the subgroup in the same quality measure is introduced. The corresponding efficient algorithm, `FSSD` (Fast Subgroup Set Discovery) is able to discover overall better and more diverse subgroups than `CN2-SD` and `DSSD` in a shorter

amount of time. The concept of *skyline* was exploited in (Van Leeuwen and Ukkonen, 2013) to mine for sets of high-quality non-redundant subgroups that offer the best trade-offs between quality and diversity. (Li et al., 2014) also introduced an optimal algorithm for the discovery of statistically non-redundant subgroups.
Finally, (Bosc et al., 2018) proposed an anytime algorithm – `MCTS4DM` – for the discovery of a diverse set of subgroups by cross-fertilization of Monte Carlo Tree Search (MCTS) and SD. MCTS finds local optima iteratively by generating random simulations of the search tree and guiding the search exploiting an exploration/exploitation trade-off, which ensures the diversity of the resultant subgroups. One of the main strengths of `MCTS4DM` is that any pattern language can theoretically be used, e.g., nominal data, numerical data, using conjunctions, or disjunctions on attributes, etc.

2.2.4 Optimizing the Search

Compressing the search space.

In subgroup discovery, the size of the pattern search space can quickly become too large to handle, especially when exhaustive enumeration strategies are involved. In Association Rule Mining, condensed representations of the data in the form of δ-free sets have been used to discover simple rules the characterize classes (Crémilleux and Boulicaut, 2003). Using closure operators and equivalence classes (Bastide et al., 2000, Grosskreutz and IAIS, 2012, Soulet et al., 2004, Wang, 2005) are popular solutions to reduce the number of explored subgroups. (Garriga et al., 2008) introduced the concept of closed-on-the-positives for binary labeled data by adapting the existing closure system of itemsets. In (Boley and Grosskreutz, 2009), the authors make use of extension-based classes of equivalence, such that each extension has only one pattern. (Lemmerich et al., 2010) developed the `BSD` algorithm for the discovery of relevant subgroups. Relevant subgroups are a very restrictive class of patterns, that is, a subset of closed and closed-on-the-positives subgroups. (Grosskreutz and Paurat, 2011) also introduced an efficient algorithm for top-K subgroup discovery using a relevancy check on patterns. It is interesting to note that these methods for compressing the search space can also be used to avoid redundancy, as in (Boley and Grosskreutz, 2009, Li et al., 2014).

Pruning the search space.

When the number of potential subgroups is too large to be efficiently explored, pruning techniques that allow for removing entire parts of search space from the discovery process can be exploited. Optimistic estimates, that define upper-bound values for the quality of entire sets of patterns, are the most common way of pruning the search space (Li et al., 2014, Morishita and Sese, 2000): if the optimistic estimate of a subgroup is lower than the required minimal quality, it is useless to consider its specializations.

Definition 5. *Given a subgroup p and a quality measure q, an optimistic estimate for q, denoted as bs_q, is a function that gives an upper bound for the quality of all specializations of p. Formally, $\forall s \subseteq p : q(s) \leqslant bs_q(p)$.*

In (Grosskreutz et al., 2008), the authors introduce the concept of *tight* optimistic estimates for subgroup discovery with binary targets. An optimistic estimate is said to be tight if it is the most restrictive estimate that can be made (i.e., the lowest value) according to the information available. Optimistic estimates for numerical targets have also been investigated in (Lemmerich, 2014, Lemmerich et al., 2016a). Although optimistic estimates cover most of the pruning used in SD, anti-monotone constraints can also be exploited in certain cases, e.g., if a minimum coverage is defined for the subgroups.

Handling big data.

Subgroup discovery in big data is explored in both (Cano et al., 2008) and (Padillo et al., 2016). Indeed, the subgroup discovery process faces difficulties when either the search space is too large, or when the cost of computing each subgroup is too high. The use of a combination of stratification and instance selection algorithms is investigated in (Cano et al., 2008) to remedy these issues. Using a different approach, (Padillo et al., 2016) makes use of the MapReduce framework and optimistic estimates to efficiently explore the search space. Both approaches show the relevance of their methods in experimental studies.

2.2.5 Tools and Applications

To democratize the use of SD, several easy-to-use tools have been developed. Although numerous software allow for the development, plug-in, and use of subgroup discovery algorithms (Alcalá-Fdez et al., 2009, Berthold et al., 2009, Meeng and Knobbe, 2011, Witten and Frank, 2002), few specialized SD tools exist. The VIKAMINE (Visual, Interactive and Knowledge-intensive Analysis and Mining Environment) system (Atzmueller and Lemmerich, 2012) was introduced in 2005, and then refined throughout the years. The system contains several subgroup discovery algorithms, as well as widely used quality measures that enable fast and efficient subgroup discovery for any user. Data preparation and visualization tools are also available. Numerous advanced functions for the analysis of the characteristics of the discovered subgroups are also accessible.
With Python becoming the standard programming language for data analysis-related tasks, the *pysubgroup* (Lemmerich and Becker, 2018) package for subgroup discovery was recently developed. Based on the most widely used data processing Python packages – *Pandas* and *Numpy* – *pysubgroup* provides a simple and exploitable framework where only a few lines of code are needed to run a subgroup discovery process on a dataset. It currently features the most widely used subgroup discovery enumeration strategies (e.g., `Apriori`, `beam search`, `BSD`, depth-first-search, best-first-search) and numerous quality measures for binary and numerical targets. Furthermore, the package can easily be extended with new algorithms and quality measures by users. Finally, visualization functionalities that improve the exploitability of the results are also available.

Although the subgroup discovery process itself still struggles to find its place in the toolbox of most companies for data-related tasks, numerous real-life application cases (Atzmueller and Puppe, 2008, Lavrač et al., 2004) to subgroup discovery in diverse domains have been presented in the last 20 years.

A common use of subgroup discovery at the start of the century was its application in the medical field. Indeed, we find several application cases detailed in (Gamberger et al., 2003) for the detection of coronary heart disease risk groups, in (Mueller et al., 2009) for breast cancer diagnosis, and also in (Gamberger et al., 2007) where an iterative approach to the discovery process is applied to analyze brain ischaemia data.
Election data analysis was explored in (Grosskreutz et al., 2010). Using 2009 German federal Bundestag election data and socio-economic information, the authors seek out to discover subgroups that describe interesting voting behavior. Among other results, the authors discover subgroups of voters with a strong preference for the winning party, and whose socio-economic description contains information such as "high average living space per accommodation" and "high share of detached houses".
More recently, SD was exploited to identify key factors of student academic performance (Pass or Fail) (Helal et al., 2019). In their experiments, the authors find, among other results, that the students who are the most likely to fail either come from low socio-economic backgrounds or were admitted through special entry requirements.
In (Centeio Jorge et al., 2021), spatio-temporal data that describes interactions between children in the school play yard was analyzed and the subgroup discovery process was exploited. One of the goals was to discover subgroups of children that presented exceptional behavior. Relevant behaviors, although already known by domain experts, such as gender homophily, and individuals having strong influence on groups of peers were detected.
Subgroup discovery was also applied to biological data aggregation in (Pieters et al., 2010), to UK traffic data analysis in (Kavsek and Lavrac, 2004b), to logistics data in (Sternberg and Atzmueller, 2018), to smart electricity meter data in (Jin et al., 2014), and for uncovering structure-property relationships of materials in (Goldsmith et al., 2017).

2.2.6 Subgroup Discovery in Atypical Data

Although we have now given a wide overview on contributions related to SD, several other specialized contributions also exist. We first find a Redescription Mining approach to SD in (Gallo et al., 2008). Given a dataset made of boolean values, the goal is to discover subgroups for which at least two significantly different descriptions exist (in terms of Boolean formulas). Algorithms that exploit pruning techniques are introduced, and experimental results show the relevance of the approach.
Community Detection in graphs using subgroup discovery is investigated in (Atzmueller et al., 2016). The goal is to discover subsets of nodes (subgroups) that show a deviation from the norm of the overall graph. An exhaustive branch-and-bound algorithm that exploits efficient pruning techniques is also presented. The discovery of interesting subgroups in graph data is also explored in (Deng et al., 2020). A method is developed to mine for pairs of nodes (subgroups) whose edge density is significantly different (higher or lower) from that of the overall graph.
Finally, subgroup discovery in sequential data is proposed in (Mathonat et al., 2019) and has been extended in (Mathonat et al., 2021). The anytime sampling algorithm – called `SeqScout` – exploits a multi-armed bandit model to mine interesting sequential patterns. Given a budget, the algorithm discovers locally optimal subgroups. Furthermore, the method possesses two main advantages (i) its configuration is relatively simple (ii) it is generic to any quality measure. The relevance of the approach is verified through qualitative and

quantitative results.

2.3 Subgroup Discovery with Numerical Attributes

2.3.1 Dataset, Pattern Language and Search Space

We assume that a numerical dataset (G, M, T) is a set of objects G, a set of numerical attributes M, and a single target T. In a given dataset, the domain of any attribute $m \in M$ is a finite ordered set denoted D_m, and the target T can contain real or nominal values. Figure 2.1 (left) depicts an example of numerical dataset structure used in SD. It is made of 2 numerical descriptive attributes and a unique binary target.

To deal with numerical attributes natively, the *pattern language* usually involves conjunctions or disjunctions of intervals over the domain of the considered attributes. An interval is made of 2 components, called cut-points or bounds. The left bound is the lower bound, while the right bound is the upper bound. Although closed intervals are much more common, open and half-open intervals can also be used.
For example, given the toy dataset of Figure 2.1 (left), $m_1 \in [2, 4]$ means that $m_1 \geqslant 2$ (lower bound) AND $m_1 \leqslant 4$ (upper bound). Furthermore, an example of a subgroup intent given a *pattern language* that involves conjunctions of closed intervals is $\langle m_1 \in [3, 4]$ AND $m_2 \in [3, 3]\rangle$, and the associated extent is $\{g_5, g_6\}$.

While *pattern flooding* – the exponential growth of the pattern search space as the number of attributes and attribute values increase – is a well-known problem with nominal data, it is even worse when it comes to numerical data. Indeed, given a set of M numerical attributes, the size of the search space of intervals Σ is given by:

$$|\Sigma| = \prod_{i \in \{1,\ldots,|M|\}} \frac{(|D_{m_i}| \times (|D_{m_i}| + 1))}{2}$$

For example, given the small toy dataset of Figure 2.1 (left), there are $10 \times 6 = 60$ possibles patterns. For larger datasets, it is easy to see how the number of patterns quickly becomes intractable.

	m_1	m_2	T
g_1	1	1	-
g_2	1	2	+
g_3	2	2	+
g_4	3	2	-
g_5	3	3	+
g_6	4	3	+

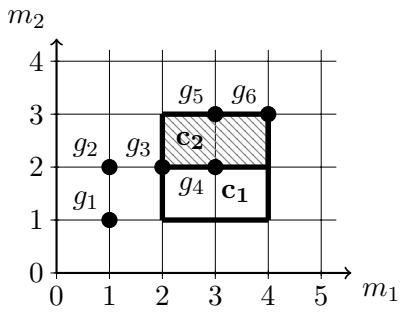

Figure 2.1: (**left**) Example of a numerical dataset involving 2 numerical attributes and a binary target. (**right**) Non-closed ($c_1 = \langle [2, 4], [1, 3]\rangle$, non-hatched) and closed ($c_2 = \langle [2, 4], [2, 3]\rangle$, hatched) interval patterns.

2.3.2 Numerical Attributes in Association Rule Mining

In Association Rule Mining (ARM), dealing with numerical attributes has seen numerous contributions throughout the years. The first occurrence of rules involving optimized intervals on numerical attributes dates back to 1996, with the introduction of *Quantitative Association Rules* in (Srikant and Agrawal, 1996). In this paper, the authors define a new type of rule where the antecedent can include intervals of discretized numerical attributes. The rules take the following form: $X \in [a, b] \implies Y$, where X and Y are attributes and $[a, b]$ is an interval of values of X. The discretization, however, leads to loss of information and possibly irrelevant rules.
(Miller and Yang, 1997) proposed an alternative quality measure that takes into account the quantitative properties of intervals to define more relevant cut-points for numerical attributes. They introduced a two-step approach. First, a clustering algorithm is applied to discover the proper discretization of intervals, and then a standard ARM algorithm can be applied. Using this method, the quality and relevance of rules are improved compared to (Srikant and Agrawal, 1996). (Zhang et al., 1997) also proposed an approach that employs clustering to improve interval-based ARM.
(Fukuda et al., 1996a) investigated the discovery of *optimized association rules* with numerical attributes. In their work, they detail a method for finding the optimal interval in the antecedent that leads to the highest rule quality (according to measures such as confidence and support). However, only one interval can be used in the antecedent of the rules. An extension of this method has been proposed in (Fukuda et al., 1996b) for rules with 2 intervals in the antecedent. (Brin et al., 2003) proposed an improvement over (Fukuda et al., 1996a,b). They propose to mine for optimized association rules using *disjunctions* of intervals. Interestingly, their approach allows for more than 2 intervals in the antecedent of the rules.
An evolutionary approach that allows for the discovery of all frequent patterns involving numerical attributes without a priori discretization was also investigated (Mata et al., 2002). The detailed evolutionary algorithm enables the discovery of the intervals of each numerical attribute that induce a frequent pattern. *QUANTMINER* (Salleb-Aouissi et al., 2007), a genetic algorithm for quantitative association rule mining was also introduced. The algorithm allows for the discovery of good intervals by finding optimized trade-offs between support and confidence.
Finally, an innovative approach that considers not intervals, but a weighted sum of numerical attributes as the antecedent of the rules was proposed in (Ruckert et al., 2004).

2.3.3 Numerical Attributes in Subgroup Discovery

Traditionally, subgroup discovery has been mainly concerned with nominal attributes and binary target labels. To handle numerical attributes, typical methods resort to the discretization of the numerical variables into nominal ones (Fayyad and Irani, 1993, Garcia et al., 2013), which inevitably leads to loss of information, suboptimal results, and even irrelevant patterns. As pre-discretization – also called offline discretization – of attributes could not offer satisfying results, online-discretization, which entails finding the best cut-points within the SD algorithm, was more recently investigated.
To avoid the use of discretization techniques, (Kralj et al., 2005, Lavrač and Gamberger, 2006) explored the use of a binarization scheme where each distinct value of each numerical

attribute is transformed into a boolean attribute. Doing this, it is then theoretically possible to apply an exhaustive search algorithm for nominal data, such as `SD-Map` (Atzmueller and Puppe, 2006a). Using such a technique, however, leads to a huge increase in the size of the search space, making the exhaustive discovery process intractable.
In (Nguyen and Vreeken, 2016), the authors propose a method for mining better subgroups in numerical data. They employ a binning strategy on each numerical attribute whose goal is the optimization of the average quality of the subgroups generated by said binning. The main advantages of the proposed model are that (i) by creating a discretization that seeks to maximize the average quality of the patterns, they obtain better overall subgroups (ii) the algorithm places no restriction on the target which can be univariate or multivariate, and handles numerical, nominal and binary data. Although the proposed approach allows for finding good numerical patterns for numerical targets, its non-exhaustive nature cannot guarantee the discovery of optimal solutions.
The following investigated methods deal with numerical attributes natively, i.e., without using discretization. The `BestInterval` algorithm was proposed (Mampaey et al., 2012, Mampaey et al., 2015) to compute the optimal direct specialization of a subgroup, given a numerical attribute to optimize. It enables the discovery of the optimal interval that maximizes the quality of the pattern. This is done efficiently by only considering the subgroup specializations that lie on the convex hull in ROC space. The procedure can be directly integrated in a standard algorithm, be it a greedy approach such as beam search or an exhaustive method. It is interesting to note that this only works for convex quality measures.
When it comes to exhaustive subgroup discovery in numerical domains, (Grosskreutz and Rüping, 2009) introduced an efficient algorithm, `MergeSD`, which makes use of an advanced new pruning scheme to optimize the search. No discretization is applied and overlapping intervals are considered, such that no information is lost. The authors detail new bounds on the quality of the specializations of a subgroup based on constraints proper to overlapping intervals. Furthermore, the authors provide an in-depth comparison of the results obtained using several commonly used discretization techniques, compared to the results obtained by applying an exhaustive search with `MergeSD`. In their results, they conclude that using either entropy discretization or frequency discretization with both overlapping and non-overlapping intervals leads to suboptimal results in most scenarios.
`MinIntChange` (Kaytoue et al., 2011) was proposed as a new framework for the comprehensive mining of numerical patterns with *Formal Concept Analysis* (FCA, (Ganter and Wille, 1998)). In FCA, an interval pattern represents a vector of intervals where each interval corresponds to the space of values taken by an attribute. The goal is then to compute the complete set of interval patterns efficiently. In their work, the authors exploit equivalence classes and closure operators to efficiently and exhaustively traverse the pattern search space.

Let us provide several definitions from (Kaytoue et al., 2011).

Definition 6. *Given a numerical dataset (G, M, T), an interval pattern p is a vector of intervals $p = \langle [b_i, c_i] \rangle_{i \in \{1,...,|M|\}}$ where $b_i, c_i \in D_{mi}$, each interval is a restriction on an attribute of M, and $|M|$ is the number of attributes.*

Next, we can consider the extent of an interval pattern.

Definition 7. *An object $g \in G$ is in the extent of an interval pattern $p = \langle [b_i, c_i] \rangle_{i \in \{1,...,|M|\}}$ iff $\forall i \in \{1, ..., |M|\}, m_i(g) \in [b_i, c_i]$.*

A specialization of an interval pattern is defined as follows.

Definition 8. *Let p_1 and p_2 be two interval patterns. $p_1 \subseteq p_2$ means that p_2 encloses p_1, i.e., the hyper-rectangle of p_1 is included in that of p_2. It is said that p_1 is a specialization of p_2.*

Finally, we can introduce the concept of closed interval pattern.

Definition 9. *Given an interval pattern p and its extent $ext(p)$, p is defined as closed if and only if it represents the most restrictive pattern (i.e., the smallest hyper-rectangle) that contains $ext(p)$.*

For example, in the toy dataset of Fig. 2.1 (left), the domain of m_1 is $\{1, 2, 3, 4\}$ and $\langle [2, 4], [1, 3] \rangle$ is the interval pattern that denotes a subgroup whose extent is $\{g_3, g_4, g_5, g_6\}$. Fig. 2.1 (right) depicts the same dataset in a cartesian plane as well as a comparison between a non-closed (c_1) and a closed (c_2) interval pattern.

Extending the work of both (Kaytoue et al., 2011) and (Garriga et al., 2008), (Guyet et al., 2017) introduced an algorithm that enables the extraction of closed-on-the-positives and relevant interval patterns for binary labeled data.
In (Bosc et al., 2018), a generic anytime algorithm based on Monte Carlo Tree Search (MCTS) for SD is introduced. It exploits a closure system, it can find diverse subgroups, and it is agnostic of the pattern language, enabling its use for SD in numerical domains without the need for prior discretization. However, the use of MCTS leads quickly to high memory usage, and no guarantee is provided on the optimality of the search on empirical data. Based on the interval pattern framework, (Belfodil et al., 2018) also proposed an anytime algorithm with guarantees to mine patterns in numerical domains with binary target variables. The use of an advanced closure scheme on interval patterns removes the need for discretization, such that the algorithm can run an exhaustive search if given enough time. However, no pruning strategy based on optimistic estimates is employed. It limits the efficiency of the search and its application to real problems.
In (Meeng et al., 2014), a heuristic ROC-guided algorithm for SD with numerical attributes without prior discretization is introduced. A main advantage of this method is the fact that contrary to typical beam search approaches, no parameter needs to be set. This is due to the fact that at each level of the search, the algorithm defines an ideal size for the width of the beam of the next level. When compared to typical beam search, it provides better results faster. This, combined with the fact the numerical attributes are treated natively makes it an attractive approach when exhaustiveness is not needed.
For a more exhaustive overview of numerical data in SD, the reader is invited to consult the recent survey (Meeng and Knobbe, 2021), which explains in detail the problems surrounding numerical SD, and provides a thorough comparison of existing methods.

2.4 Subgroup Discovery with Numerical Targets

Historically, SD has mostly been investigated for binary labeled data. For numerical targets, researchers made use of discretization techniques, that again inevitably lead to loss of information and suboptimal results. More than that, numerous real-life applications of SD involve numerical targets, and it would be useful that proper methods can treat the problem

natively. Fortunately, the interest for numerical data in SD seems to have picked up in the last few years, and several new contributions have been made. In this section, we propose an overview of SD with numerical targets.

2.4.1 Numerical Targets in Association Rule Mining

To understand numerical targets in ARM, we have to go back to its inception, with the introduction of *Quantitative Association Rules* in (Srikant and Agrawal, 1996). In their proposal, the authors define a new type of rule that can take the form $X \in [a, b] \implies Y \in [c, d]$, where X and Y are attributes and $[a, b]$ and $[c, d]$ are intervals of values of these attributes. While this type of rule does allow for the discovery of patterns with numerical intervals as consequent, said intervals are based on discretization, which is well-known for not only being suboptimal but can also lead to irrelevant rules. Furthermore, intervals make poor representatives of the distribution of numerical targets.
With the understanding that numerous problems cannot be solved using discretization, (Aumann and Lindell, 1999) extended the concept of *Quantitative Association Rules* by introducing a new rule concept where a rule consequent is the mean or the variance of a numerical attribute. A rule is then defined as interesting if its mean or variance significantly deviates from that of its complement. Two types of rules are defined: (i) *categorical to quantitative association rules* that involve a nominal antecedent and a statistical distribution over a numerical consequent, and (ii) *quantitative to quantitative association rules* where the antecedent corresponds to an interval of a numerical attribute, and the consequent is a statistical distribution over a numerical attribute. Furthermore, constraints on the support and confidence on the rule are used, such that said rules are in essence very close to what we consider nowadays as subgroup discovery.
Later on, (Webb, 2001) proposed an extension of such quantitative rules called *Impact Rules*. In this work, the author argues that measures based on statistical distribution might lack interest when one is looking to identify a group of objects with a large contribution with regard to the total of a given target. New aggregate measures are therefore designed, such as the sum of the values of the target in the subgroup.
Tight optimistic estimates for association rule mining with numerical targets were introduced in (Morishita and Sese, 2000). Several common convex interestingness measures, such as correlation and chi-squared are studied.

2.4.2 Subgroup Discovery Approaches

We now consider SD contributions that involve numerical targets. For a more exhaustive view of the subject, the reader is referred to both (Lemmerich, 2014) and (Pieters et al., 2010). Most typical SD methods employ standard discretization techniques when numerical targets are involved. In (Moreland and Truemper, 2009) the `TargetCluster` algorithm was introduced. It allows to find adequate cut-points for numerical target concepts using a clustering approach. The method is compared to a discretization technique that combines equal-width-intervals and equal-frequency-intervals, called EWF. The authors show that `TargetCluster` supports the discovery of better subgroups than using standard techniques. However, it still involves discretization, and therefore does not give a proper answer to SD with numerical targets.

The Explora system (Klösgen, 1996) introduced the first SD algorithm that enables the discovery of subgroups with numerical targets without prior discretization. Explora allows for the discovery of subgroups with a mean that significantly deviates from the overall dataset.
(Grosskreutz, 2008) introduced an iterative method for diverse subgroup set discovery with a numerical target. They make use of a framework that combines standard SD and rule-based regression to build a prediction model for the target value of the subgroups. In each iteration, they look for the best subgroup in the subset of the overall data where the prediction currently deviates the most from the real target value. They show that the resulting subgroup set possesses better diversity than using an exhaustive search, with or without the use of condensed representations.
Later on, the *SD-MAP⋆* algorithm (Atzmueller and Lemmerich, 2009) was introduced as an exhaustive subgroup discovery method for numerical targets. The algorithm takes advantage of optimistic estimate pruning, using new tight optimistic estimates for well-known measures such as *Continuous Piatetsky-Shapiro*, *Continuous LIFT*, and *Continuous Weighted Relative Accuracy*. This work has been significantly extended in both (Lemmerich et al., 2016a) and (Lemmerich, 2014). (Lemmerich, 2014) introduces a new algorithm for exhaustive SD with numerical targets, called `NumBSD`, an adaptation of the *BSD* algorithm (Lemmerich et al., 2010) used for SD with binary targets. It employs a special bitset-based data structure that allows for the fast discovery of subgroups. Numerous new quality measures and corresponding optimistic estimates are also introduced, including mean-based, variance-based, median-based, rank-based, and distribution-based measures. It however only works with nominal attributes, and numerical attributes have to be pre-discretized which limits its usability in real-life settings.
A new quality measure and corresponding tight optimistic estimate to improve existing quality measures was introduced in (Boley et al., 2017). In their work, the authors argue that current measures lead to unreliable results since the variance is not optimized when looking for high-quality subgroups. A branch-and-bound algorithm that exploits the proposed tight estimator is developed and shown to be very efficient. However, it is only applicable to median-based metrics, while most use cases involve other types of quality measures, e.g., based on the mean and/or the variance.
A more generic approach to SD with numerical targets is proposed in (Lijffijt et al., 2018). The authors develop a method for SD whose goal is to discover subjectively interesting patterns, i.e., that are interesting according to the knowledge of an expert. To do this, a background distribution of the numerical target is generated using expert knowledge. The goal is then to look for subgroups that maximize the information gained compared to this subjective distribution. Two types of subgroups are mined using this method: *location* patterns, which look for subgroups of objects whose statistical distributions significantly deviate from the background distribution, and *spread* patterns, which exploit each discovered significant *location* pattern to look for exceptional dispersion around its statistical distribution. For example, given a *location* pattern whose mean significantly deviates from expert knowledge, it could also be defined as a *spread* pattern if its variance is somehow exceptional. The proposed algorithms have been shown to be both effective and efficient for SD. It is interesting to note that this framework is also extended to the case where multiple numerical targets exist, i.e., Exceptional Model Mining.
A Minimum Description Length (MDL) approach to SD with numerical targets has also been proposed in (Proença et al., 2021). Their goal is to discover a set of diverse patterns, called

subgroup list that when combined, offer a good overall representation of the distribution of the numerical target over the whole dataset. A new quality measure that maximizes the Sum of Weighted Kullback-Leibler divergences is introduced. It allows for the discovery of subgroups whose mean significantly deviate from the norm while keeping the dispersion of the objects low. Finally, a new greedy algorithm based on beam search, called `SSD++`, is presented and shown to achieve better performance than existing methods.
Recently, (Meeng et al., 2020) introduced a new type of interestingness measure for numerical targets. The authors explain that using simple statistical measures such as the mean or the variance is inadequate, and that interesting subgroups can be missed. They argue for the use of probability density models – using techniques such as kernel density estimation and histograms – to discover more diverse types of deviations in the distribution of the targets.

2.5 Optimal Subgroup Discovery

Although most subgroup discovery methods support the discovery of a set of high-quality patterns, algorithms that can discover the optimal subgroup with respect to a quality measure or the proper set of top-K optimal subgroups are rare. Optimal SD necessarily implies the use of exhaustive enumeration strategies, since other strategies, i.e., heuristic ones including sampling-based and anytime ones provide no guarantee on the quality of the results.
Mining optimal patterns has been investigated in the past for association rule mining with numeric attributes. The proper term in the domain is *optimized association rules*, and it consists in finding a rule that contains one or a set of numerical attributes as antecedent, and which optimizes a target quality measure, such as confidence, support, or gain.
Mining optimal subgroups has also been investigated, although unevenly depending on the type of data considered. We first formally define the concept of optimal subgroup.

Definition 10. *Let (G, M, T) be a dataset, q a quality measure and P the set of all subgroups of (G, M, T). A subgroup is said to be optimal iff $\forall p' \in P : q(p') \leqslant q(p)$.*

Notice that several subgroups can have the same optimal quality. In such situations, it is up to the user to find a way to determine which subgroup(s) is (are) more suited to his needs.

Nominal attributes with binary targets. We first review contributions that find optimal subgroups in nominal data, i.e., data made of nominal attributes and a binary target concept. This is the area that has seen the most work done for Optimal Subgroup Discovery, probably due to the fact that nominal data has by far been the most studied setting for SD. (Garriga et al., 2008) was the first to introduce the concept of close-on-the-positives and the theory of relevance for subgroup discovery in labeled data. Using this closure system, an exhaustive search can be applied to return the optimal subgroup. Cluster-grouping (Zimmermann and De Raedt, 2009) exploits pruning techniques through optimistic estimates in a branch-and-bound algorithm that allows for the discovery of the optimal subgroup in nominal data. The popular `SD-Map` algorithm (Atzmueller and Puppe, 2006a) employs an exhaustive enumeration strategy, made possible by the use of an optimized data structure (FP-trees), with a guarantee to discover the optimal subgroup. (Li et al., 2014) can provide a guarantee on the discovery of the statistically non-redundant optimal subgroup, although

only in nominal data. The optimality of the search is ensured by the use of a pruning scheme based on the interestingness measure considered. Finally, the concept of *relevance*, as in (Grosskreutz and Paurat, 2011, Guyet et al., 2017, Lemmerich et al., 2010) can also allow for the guaranteed discovery of the optimal subgroup.

Nominal attributes with numerical targets. We now review proposals made for Optimal Subgroup Discovery in data made of nominal attributes, and a numerical target. Contributions in this setting have been relatively rare. The `SD-Map*` algorithm (Lemmerich et al., 2016a), an improved version of `SD-Map` is perhaps the most well-known that can handle numerical targets with an exhaustive strategy. Using the same data structure as `SD-Map`, the authors also employ advanced pruning techniques based on tight optimistic estimates to make the search tractable. However, only nominal attributes can be handled, meaning that numerical attributes have to be discretized. Building on the work from (Lemmerich et al., 2016a), (Boley et al., 2017) develop a new class of quality measures and corresponding tight optimistic estimates for numerical targets. The authors argue that current quality measures are insufficient since they do not optimize for the error or dispersion of the subgroups. Using this new scheme within a branch-and-bound algorithm, the authors guarantee the discovery of the optimal subgroup with regard to the proposed quality measure.

Numerical attributes with binary targets. Although several contributions have been made for subgroup discovery with numerical attributes, very few provide a guarantee on the optimality of the search. (Meeng et al., 2014) introduced a ROC-guided algorithm for subgroup discovery with numerical attributes. However, the optimality guaranteed by their method is not based on the quality of the subgroups, but on a minimized cost regarding a cost assignment for false positive and false negative objects. Therefore, we can not consider their contribution as being able to provide the optimal subgroup. `MergeSD` (Grosskreutz and Rüping, 2009) allows for an exhaustive search in numerical data. Overlapping intervals are considered without pre-discretization, such that exhaustiveness can be guaranteed. This is made possible by introducing and exploiting new advanced pruning techniques. Finally, the `Refine&Mine` algorithm (Belfodil et al., 2018), although based on an anytime strategy, can return the optimal subgroup, provided that enough time is given to the algorithm to converge. However, as it was not created with the specific goal of discovering an optimal subgroup, pruning strategies are not exploited.

Numerical attributes with numerical targets. The investigation of purely numerical data in SD has been close to non-existent. The few contributions that consider numerical targets either ignore the case of numerical attributes or employ discretization technique to make the search tractable. `MCTS4DM` (Bosc et al., 2018) is, to our knowledge, the only known algorithm that theoretically enables the discovery of optimal subgroups in purely numerical data without prior discretization, due to its agnosticism toward both the used pattern language and quality measure. This is also only possible if a large enough time budget is given so that it produces an exhaustive search. However, there is no guarantee that the search will be complete in a finished amount of time, even for small datasets, as the algorithm was not built for exhaustive search and lacks advanced pruning and compression strategies. Furthermore, `MCTS4DM` is limited by its high memory usage, which is problematic

for exhaustive exploration. To the best of our knowledge, there is therefore a research gap that needs to be filled regarding Optimal Subgroup Discovery in purely numerical data when no discretization techniques are allowed.

2.6 Overview of Exceptional Model Mining

2.6.1 A Generalization of Subgroup Discovery

Exceptional Model Mining (EMM) was introduced over 10 years ago in (Leman et al., 2008) as a generalization of subgroup discovery for problems involving multiple targets. In subgroup discovery, we have only one target. The quality of a subgroup is usually defined as the discrepancy between the distribution of the target variable in the subgroup and its distribution over the entire dataset. Exceptional Model Mining enables for two or more target variables depending on the chosen model class. A model class can be any mathematical model that involves and measures complex interactions between a set of targets. In EMM, a dataset (G, M, T) is a set of objects G, a set of attributes M and a set of targets T. In a given dataset, the set of attributes M and the set of targets T contain real and nominal values. Table 2.2 depicts an example of dataset used in EMM made of 5 descriptive attributes (nominal or real) and 2 numerical targets.

Table 2.2: Example of a dataset with 5 attributes (nominal and real) and 2 numerical targets.

	m_1	m_2	m_3	m_4	m_5	t_1	t_2
g_1	1.4	A	F	3	0	0.5	1200
g_2	5.6	B	G	6	1	0.3	400
g_3	10.2	A	H	8	0	0.75	2560
g_4	7.3	C	H	7	1	0.97	1812
g_5	9.4	D	H	2	1	0.15	727

Most definitions that hold for SD also hold for EMM, i.e., *pattern language*, *intent*, *extent*, and *specialization*. It is important to note that since a given model class only involves the target variables, pattern languages that can be used for EMM are the same as those used for SD. In the standard EMM setting, the interestingness of a subgroup is measured by a numerical value that quantifies the deviation between the model fitted on the subgroup and the model fitted on another subset of the data. There are usually two options about the subset that is chosen for comparison: we can compare the model of the subgroup either to the model of its complement or to the model of the whole dataset. Choosing one or the other can lead to very different results and may depend on the considered application setting. In (Duivesteijn et al., 2016), the authors show that there is not always a clear-cut best solution for which subsets to compare, and that several parameters should be taken into account by the Exceptional Model Miner before making a choice.
From an algorithmic perspective, the search space of subgroups for most EMM algorithms is traversed in a general to specific way. At each stage, a specialization operator is applied to create more complex subgroups by addition of a restriction on an attribute. Since EMM is still a fairly recent addition to the pattern mining field of study, relatively few works have

been introduced so far, although its popularity seems to have picked up in the last couple of years.

2.6.2 Enumeration Strategies

While EMM is still in its infancy, several heuristic and exhaustive, methods have been developed. We investigate the several enumeration strategies which have been introduced to make EMM efficient. We first explore heuristic proposals.
In the introduction of EMM (Leman et al., 2008), the researchers proposed the use of a standard beam search strategy, an algorithm that performs a level-wise exploration of search space. A standard beam search possesses two main parameters: a maximum depth of exploration d (i.e., the maximal number of restrictions in the description of a subgroup), and a beam-width w (i.e., the number of subgroups specialized at each level). In their strategy, the authors run the search starting from the most general pattern and apply a specialization operator to generate the candidates of the lower levels. At each level, the best w subgroups according to the chosen quality measure are stored to be specialized in the next level. During the whole search, the overall top-K best subgroups are also stored and updated when better subgroups are discovered. The search stops once the level d of exploration is reached, and the best subgroups are returned. In (Duivesteijn et al., 2016), the authors provide a more detailed version of this strategy, that they name *Beam Search for Top-q Exceptional Model Mining*. Since then, beam search has become the most common strategy for EMM. It is interesting to note that although beam search for EMM is an interesting heuristic, it provides no guarantee on the discovery of the optimal subgroup.
The use of a new heuristic strategy called *Tree-Constrained Gradient Ascent* (TCGA) to mine for exceptional models is developed in (Krak and Feelders, 2015). A goal of TCGA is to find relevant and exploitable information about the influence of a single object on the quality of a subgroup. To do that, they rewrite the quality measure as an objective function to be optimized. They transform the notion of subgroup into fuzzy subgroup by creating a concept of *inclusion weight* for each object of a given extent. Then, using a numerical optimization technique – gradient ascent – they find the locally optimal extent that optimizes the objective. The weights of each object of the extent are then rounded to obtain a crisp extent. The next step is to discover the subgroup description from the extent. To ensure that concise descriptions can be extracted from extents, the numerical optimization step is modified, leading to the introduction of a constrained gradient ascent method. The relevance of TCGA is studied in-depth on synthetic and real-life data for linear regression EMM as well as for typical SD. On synthetic data, TCGA is found to be superior to beam search. However, on real-life data, TCGA performs as well as beam search for EMM, and way worse than beam search for SD.
In (Lemmerich et al., 2012), the authors introduce the first method for fast exhaustive EMM, titled *Generic Pattern Growth* (GP-growth). In this work, the well-known concept of FP-tree (Han et al., 2000) is extended to mine for exceptional models. To do this, a new concept called *valuation basis* is presented, which replaces the original frequency data used in typical FP-trees. A *valuation basis* consists of the minimal amount of information about a set of objects needed to compute the model corresponding to the considered model class. For example, given the simplistic mean model with one target variable, a valuation basis could involve (i) the number of objects, (ii) the sum of the values of the target variable of the

objects considered. Using only this information, the mean target value of the objects can be reconstructed, and a metric that measures the discrepancy between this mean and that of the entire dataset can be computed. The same goes for more complex model classes. In order for GP-growth to be efficient, valuation bases have to be as small a possible. It is interesting to note that GP-growth for EMM is a generalization of both FP-growth and `SD-Map`, as these algorithms can be implemented with GP-growth by simply using the corresponding valuation basis. Among the contributions, several valuation bases for well-known EMM model classes – such as variance, correlation, and linear regression – are also detailed. In the empirical study, the superiority of GP-growth compared to a naive exhaustive search algorithm is confirmed. Finally, we investigate the use of weighted controlled pattern sampling for instant EMM proposed by (Moens and Boley, 2014). In their work, the authors argue that interactive discovery is necessary to make pattern discovery more relevant and actionable to users. For interactive discovery, heuristic and exhaustive approaches are usually too slow, justifying the need for algorithms that can discover high-quality patterns instantly. In this paper, Controlled Direct Pattern Sampling (CDPS) (Boley et al., 2012) is extended by applying a utility weight to each object of the dataset. Using these weights, the notion of *weighted frequency* – the relative total weight of a pattern compared to the total weight of the dataset – can be computed. Then, using a predefined distribution that gives high generation probability to patterns with high *weighted frequency* in their positives objects (or to other subsets of the data depending on the model class and the definition of interestingness considered), random patterns can be sampled from the search space. By exploiting this method, subgroups with high generalization and whose models deviate significantly from the global model can be discovered almost instantly. In their experiments, the authors confirm the relevance of their approach for instant discovery of subgroups whose quality is close to that of a beam search strategy.

2.6.3 Model Classes

In EMM, each problem to be solved relates to a particular model class. Indeed, if we consider the two following problems: mining for exceptional correlations and mining for exceptional Bayesian networks, each problem needs its own approach and quality measures to be solved. The notion of model class has been introduced in (Leman et al., 2008).
In this paper, 3 classes of models are presented as a basis to justify the relevance of EMM. First, the correlation model class is introduced, for which the authors consider 2 numerical variables and their linear association according to their correlation coefficient. We now detail the correlation model and 2 of its quality measures for a better understanding of the EMM framework. The objective is to estimate the deviation between the correlation of a given subgroup p, and the correlation of its complement. Given the two numerical targets t_1 and t_2, the correlation coefficient is estimated by the sample correlation coefficient r as follows:

$$r = \frac{\sum(t_1^i - \bar{t}_1)(t_2^i - \bar{t}_2)}{\sqrt{\sum(t_1^i - \bar{t}_1)^2 \sum(t_2^i - \bar{t}_2)^2}}$$

with t^i the i^{th} object of t and $\bar{t}$ the mean of t.
A first simple quality measure that can be defined is the absolute difference between the correlation of the subgroup, denoted G, and its complement, denoted $\bar{G}$. Therefore, we have:

$$q_{abs}(p) = |r_G - r_{\bar{G}}|$$

This measure however does not take into account the generalization of the subgroup. Consequently, small subgroups whose correlation can easily deviate from the norm would be given a high quality. To resolve this issue, a measure that involves the entropy of the split between the subgroup and its complement can be used (Leman et al., 2008).

Definition 11. *The entropy of a subgroup p is:*

$$Entropy(p) = \left(-\frac{n}{N} \lg\left(\frac{n}{N}\right) - \frac{N-n}{N} \lg\left(\frac{N-n}{N}\right) \right)$$

where lg *denotes the binary logarithm, n the number of objects of p, and N the number of objects of its complement.*

The entropy favors balanced splits over unbalanced ones. It returns 0 when the subgroup or its complement is empty. It returns 1 when a perfect 50/50 split is achieved. Notice however that it introduces a bias against subgroups with a large cover. The improved quality measure is therefore as follows:

$$q_{ent}(p) = Entropy(p) \times |r_G - r_{\bar{G}}|$$

Using any of these 2 quality measures and the model class defined, a standard EMM algorithm can then easily be used to mine for exceptional correlation models.
Next, a model class for regression problems is investigated. In their work, the authors consider the simple linear regression model described by $y_i = a + bx_i + c_i$ and introduce a metric that measures the significance of the slope difference between the model fitted on the subgroup, and the model fitted on its complement.
A model class for classification models is also explored. Although EMM allows for any complex method, only 2 simple classifiers are considered: Logistic Regression and Decision Table Majority (DTM) Classifier. For both classification methods, an appropriate quality measure is detailed.

For a recent and up-to-date introduction to Exceptional Model Mining, the reader is referred to (Duivesteijn et al., 2016).
After the introduction of the EMM framework, researchers started working on more complex problems than what had been done until then, when subgroup discovery involving a single target was the only tool available. In (Duivesteijn et al., 2010), the discovery of exceptional Bayesian networks is investigated. The authors argue that when dealing with datasets with several discrete targets, studying their interdependencies is an interesting task. To do this, the interdependency relationship is modeled by Bayesian networks. They look for subgroups whose network structure is significantly different from the structure of the model over the entire dataset. A quality measure based on edit distance is designed to discover those exceptional models. The relevance of their approach is verified statistically on several datasets from different domains.
In (Duivesteijn et al., 2012a), the authors take on what they call the "workhorse" of data analysis problems, namely Linear Regression. They introduce a new model class for exceptional regression model mining thanks to a quality measure based on Cook's distance. They also exploit interesting bounds to avoid computing the model on unfit subgroups. Model classes for classification problems have also been explored in (Duivesteijn and Thaele, 2014) and (Duivesteijn et al., 2012b). In the first approach, the authors look for subspaces of the search

space where a given classifier performs particularly well or badly, giving the user insights on which parts of their classifier they must focus on. In the second approach, the authors propose a method for identifying and exploiting exceptional interdependencies between labels in a multi-label classification setting, allowing them to improve the classifier overall quality.
In 2016, (Lemmerich et al., 2016b) introduced a new EMM class exploiting first-order Markov chains to mine for exceptional transition behavior in sequential data. Discovering deviating models can be useful, for example on mobility and internet user data. A proper quality measure adapted to the model class is detailed, and the applicability of the approach is studied on synthetic and empirical data. Exceptional Preferences Mining (EPM) (de Sá et al., 2016) was introduced as a cross-fertilization between EMM and preference learning. In EPM, they look for subgroups whose preference relations significantly deviate from the norm, using a specialized quality measure. (Luna et al., 2016) formalizes the concept of Exceptional Relationship Mining (ERM) and details a grammar-guided genetic programming algorithm to discover such models. The goal of ERM is to discover any kind of exceptional relationship between a set of variables. In their empirical study, they look for exceptional relationships between several quality measures used in association rule mining. Interestingly, they find that under some constraints, the support and leverage measures are negatively correlated, which goes against expert knowledge.
The discovery of exceptional correlations has also been investigated more in-depth in (Downar and Duivesteijn, 2017), (Hammal et al., 2019) and (Luna et al., 2020). In (Downar and Duivesteijn, 2017), the authors mine for exceptional monotone relations between two predefined targets in terms of rank correlation. The work of (Hammal et al., 2019) can be seen as an extension of (Downar and Duivesteijn, 2017), which generalizes the discovery of exceptional rank correlations to any number of targets. In (Luna et al., 2020), the authors observe that current EMM proposals only consider the discovery of one exceptional behavior for a given subgroup. This leads to the question of whether finding subgroups with multiple occurrences of exceptional behavior is possible. In this work, a first answer is given with the introduction of the Subsets of Pairwise Exceptional Correlations (SPEC) model class. In SPEC, a subgroup is deemed exceptional if multiple pairs of target concepts show exceptional rank correlation behavior. Since typical EMM algorithms can not be exploited for SPEC, the authors also introduce several heuristic and exhaustive search strategies.
In (Belfodil et al., 2017), the discovery of exceptional pairwise behavior in voting and rating data is investigated. For example, there is usually a clear difference of position between far-left and far-right political parties on most issues. However, for some issues, these political parties might present the same behavior, which can be reflected in voting data. In their approach, the authors look for such exceptional behavior. To that end, the Discovering Similarities Changes method and its corresponding quality measure are introduced and validated on European parliament votes and collaborative movie reviews. Following this work, (Belfodil et al., 2019b) detailed a new method for the discovery of statistically significant exceptional agreements or disagreements within groups. The `DEvIANT` branch-and-bound algorithm is introduced, which leverages several techniques for efficiency optimization, such as closure operators, optimistic estimates, and confidence intervals.
Recently, we also find proposals about the discovery of exceptional models with real-valued targets (Lijffijt et al., 2018), exceptional descriptions of people (Hendrickson et al., 2018), exceptional mediation models (Lemmerich et al., 2020), and exceptional spatio-temporal behavior (Du et al., 2020).

2.7 Conclusion

In this chapter, we investigated the literature of SD, a pattern discovery task that was first introduced 25 years ago. We first gave a full overview of SD, including its relationship to other mining tasks, its formalization, and the many contributions w.r.t. its different components.
As part of this thesis is focused on SD in purely numerical data, we investigated in detail both SD with numerical attributes and SD with numerical targets. SD with numerical attributes has historically been of relatively low interest for researchers, with few contributions existing in the literature. Fortunately, the study of numerical attributes has been receiving more attention for a few years now. When considering numerical attributes, relatively few approaches propose proper strategies that do not rely on discretization techniques. Therefore, methods for treating numerical attributes natively will likely be of interest to researchers in the near future. SD with numerical targets has also seen sparse contributions. This is problematic since many real-life scenarios involve numerical objectives, further demonstrating the need for proper techniques that avoid loss of information.
As we are interested in discovering optimal parameter values for optimization problems, algorithms that allow for the discovery of an optimal subgroup are highly relevant to us. For this reason, we reviewed Optimal SD in different types of data. While exhaustive approaches are relatively numerous for nominal data, numerical domains once again fall short of what would be expected, given the pervasiveness of numerical data nowadays. Indeed for SD in data with numerical attributes and a binary label, we found only 2 methods that allow for an exhaustive search, and both employ suboptimal techniques for search space compression and pruning. For SD in purely numerical data (i.e., numerical attributes and numerical target), there is currently no approach that has proved empirically its ability to discover an optimal subgroup. The only existing method, `MCTS4DM`, can only find an optimal subgroup in principle. Indeed, the drawbacks of the method (i.e., high memory usage, lack of pruning, and optimized compression scheme) would likely render the search intractable even for small datasets.
We also investigated EMM, a generalized framework for SD with an undefined number of targets. Few contributions have been made to the field, especially in the first few years of the previous decade. Fortunately, more and more approaches are being introduced and its interest seems to have increased recently.

Let us now imagine a setting where we have at hand a purely numerical dataset – i.e., made of a set of numerical attributes, and one or several numerical targets – and we want to find the attribute values that optimize the target(s). In this setting, using SD or EMM – depending on the number of targets – is extremely relevant. Indeed, the description of the best subgroups could provide interesting and actionable information regarding the attribute values that lead to optimized targets.
For SD (i.e., a unique target to optimize), discovering an optimal subgroup would be even more relevant than discovering the top-K subgroups with no optimality guarantee. There is currently no efficient algorithm that support the discovery of the optimal subgroup for this kind of problem. For EMM (i.e., a set of targets to optimize simultaneously), discovering subgroups that provide actionable information on the attribute values that lead to optimal trade-offs between several targets is a problem that has not been investigated yet. These

observations motivate our contributions in Chapter 4 and Chapter 5.

Chapter 3

Overview of Multi-Objective Optimization

In this chapter, we propose an overview of Multi-objective Optimization (MOO). We aim to show the limits of existing methods in our context and to motivate why a combination of Pattern Discovery and Multi-objective Optimization concepts could be appealing. We first review classical approaches, which involve an a priori definition of the importance of each objective. Pareto-based Multi-objective Optimization – a framework that does not require knowledge about the importance of the objectives – is then discussed. The literature of quality evaluation methods is studied, and existing benchmarks, tools, and application cases are detailed. Finally, cross-fertilization between Pattern Discovery and Multi-objective optimization is considered.

3.1 Introduction

Multi-objective optimization (Deb, 2014) is a sub-field of Multi-criteria Decision Making (Chankong and Haimes, 2008, Tzeng and Huang, 2011, Zeleny, 2012) that is focused on finding globally optimal solutions for real-life problems that involve a set of usually conflicting objectives. For simple problems, applying methods that transform the multi-objective optimization problem into a single-objective one is often enough, and yields a single globally optimal solution according to the applied scalarization method. When dealing with less trivial scenarios, scalarization techniques lead to sub-optimal results, and the use of proper MOO methods that yield not one, but a set of Pareto optimal solutions is needed.
Inspired by nature and based on concepts from the theory of evolution (Eiben and Smith, 2015), evolutionary algorithms, and more precisely genetic algorithms represent the most widely used methods in MOO. As global optimization techniques (Törn and Zilinskas, 1989), genetic algorithms are driven to converge toward global solutions, rather than local ones. Therefore, in the MOO setting, genetic algorithms aim at discovering the set of globally optimal solutions, i.e., the globally optimal trade-offs between the considered objectives.
Reinforcement Learning (Sutton and Barto, 2018, Szepesvári, 2010) is a closely related field that aims at optimizing a long-term objective through sequential decision making. In cases where multiple long-term objectives have to be optimized simultaneously, Multiobjective Reinforcement Learning (Liu et al., 2014, Wang and Sebag, 2012) that often exploits Pareto-based concepts is used.
Another related area of research is optimal design (Pukelsheim, 2006, Silvey, 2013), which is concerned with finding the set of optimal parameter values for designing a given experiment, process, or product. In optimal design, several possibly conflicting parameters have to be considered and optimized at the same time to obtain the optimal output.
Finally, active learning (Ienco et al., 2013, Settles, 2012) is another close field of research. In active learning, a learning algorithm involves the user in an interactive way to label data to optimize the quality and reduce the number of experiments needed to attain good results. This relates to interactive MOO methods, where user preference is taken into account after each iteration to converge faster toward optimal solutions.
In this thesis, we consider an MOO setting where there is a need to discover relevant information about a set of descriptive attributes when several numerical targets have to be optimized at the same time. The corresponding application scenario that motivates this research is the design of better plant growth recipes.
Plant growth optimization is an intrinsic MOO problem. Indeed, when trying to optimize the yield, the size, or the taste of plants, other parameters like the energy cost have to be considered, especially in controlled environments. Therefore, optimizing plant growth means finding the best trade-offs between several competing objectives. This is a difficult task: when optimizing recipes, the underlying model is unknown and experiments are limited due to time and cost constraints, making it impossible to exploit typical MOO approaches. There is a need for methods that would support the discovery of relevant and exploitable information in such MOO problems.
A first intuition is to cross-fertilize Pattern Discovery methods and MOO-related concepts. Indeed, Pattern Discovery could be exploited to discover interesting insights about the descriptive attributes, while MOO could be leveraged to optimize the numerical targets simultaneously.

We now propose an overview of MOO, that provides important information regarding the relevance and actionability of existing methods to solve our problem. The remaining of this chapter is organized as follows. In Section 3.2, we review classical approaches to MOO. Pareto-based MOO is formalized and numerous approaches are reviewed in Section 3.3. The literature of quality evaluation for MOO is detailed in Section 3.4. In Section 3.5, we discuss benchmarks, tools and applications for MOO. Cross-fertilization between Pattern Discovery and MOO is studied in Section 3.6. Finally, Section 3.7 concludes.

3.2 Classical Approaches

Early approaches to solving multi-objective optimization problems involved a priori definition of user preferences with regard to the importance of each objective function. Due to a lack of proper optimization methodology (Deb, 2014), these methods rely on transforming multi-objective problems into single-objective ones. They are therefore only able to find a single globally optimal solution according to the a priori preferences.
The *Weighted global criterion method* (Yu, 1974) is the most common generic scalarization technique that is used to solve MOO problems. In this approach, the set of objectives functions is combined into a single objective to optimize using a scalarizing function. A popular method is the *Weighted sum method* (Zadeh, 1963), which creates a single aggregate objective function by assigning a weight to each objective according to user preferences and then computing the sum of these weighted objectives. Formally, we have:

$$WS = \sum_{i=1}^{n} w_i F_i(x)$$

with n the number of objectives.
Among other approaches with an a priori definition of preferences, we also find the *Exponential weighted criterion* (Athan and Papalambros, 1996) – introduced to solve some of the *Weighted sum method* shortcomings – the *Weighted product* (Bridgman, 1922), and the *Bounded objective function method* (Ching-Lai and Abu, 1979, Haimes, 1971).
The *Lexicographic method* (Stadler, 1988) requires for an order of importance to be defined a priori on the set of objective functions. Contrary to scalarizing methods, assigning a precise weight for each objective is not needed here. *Goal programming* was also introduced in (Charnes et al., 1955) where a target value to be reached is assigned to each objective function. The algorithm then looks to minimize the total deviation from these goals. In 1996, (Messac, 1996) introduced *Physical programming* where each objective function is divided into a set of degrees of desirability according to user preferences. For example, for a given objective, a user would define which ranges of values are unacceptable, undesirable, tolerable, and desirable. This approach has several advantages over scalarizing methods: (i) it removes the need to define proper weights for each objective, (ii) the quantity of a priori knowledge required is greatly reduced, (iii) setting degrees of desirability is more natural to the user than choosing often improper weights.
There are however several major issues with these classical approaches: (i) the preferences have to be known a priori, which requires a deep knowledge of the problem at hand, (ii) if a diverse set of optimal solutions is required, multiple iterations of an algorithm with different sets of parameters are needed, (iii) for more than one solution, these methods provide

no guarantee on the discovery of any globally optimal solution, (iv) the optimal solutions discovered are biased by a priori preferences, which can lead to sub-optimal, or even bad solutions.

3.3 Pareto-based Multi-Objective Optimization

3.3.1 Concepts

Many real-world optimization problems are intrinsically multi-objective.

Definition 12. *A multi-objective optimization problem can be defined as follows:*

$$Minimize\ F(x) = (f_1(x), ..., f_n(x))^T, x \in M$$

where M is the attribute space and x is an attribute vector. $F(x)$ consists of n objective functions $f_i : M \to \mathbb{R}, i \in \{1, \ldots, n\}$, where $\mathbb{R}^n$ is the objective space. In terms of an available dataset (G, M, T), the objective functions correspond to the targets in T.

The objectives usually conflict with each other and the improvement of one objective might lead to a degradation for others. For this reason, we lack a single solution that enables the optimization of all objectives at the same time. When no order or relevance can be defined a priori on the different objectives, a Pareto-based optimization method is required. It is based on the dominance between solutions of the objective space.

The *weak dominance* relation can be defined as follows.

Definition 13. *A vector $a = (a_1, \ldots, a_n)^T$ weakly dominates a vector $b = (b_1, \ldots, b_n)^T$, denoted $a \preceq b$ if and only if $\forall i \in \{1, \ldots, n\},\ a_i \leqslant b_i$ and $a \neq b$.*

The *dominance* relation, which is most commonly used in the literature, can be defined in the following way.

Definition 14. *A vector $a = (a_1, \ldots, a_n)^T$ dominates a vector $b = (b_1, \ldots, b_n)^T$, denoted $a \prec b$ if and only if $\forall i \in \{1, \ldots, n\}, a_i \leqslant b_i$ and $\exists i \in \{1, \ldots, n\}, a_i < b_i$.*

Finally, the *strict dominance* relation is as follows.

Definition 15. *A vector $a = (a_1, \ldots, a_n)^T$ strictly dominates a vector $b = (b_1, \ldots, b_n)^T$, denoted $a \prec\prec b$ if and only if $\forall i \in \{1, \ldots, n\}, a_i < b_i$.*

A non-dominated solution is called *Pareto optimal* (Pareto, 1906).

Definition 16. *A solution x is called Pareto optimal if and only if $\nexists y \in M$ such that $F(y) \prec F(x)$.*

Definition 17. *The set of all Pareto optimal solutions is called the (true) Pareto Front:*

$$PF = \{F(x) | x \in M | \nexists y \in M, F(y) \prec F(x)\}$$

Numerous test functions for multi-objective algorithms have been proposed in the literature. The true Pareto front of these functions is usually known and they are designed such that Pareto front approximation by algorithms is difficult. To illustrate our work and its related concepts, we consider the Fonseca-Fleming function (Fonseca and Fleming, 1995) that implies 3 descriptive variables from $\{x_1, x_2, x_3\}$ and 2 objectives. It is described by functions f_1 and f_2 that both need to be minimized:

$$f_1(p) = 1 - exp\left(-\sum_{i=1}^{3}\left(x_i - \frac{1}{\sqrt{3}}\right)\right), x_i \in [-4, 4]$$

$$f_2(p) = 1 - exp\left(-\sum_{i=1}^{3}\left(x_i + \frac{1}{\sqrt{3}}\right)\right), x_i \in [-4, 4]$$

We generate 5000 random objects using the Fonseca-Fleming function – that we name `Fonseca` – and we retrieve the true Pareto front of the function – which is composed of 434 objects – from jMetal[1] (Durillo and Nebro, 2011). Table 3.1 provides a toy dataset that is a subset of `Fonseca`.

Table 3.1: Toy dataset related to the Fonseca-Fleming function.

	x_1	x_2	x_3	f_1	f_2
g_1	-3.48	2.57	-0.12	0.99	0.99
g_2	-1.94	-0.24	-1.05	0.99	0.89
g_3	0.38	-2.09	0.99	0.99	0.99
g_4	0.39	0.54	0.34	0.09	0.95
g_5	-0.28	-0.09	-1.35	0.99	0.60

Figure 3.1 depicts the Pareto front (i.e., the non-dominated solutions) of the Fonseca-Fleming function. The dominance relation can be illustrated using Figure 3.1. Indeed, in the figure we can see that $A \prec B$ (i.e., object A strictly dominates object B) since $f_1(A) < f_1(B)$ and $f_2(A) < f_2(B)$.
It is interesting to note that very few points lie close to the true Pareto front in Figure 3.1. This is due to (i) the Pareto front of the Fonseca-Fleming function being hard to approximate, (ii) the random sampling method used to generate the data points, which is sub-optimal to discover optimal points for MOO problems.

3.3.2 Evolutionary Approaches

While some MOO problems can be solved by transforming them into single-objective, most non-trivial problems need to be handled by making use of methods based on Pareto optimization. The goal is then to design algorithms that approximate as well as possible the true Pareto front of a given problem. Most algorithms introduced for multi-objective optimization are Multi-objective Optimization Evolutionary Algorithms (MOEAs) (Branke et al., 2008, Zhou et al., 2011, Zitzler et al., 2000). Evolutionary Algorithms are population-based methods that reproduce natural and biological processes, such as evolution, bird migration, and

[1] `http://jmetal.sourceforge.net/problems.html`

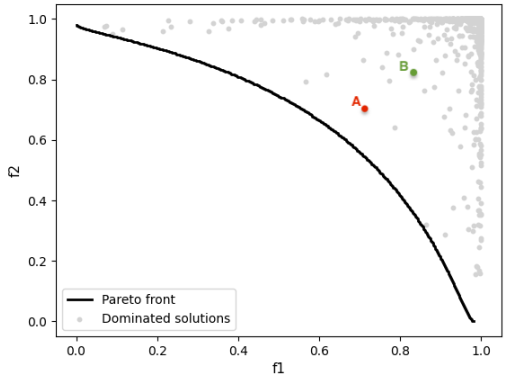

Figure 3.1: Pareto front of the Fonseca-Fleming function.

ant colonies. Among them, Genetic Algorithms (GA) (Koza, 1992) are the most popular. Genetic Algorithms are metaheuristics inspired by the concept of natural selection. In GAs, a population (i.e., a set of solutions) evolves generation after generation toward optimal solutions. For each generation, predefined proportions of fittest and less fit solutions (according to their objective values) are chosen to reproduce – using crossover and mutation operators – and create the population of the next generation. Using this procedure, the average fitness of the population improves generation after generation. GAs have long been an area of interest for researchers looking to solve optimization problems, making them a fitting choice for multi-objective optimization.

We find the first iteration of GAs starting with (Schaffer, 1985), which introduces the Vector Evaluated Genetic Algorithm. In (Ishibuchi and Murata, 1996), the authors introduce an *elitist strategy* for GAs. In a classical GA, only the population of the current generation is stored. With an *elitist strategy*, a second set of currently non-dominated solutions is also stored. After each iteration, all the solutions from the general population which are not dominated by any points in the non-dominated set are added to the set. In the next generation, a number of randomly select points from the non-dominated set – called *elite points* – are reintroduced into the general population. This ensures that good solutions are not lost from one generation to another, and it improves the convergence toward the globally optimal Pareto front. Surveys of the first generations of genetic and evolutionary algorithms have been compiled in (Fonseca et al., 1993, Van Veldhuizen and Lamont, 1998). In 1994, (Horn et al., 1994) introduced the *tournament selection technique*, which consists in randomly selecting points from the current population, to make them compete for their survival in the next generation. The competition is held using a tournament setting, where the points need to be fitter (i.e., non-dominated) than a given subset of the general population in order to survive.

We now discuss `NSGA-II` (Non-dominated Sorting Genetic Algorithm), the most popular MOEA, which was introduced almost 20 years ago in (Deb et al., 2002a). The `NSGA-II` algorithm possess several important features (i) it considers an *elitist strategy*, (ii) it employs a *tournament selection technique*, (iii) it uses a diversity preserving operator. At the start of the process, a random parent population is generated. Using standard genetic operators

(crossover and mutation) as well as tournament selection, a first offspring population is then produced. A new procedure is then introduced and repeated for each generation. These two populations – parents and offspring – are then combined, and a non-dominated sorting procedure is applied to generate an ordered set of non-dominated fronts according to their fitness level. To create the parent population of the next generation, the solutions of the best non-dominated fronts are then added one by one, until the currently considered front is too large to add all its solutions to the next parent population. When such a front is reached, the remaining parents of the current population are then chosen according to a diversity operator, instead of being chosen randomly. Then the offspring population of the next generation is created using the same techniques as for the first offspring generation.
Even though `NSGA-II` is by far the most popular GA, other highly competitive algorithms exist. Among them, we find the Pareto Envelope-based Selection Algorithm (Corne et al., 2000), the Pareto Archived Evolution Strategy (`PAES`) (Knowles and Corne, 2000) and the Strength Pareto Evolutionary Algorithm (`SPEA2`) (Zitzler et al., 2001). ε-MOEA (Deb and Jain, 2003) has also proved to be able to provide high-quality solution sets and the use of the ε-dominance (Laumanns et al., 2002) can be useful to solve high dimensional MOO problems. In (Drugan and Thierens, 2012), a Stochastic Pareto Local Search algorithm that exploits genetic operators was also introduced. While GAs are definitely an important part of Evolutionary Multi-objective Optimization (EMO), numerous other competing evolution-based methods have been introduced. We first find Particle Swarm Optimization (PSO), a metaheuristic whose original goal was to reproduce the behavior of animal packs, such as wolves, birds, or fish. In PSO, a population (the swarm) of individuals (the particles) is first generated. The particles are then driven to explore the search space following both their best-known position and the best-known position of the swarm. To solve MOO problems, several methods, called Multi-objective Particle Swarm Optimization algorithms have been introduced (Coello and Lechuga, 2002, Coello et al., 2004, Mostaghim and Teich, 2003) and have shown to be competitive with the best EMO methods, such as `NSGA-II` or `PAES`. Next, approaches based on ant-colony behavior (Alaya et al., 2007, Gravel et al., 2002, McMullen, 2001) have been investigated, and have also shown promising results compared to the state of the art on the studied application cases. Other evolutionary methods, like the Dragonfly algorithm (Mirjalili, 2016) and adaptations of differential evolution to MOO, have also been explored (Xue et al., 2003). Interactive evolutionary methods have also been investigated (Branke et al., 2009, Deb et al., 2010). Interactive MOO algorithms are iterative methods where user input is expected after each iteration of the algorithm. Consequently, the user has to possess the expert knowledge needed to express his preferences with regard to the direction that the search for optimal solutions is going to take at each step of the process.
While GAs have proved useful in many application scenarios, they still suffer from several drawbacks. As generic methods, choosing proper values for the numerous parameters is difficult (Boyabatli and Sabuncuoglu, 2004, Eiben et al., 1999, Srinivas and Patnaik, 1994). Indeed, tuning parameters such as population size, mutation rate, and crossover rate, but also formulating a proper fitness function or choosing the genetic encoding and elitism ratio is often made through trial and error. Furthermore, premature convergence is an issue that has also been widely studied (Pandey et al., 2014).

3.3.3 Non-Evolutionary Approaches

Although evolutionary approaches represent the bulk of the methods investigated for Pareto-based multi-objective optimization, other methods have been explored. Simulated Annealing (Suman and Kumar, 2006) is a global optimization metaheuristic that has shown good results in multi-objective optimization settings. In particular, the Archived Multi-objective Simulated Annealing algorithm introduced in (Bandyopadhyay et al., 2008) has proved to be superior to state-of-the-art MOEAs for many-objective optimization problems. Mathematical programming-based approaches propose a different approach to multi-objective optimization. In these methods, the algorithm is executed iteratively, and each iteration produces a different Pareto optimal solution. Examples of such approaches are the Normal Boundary Intersection method (NBI) (Das and Dennis, 1998), and the Normal Constraint method (NC) (Messac et al., 2003), an improvement over NBI. Indeed, while NBI provides no guarantee on the discovery of Pareto optimal solutions, NC always produces Pareto optimal solutions.

3.4 Quality Evaluation of Solution Sets

3.4.1 Types of Quality Measures

While algorithms can create sets of solutions, quality indicators that allow for the comparison and assessment of the quality of these solutions are needed. In the case of single-objective optimization, comparing the quality of the solutions provided by different algorithms is straightforward; we need only look at the value of the best solution found by each algorithm. In multi-objective optimization, a solution set is made of multiple optimal solutions, which makes the comparison of different algorithms harder. While visualization methods can be a simple and in some cases efficient way of comparing solutions, they become harder to exploit as the number of objectives goes up, and they lack the ability to provide an accurate measurement of the difference between sets of solutions. There is therefore a need for performance measures that can summarize the different qualities of a set of solutions into a unique value. Such performance indicators exist, and they can be summed up into 4 categories: *convergence*, *spread*, *uniformity*, and *cardinality* (Li and Yao, 2019). In the literature, we find quality measures that specialize in one category, but also measures that take into account several or all of the categories when assessing the quality of a solution set. For the convergence criteria, we find dominance-based and distance-based indicators. The most famous dominance-based measure is the C indicator (Zitzler and Thiele, 1998) and its variations (Datta and Figueira, 2012). They use the dominance relationship between two sets of solutions to measure their quality. Given two sets X and Y, $C(X, Y)$ measures the proportion of solutions of Y which are dominated by at least one solution of X. Formally, we have:

$$C(X, Y) = \frac{|y \in Y | \exists x \in X : x \prec y|}{|Y|}$$

The indicator takes values between 0 and 1, with 0 meaning that no solution of Y is dominated by any solution of X, and 1 meaning that all solutions of Y are dominated by at least one solution of X. It is important to note that to compare the performance of X and Y, both $C(X, Y)$ and $C(Y, X)$ need to be computed. Next, we look into the most used distance-based convergence indicator, the *Generational Distance (GD)* (Ishibuchi et al., 2015, Schutze et al.,

2012, Van Veldhuizen and Lamont, 1998) and its variations (Ishibuchi et al., 2015, Schutze et al., 2012). It measures the mean of the Euclidean distances between each point in the considered solution set and its closest point on the true Pareto front. Formally, given a solution set $X = \{x_1, ..., x_N\}$, it is defined as:

$$GD(X) = \frac{1}{N}\left(\sum_{i=1}^{N} mind(x_i, PF)\right)$$

where *mind* computes the minimal Euclidean distance from point i of the solution set to the true Pareto front.
For the *Generational Distance*, lower values are better and a GD of 0 means that the solution set lies entirely on the true Pareto front. While in the first iteration of the measure the *quadratic mean* was used, it has now become common to use the *arithmetic mean* instead since it is more resilient to outliers.
Let us now look at the spread criterion. To have a high-quality spread, a solution set should contain solutions close to every part of the true Pareto front. The most common spread measure is the *Maximum Spread (MS)* (Wu and Azarm, 2001, Zitzler et al., 2000). It computes the reach of a solution set by examining the range of each objective. Formally, we have:

$$MS(X) = \sqrt{\sum_{i=1}^{M} \underset{x,x' \in X}{max} (x_j - x'_j)^2}$$

with m the number of objectives. The *Maximum Spread* needs to be maximized. However, since the MS measure only considers extreme solutions, its measuring can often wrongly represent the actual spread of the solution set, especially when extreme solutions are outliers.

Let us now consider performance measures that compute the uniformity of a solution set. A set with good uniformity should have close or equal space between its solutions. The *Spacing (SP)* (Schott, 1995) indicator is the most commonly used uniformity measure. Informally, SP measures the variation of the space between solutions in a solution set using the Manhattan distance and needs to be minimized.
Finally, we also find indicators that measure the cardinality of solution sets. The idea behind this criterion is to count the number of non-dominated solutions. When the true Pareto front is involved, the idea is to count the number of points of the solution set that lie on the true Pareto front. In this setting, a common quality measure is the *Error Ratio (ER)* (Van Veldhuizen, 1999) which computes the proportion of points of a solution set that are not part of the true Pareto front. For ER, lower values are better since they represent sets with large proportions of Pareto optimal solutions.
While considering one or two of the four defining criteria is a good start, building performance indicators this way often falls short of providing a complete and accurate picture of the quality of solution sets. Consequently, researchers introduced indicators that involve all four criteria for obtaining good solutions. Since then, these measures have become widely used in the literature. Let us first consider a well-known distance-based measure, the *Inverted Generational Distance (IGD)* (Coello and Sierra, 2004). As per the name, IGD is the inverse of the GD indicator and measures the distance from the true Pareto front to the considered

solution set. Formally, given the true Pareto front $PF = \{pf_1, ..., pf_N, \}$ and a solution set X, it can be defined as:

$$IGD(X, PF) = \frac{1}{M}\left(\sum_{i=1}^{M} mind(pf_i, X)\right)$$

where M is the number of Pareto optimal solutions in the true Pareto front, and *mind* computes the minimal Euclidean distance from point i of the true Pareto front to the solution set X.
For IGD, lower values are better and reflect good values for all four criteria for a given solution set. In the absence of the true Pareto front, a reference set has to be used, and its ability to accurately represent the true Pareto front is crucial for the relevance of IGD. Since IGD is among the most used performance assessors in MOO, numerous variations have been introduced to improve upon it (Ibrahim et al., 2018, Ishibuchi et al., 2015, Schutze et al., 2012, Tian et al., 2016).
We can also look at volume-based measures which tick all four criteria needed for accurate performance assessments. The *Hypervolume (HV)* (Zitzler and Thiele, 1998) is the most widely used metric in MOO. It measures the volume of the area enclosed by the Pareto front and a specified reference point. The HV between a given solution set X and its reference point r is:

$$HV(X, r) = \lambda\left(\bigcup_{a \in X} \{x | a \prec x \prec r\}\right)$$

where λ is the Lebesgue measure.
However, the HV measure suffers from several drawbacks: (i) it needs a reference point, which can be hard to define precisely, and different reference points will favor different sets of solutions, (ii) its runtime increases exponentially with regard to the number of objectives, (iii) it favors convex regions over concave ones. Despite its limitations, HV has long been the preferred performance assessor in the MOO field, and can accurately represent the quality of solution sets in most scenarios. It is interesting to note that several approaches have been introduced to solve the problem of choosing a single solution or a reduced set of solutions from a set of Pareto optimal points (Ferreira et al., 2007, Fuente et al., 2018, Venkat et al., 2004).

3.4.2 Background Knowledge

Generally speaking, we find quality indicators that need a reference set of solutions (e.g. IGD) to compare new solution sets against, and quality indicators that need a reference point (e.g., HV) – such as the Nadir point or the ideal point – to be evaluated. Defining a proper reference set can be difficult in most application scenarios. Indeed, the reference set needs to be as close as possible to the true Pareto front, and must represent a front with high qualities for *convergence*, *spread*, *uniformity*, and *cardinality*. A good alternative is to exploit an empirical method by building the reference set out of all the globally non-dominated solutions found so far across all generated solution sets.
Reference points are needed for several widely used performance metrics. The most common are the ideal point (Vincent and Grantham, 1981, Zeleny, 1973) and the Nadir point (Deb et al., 2006). The ideal point consists of the optimal values of each objective in the entire

search space. While the ideal point or a close approximation of it (i.e., using the best-known values instead of the unknown optimal ones) can easily be defined using the available solution sets, determining a proper value for the Nadir point can be very difficult. The Nadir point is defined as the vector of the worst possible value of each objective in the optimal true Pareto front. One issue with the Nadir point is its impossibility to be precisely estimated in most situations. Indeed, it requires an optimal or near-optimal Pareto front to get a good estimate of the worst value of each objective, which is rarely computable in real-life scenarios. While some studies have been done to introduce methodologies for defining proper Nadir points, researchers have discovered solutions to the problem only in few cases (Cao et al., 2015).
For a lot of existing quality measures, objectives need to be scaled, so that objectives with larger range of values have the same effect on the metric as other objectives. A common method for scaling is to normalize each objective with values in [0,1] by using the following formula: $x_j^{'} = (x_j - min_j)/(max_j - min_j)$, where min_j and max_j are respectively the minimum and maximum of Objective j. Scaling is not needed for some measures that are called *scaling independent* (Zitzler et al., 2008).

3.5 Benchmark Functions, Applications and Tools

To provide a performance assessment of any MOEA, test functions must be used. These benchmarks were created so that all algorithms could be compared on the same problems. For these problems, the Pareto optimal solutions and the shape of the true Pareto front are known, such that quality measures can be used to evaluate how close to the ideal front the solutions generated by the algorithms are. Early test problems suffered from several limitations: they were either too simple to solve, not scalable, or impossible to visualize. Among the first proposed benchmark functions, we find the (Kursawe, 1990) and (Fonseca and Fleming, 1995) unconstrained problems which both involve 2 objectives and an unlimited number of attributes. A constrained test problem was introduced in (Binh, 1999, Binh and Korn, 1997). It involves 2 objectives, 2 attributes, and 2 constraints. The benchmark `Fonseca` has been introduced in Section 3.3.1 and it will be used in Chapter 5.
While these functions allowed for the quality assessment of early evolutionary algorithms, more complex benchmark problems were needed. Indeed, to mimic real-life applications where large amount of competing objectives and constraints are common-place, the scalability and complexity of new and existing approaches had to be empirically investigated. In (Zitzler et al., 2000), the authors introduced the ZDT test toolkit, followed by the well-known DTLZ (Deb et al., 2002b, 2005) and WFG (Huband et al., 2006) problem suites. Thanks to these benchmark suites, the quality of state-of-the-art multi-objective algorithms can be assessed on synthetic data.
The main issue with synthetic data is that it can sometimes be far removed from real-life scenarios. For this reason, most performance studies of new algorithms also involve some type of investigation on a few real-life problems. However, until recently, there was no toolkit that involved a set of real-life diverse multi-objective optimization problems on which algorithms could be tested and compared. Consequently, the REal world problem suite RE (Tanabe and Ishibuchi, 2020) was introduced to fill this gap. It involves 16 real-world problems with low computational cost, so that algorithms can be tested efficiently on a more diverse and lifelike set of problems.

Potential applications for multi-objective optimization algorithms are numerous. While optimization problems involving more than 1 objective used to be essentially treated as single-objective, the advent of computing and the introduction of more complex approaches enabled the use of MOO methods in a large number of domains.
In (Surekha et al., 2012), the authors exploit genetic and particle swarm algorithms to optimize the green sand mould system. In this scenario, several parameters such as grain fineness or percentage of clay lead to different mould properties, like green compression strength, permeability, or hardness. These properties are the defining factors driving the quality of the end products, i.e., the casts. The authors therefore apply MOO algorithms on the 4 objective optimization problem to obtain the best compromise leading to high-quality casts.
In the literature, we find numerous use cases of MOO involving ecological issues, such as building retrofit strategies (Asadi et al., 2012), nearly-zero-energy-building design (Hamdy et al., 2016), and environmental protection (Cui et al., 2017). Multi-objective optimization methods are also widely used in chemical engineering (Rangaiah, 2016), but also to solve efficiency-cost optimization problems in engineering fields (Shirazi et al., 2014), and for sensor placement optimization in indoor systems (Domingo-Perez et al., 2016).
For several machine learning optimization problems, multiple metrics need to be optimized at the same time. While in most cases optimizing one metric is still considered as being good enough, it has been shown that one metric can often not be enough to assess the quality of a model. In Shi et al. (2012), the authors argue that in many real applications, optimizing only one quality measure is sub-optimal for multi-label classification tasks. Consequently, they propose a new method called Multi-Objective Multi-Label algorithm and show its relevance compared to the state of the art to solve multi-label classification optimization problems. Finally, Caballero et al. (2010) introduce a multiclass classification algorithm based on multi-objective optimization to treat the problem of maximizing two conflicting objectives of multiclassifiers, (i) the correct classification rate level, and (ii) the classification rate for each class.

Solutions have been designed to support the visualization of MOO solutions easier. In the 2D and 3D cases, scatter plots can be used and are usually sufficient to extract the needed information. However, when MOO problems involve high-dimensional data, visualizing and extracting relevant information is a hard task. (Tušar and Filipič, 2014) proposes a comprehensive review of existing methods to visualize Pareto front approximations.
Typical high-dimensional visualization methods involve heatmaps (Pryke et al., 2007) and parallel coordinates (Inselberg and Dimsdale, 1990), where each solution is represented on a parallel coordinate system. A common problem of heatmap visualization is the random ordering of rows and columns. In (Walker et al., 2012), the authors first introduce a solution to this issue by means of spectral seriation and also propose a representation of mutually non-dominated sets based on Radviz (Hoffman et al., 1997). In (He and Yen, 2015), a new method is proposed to map high-dimensional solutions into a 2D polar coordinate plot that preserves the Pareto dominance relationship. The approach possesses several advantages: (i) it is scalable to any number of objectives, (ii) it can handle Pareto fronts with large number of solutions, (iii) multiple solution sets can be visualized at the same time, which enables the easy comparison of several fronts.
Several tools to work on multi-objective optimization problems are available. The first tool introduced is JMetal (Durillo and Nebro, 2011), a Java-based framework for multi-objective

optimization. It contains numerous implementations of well-known MOO algorithms – such as `NSGA-II`, `SPEA2` and `MOEA/D` – and it provides most common performance metrics, like the hypervolume or the generational distance. Moreover, single-objective versions of MOO algorithms and parallel algorithms are also made available. Finally, it is interesting to note that most benchmark functions – constrained and unconstrained – and test toolkits – ZDT, DTLZ, WFG – are also available. We also find the PlatEMO (Tian et al., 2017) platform, a MATLAB tool specialized in evolutionary multi-objective optimization. It proposes over 50 MOEAs and makes it easy to compare several algorithms at the same time. Numerous performance metrics and test problems are also made available to the user. Furthermore, it enables the community to develop new algorithms, metrics, and test problems by being open source. Finally, with the Python language becoming prevalent in data science and other related fields over the last few years, a new Python framework for MOO named Pymoo (Blank and Deb, 2020) was recently introduced. Once again, a large variety of single and multi-objective algorithms, performance metrics, and test problems are proposed. It is interesting to note that Pymoo also contains methods for high-dimensional visualization of solutions, and provides tools for multi-criteria decision making.

3.6 Cross-Fertilization of Pattern Discovery and Multi-Objective Optimization

3.6.1 Pattern Discovery and Multi-Objective Optimization

While both MOO and pattern discovery have been seriously investigated, contributions involving the coupling and/or interactions of both approaches have been relatively few and far between (Srinivasan and Ramakrishnan, 2011). Among these contributions, the authors of (Kaya and Alhajj, 2004) propose a new genetic algorithm based method to mine optimized fuzzy association rules. Regarding the objectives, they consider the optimization of the support, the confidence, and the number of fuzzy sets. They use the common concept of dominance and a genetic algorithm to mine for Pareto optimal fuzzy rules. In their experiments, they show the superiority of fuzzy rule mining over crisp rules when it comes to optimizing both the support and confidence of the rules. For fuzzy rule mining, we also find the work introduced in (Gacto et al., 2009), where the authors describe and compare the application of 6 evolutionary algorithms to mine for rules with high accuracy and interpretability.

In (Dehuri and Mall, 2006), the authors propose a multi-objective genetic algorithm to mine for predictive classification rules. Their goal is to optimize both the predictive accuracy and the comprehensibility of the rules at the same time, two objectives that usually have an antagonistic relationship. Their method, called `INPGA` for Improved Niched Pareto Genetic Algorithm, allows for finding better rules than other benchmark genetic algorithms. A multi-objective metaheuristic is introduced in (Reynolds and de la Iglesia, 2006) for rule induction in a context of partial classification. The authors consider the optimization of the coverage and confidence of the rules, and a modified version of the dominance relation to discover a more diverse set of classification rules. In experiments, they show that this metaheuristic using a modified dominance relation allows for the discovery of rules that provide

new information. An extension of the famous NSGA-II evolutionary algorithm to mine for quantitative association rules was introduced in (Martín et al., 2011). Although mining for quantitative association rules has been studied extensively, the focus was on optimizing only one objective. In this approach, the authors propose and show how to discover quantitative rules with the best trade-offs between interestingness, comprehensibility, and performance (i.e., the product of confidence and support).

(Soulet et al., 2011) has exploited the notion of *skyline queries*, allowing the introduction of the notion of *skyline patterns*. In their work, they focus on mining useful patterns, according to a set of user preferences. Since *skyline queries* involve multiple constraints of equal importance, a trade-off has to be found between these constraints, which is exactly the subject of MOO. They use the notion of dominance between patterns to look for those that are non-dominated according to the set of constraints. These non-dominated patterns are called *skyline patterns*, and in MOO terms, they correspond to Pareto optimal patterns, i.e., the set of patterns that lie on the Pareto front. Their approach presents several advantages (i) it finds patterns that are non-dominated by any other pattern, (ii) it is generic to any kind of pattern which can be queried through a skyline query, (iii) the study of the relationships between condensed representations of patterns and skyline pattern mining enables to compute the set of *skyline patterns* efficiently. In (Ugarte et al., 2017), the authors extend the work in (Soulet et al., 2011) by investigating further the relationships between the so-called condensed representations of patterns (Calders et al., 2006) and skyline pattern mining. As a result, they can build an interesting skypattern mining algorithm based on a dynamic constraint satisfaction problem.
The exploitation of *skyline patterns* for Skyline EMM is studied on the plant growth recipe optimization scenario of Chapter 6 to improve the diversity of the computed patterns.

3.6.2 Subgroup Discovery and Multi-Objective Optimization

If cross-fertilization between pattern discovery and multi-objective optimization is rare, the coupling of SD/EMM and MOO is almost non-existent. Indeed, only a handful of approaches have investigated the topic. In (Del Jesus et al., 2007a), the authors propose the use of a multi-objective genetic algorithm to discover interesting subgroups based on fuzzy rules. They introduce the MESDIF (Multiobjective Evolutionary Subgroup DIscovery Fuzzy rules) algorithm, based on the well-known SPEA2 genetic algorithm. The end goal is to discover subgroups with optimized trade-offs between *confidence*, *support* and a new measure of diversity called *original support* that defines the originality of the rule compared to other rules of the population. In (Del Jesus et al., 2007b), the Subgroup Discovery Iterative Genetic Algorithm (SDIGA) is proposed. The authors want to optimize the *confidence* and *support* of the subgroups. However, SDIGA is a single-objective optimization algorithm, meaning that to discover interesting subgroups, it has to transform the set of objectives into a single-objective using a predefined method. In this scenario, the authors transform the objectives by applying the weighted sum method. Both MESDIF and SDIGA were used in a real-life application case for subgroup discovery in a psychiatric emergency department (Carmona et al., 2011). The goal was to find subgroups that provided relevant information with regard to the relationship between the arrival time of patients at the emergency department, and the types of pathologies they are affected by. Three objectives – support, confidence, and unusualness

– needed to be optimized. In this study, the authors compared the subgroups discovered by both MESDIF and SDIGA, and showed that MESDIF can find higher quality subgroups. They also showed that MESDIF allowed for the discovery of interesting information, such as the fact that suicide attempts are more frequent during nighttime. SDIGA has also been used in (Romero et al., 2009) for subgroup discovery in e-learning.
In (Carmona et al., 2010), the authors introduced an evolutionary fuzzy system named NMEEF-SD, based on NSGA-II to discover interpretable and high quality subgroups. In this work, the author consider the discovery of subgroups with the best trade-offs between *support*, *fuzzy confidence* and *unusualness*. In an empirical study, they show the superiority of their approach compared to the state of the art, i.e., MESDIF and SDIGA. While their approach is interesting in their targeted context, it lacks genericity: it allows for the discovery of subgroups with a good trade-off between a few pre-defined objectives and it focuses on computing the Pareto front at the subgroup level.
The concept of *skyline* was exploited in (Van Leeuwen and Ukkonen, 2013) to mine for skylines of subgroup sets. Indeed, the authors argue that the best way to mine for sets of subgroups that are both diverse and of high quality, there is a need to consider the set of Pareto optimal subgroups with regard to those objectives. To do this, the authors detail an exhaustive and a heuristic algorithm for the discovery of top-K subgroup sets that offer the best trade-offs between quality and diversity.
A common thread between all these approaches is the computation of Pareto optimal patterns at the subgroup – and rule/pattern – level (i.e., they consider the Pareto front of subgroups and not the Pareto front of objects of the dataset). While this is a relevant way of combining multi-objective optimization and subgroup discovery, it is however interesting to note that, to the best of our knowledge, no method which cross-fertilizes MOO and SD/EMM at the object level has been proposed so far.

3.7 Conclusion

Multi-objective optimization is a sub-field of multi-criteria decision making whose goal is to discover optimal trade-offs between a number of objective functions. Numerous approaches have been proposed in the past 40 years, including classical methods that treat the problem as single-objective, and Pareto-based methods that look for the ideal set of Pareto optimal solutions. Nowadays, Pareto-based methods are largely prevalent in the literature and are widely used for most application cases. Among them, Evolutionary Multi-objective Algorithms such as NSGA-II and SPEA2 have become standard tools that perform well in most scenarios. However, their genericity can also be their downfall: finding proper values for the numerous parameters is difficult and mostly involves trial and error, which restricts their usage to settings where numerous experiments can be carried out. Pareto-based algorithms need for performance indicators that assess the quality of solution sets. Ideally, a solution set should perform well in four distinct and complementary criteria: *convergence*, *spread*, *uniformity*, and *cardinality*. To evaluate the performance of MOO algorithms, numerous test functions have been introduced, including a number of benchmarks that reflect real-life application cases. When working on MOO problems with high dimensional data (i.e., more than 3 objectives), visualizing and interpreting the results can be difficult. For this reason, several visualization approaches have been introduced to simplify the process and easily extract

actionable information from solution sets. It is interesting to note that tools for performing MOO have been developed and are accessible to all in some widely used programming languages in data-related fields, such as Python, Java, and Matlab.
Cross-fertilization between MOO and Pattern Discovery is a domain that has seen relatively few works being introduced, and even more so when the focus is being put on SD and EMM. The few approaches that exist focus on finding good trade-offs between several common metrics that define high-quality patterns.
There is however no work in the literature that has investigated the coupling of MOO and SD/EMM at the object level, i.e., working with Pareto-based concepts on the objects of the datasets. Furthermore, current existing MOO methods have several limitations (i) when the underlying model of the objective functions is unknown, existing approaches can not be used, since new points can not easily be generated, and (ii) typical MOO algorithms require a large number of points to be generated at each iteration, which is antinomic to many real-life scenarios where experiments are limited due to time and cost constraints. There is therefore a need for MOO-based methods that would not suffer from such limitations.

Chapter 4

Optimal Subgroup Discovery in Purely Numerical Data

Subgroup discovery in labeled data is the task of discovering patterns in the description space of objects to find subsets of objects whose labels show an interesting distribution, for example the disproportionate representation of a label value. Discovering interesting subgroups in purely numerical data – attributes and target label – has received little attention so far. Existing methods make use of discretization techniques that lead to loss of information and suboptimal results. This is the case for the reference algorithm `SD-Map*` (Atzmueller and Lemmerich, 2009). In this chapter, we consider the discovery of optimal subgroups according to an interestingness measure in purely numerical data. We leverage the concepts of closed interval patterns, advanced enumeration and pruning techniques. The performances of our algorithm are studied empirically and its added-value w.r.t. `SD-Map*` is illustrated.

4.1 Exploiting Labeled Numerical Data

Mining purely numerical data is quite popular. It concerns data made of objects described by numerical attributes, and one of these attributes can be considered as a target label. We can then choose to learn models to predict the value of the label for new objects, or we can apply subgroup discovery methods, which is the focus of our work. A large panel of exhaustive and heuristic subgroup discovery algorithms has been proposed so far. Regarding numerical attributes, a few approaches that avoid the use of basic discretization techniques have been introduced. However, to the best of our knowledge, we lack a method that would support an exhaustive search and thus the possibility to guarantee the computation of a global optimum for a selected quality measure without the use of discretization in some form or other. When considering numerical target labels, `SD-Map*` is the reference algorithm. Notice however that `SD-Map*` requires the prior discretization of the numerical attributes.

The guaranteed discovery of an optimal subgroup in purely numerical data is a useful task and we now motivate it for optimizing processes in urban farms. In that setting, plant growth recipes involve many numerical attributes (temperature, hydrometry, CO_2 concentration, etc) and a numerical target label (the yield, the energy consumption, etc). Our goal is to mine the recipe execution records (i.e., the collected measures) to discover the characteristics of an optimized growth. In expert hands, such characteristics can be exploited to define better recipes. In such a context, the guaranteed discovery of the optimal subset of recipes with respect to the target label is more relevant than the heuristic discovery of the k best subgroups with no optimality guarantee. The exploitation of these contributions for plant growth recipe optimization is studied in-depth on synthetic and real-life data in Chapter 6.

To achieve the search for optimality, we decided to search the space of interval patterns as defined in (Kaytoue et al., 2011). Our main contribution consists of an algorithm that exhaustively enumerates all the interval patterns. Our approach (i) exploits the concept of closure on the positives adapted to a numerical setting to operate in a subspace (ii) uses a new faster tight optimistic estimate that can be applied for several quality measures (iii) uses advanced pruning techniques (forward checking, branch reordering). The result is the efficient algorithm `OSMIND` for optimal subgroup discovery in purely numerical data without prior discretization of the attributes.

Definition 18. *A purely numerical dataset (G, M, T) is given by a set of objects G, a set of numerical attributes M and a numerical target label T. In a given dataset, the domain of any attribute $m \in M$ is a finite ordered set denoted D_m. In this context, $m(g) = d$ means that d is the value of attribute m for object g. The domain of label T is also a finite ordered set denoted D_T. $T(g) = v$ means that v is the value of label T for object g.*

Fig. 4.1 (left) is an example of a purely numerical dataset made of two attributes ($M = \{m_1, m_2\}$) and a target label T.

We want to leverage several concepts of both interval pattern mining and SD introduced in Chapter 2. Most notably, we consider a closure operator on interval patterns, as well as the q^a_{mean} quality measure and corresponding optimistic estimates to compress and prune the search space. We also make use of the enumeration strategy of `MinIntChange` for an optimal subgroup discovery – using the optimality definition of Chapter 2 – in purely numerical data. Fig. 4.1 (right) depicts the dataset of Fig. 4.1 (left) in a cartesian plane as well as a comparison between a non-closed (c_1) and a closed (c_2) interval pattern.

The material in this chapter has been published in the Proceedings of the 2020 Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD) (Millot et al., 2020a), as well as in the Proceedings of the 2020 conference Extraction et Gestion des Connaissances (EGC) (Millot et al., 2020b). For reproducibility purposes, described datasets and source code are made available in `https://bit.ly/3ilhir5`.

The remaining of this chapter is organized as follows. We detail our contributions in Section 4.2 before an empirical evaluation in Section 4.3. Section 4.4 briefly concludes.

4.2 Optimal Subgroup Discovery

4.2.1 Closure On The Positives

The closure operator on interval patterns introduced in (Kaytoue et al., 2011) has been extended to closure on the positives for binary labels in (Belfodil et al., 2018, Guyet et al., 2017).

Definition 19. *Let $p \in P$ be an interval pattern, $p' \subseteq p$ a second interval pattern, and T a binary target label. An object is said to be positive if its label value is that of the class we want to discriminate, and negative in the opposite case. Let $ext(p)^+$ be the subset of objects of $ext(p)$ whose label T is positive. p' is said to be closed on the positives if it is the most restrictive pattern enclosing $ext(p)^+$. If q is the quality measure, we have $q(p) \leqslant q(p')$.*

For all subgroups $p \in P$, if all negative objects which are not in the extent of p' are removed from the extent of p, then the subgroup quality cannot decrease. Note that closed on the positives are a subset of closed patterns.

The concept of closed on the positives for binary target labels can be extended to numerical target labels for a set of quality measures, including q^a_{mean}. We transform the numerical label into a binary label: objects whose label value is strictly higher (resp. lower or equal) than the mean of the dataset are defined as positive (resp. negative). Note that the quality measure is computed on the raw numerical label. The binarization is only used to improve search space pruning and it does not lead to loss of information concerning the resulting patterns (i.e., the optimal subgroup discovery without discretization is guaranteed). Fig. 4.2 (left) is the

	m_1	m_2	T
g_1	1	1	15
g_2	1	2	30
g_3	2	2	60
g_4	3	2	40
g_5	3	3	70
g_6	4	3	85

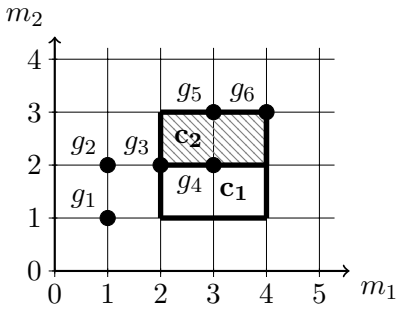

Figure 4.1: (**left**) Example of a purely numerical dataset described by 2 numerical attributes and a numerical target concept. (**right**) Non-closed ($c_1 = \langle [2,4],[1,3] \rangle$, non-hatched) and closed ($c_2 = \langle [2,4],[2,3] \rangle$, hatched) interval patterns.

	m_1	m_2	T	T_b
g_1	1	1	15	−
g_2	1	2	30	−
g_3	2	2	60	+
g_4	3	2	40	−
g_5	3	3	70	+
g_6	4	3	85	+

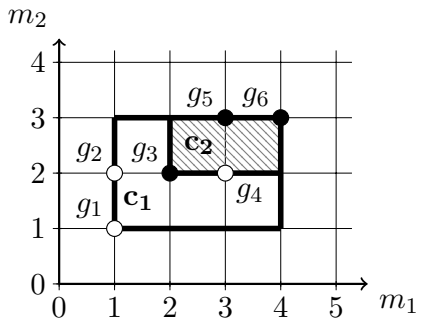

Figure 4.2: (**left**) Purely numerical dataset with binary label (T_b). (**right**) Closed ($c_1 = \langle [1,4],[1,3] \rangle$, non hatched) and closed on the positives ($c_2 = \langle [2,4],[2,3] \rangle$, hatched) interval patterns.

dataset of Fig. 4.1 with label T (mean = 50) transformed into the binary label T_b. Fig. 4.2 (right) depicts the dataset of Fig. 4.2 (left) in a cartesian plane and a comparison between a closed (c_1) and a closed on the positives (c_2) interval pattern. We separate the case where the subgroup quality is positive from the case where it is negative. Given a subgroup of positive quality, we can prove that its quality is always higher or equal if all negative objects not in the closure on the positives are removed.

Theorem 1. *Let p be an interval pattern, q^a_{mean} a set of quality measures, p^+ the closure on the positives of p such that $p^+ \subseteq p$, and $q^a_{mean}(p) \geqslant 0$, then $q^a_{mean}(p^+) \geqslant q^a_{mean}(p)$, $a \in [0,1]$.*

Proof. Let $ext(p)$ be the extent of p, $ext(p)^+$ the extent of p^+, $ext(p)^- = ext(p) \backslash ext(p)^+$ the set of negative objects of $ext(p)$ not in $ext(p)^+$, and $T(i)$ the target label value for Object i. For shorter notation, we define $e = ext(p)$ and $\theta = ext(\varnothing)$. We prove that:

$$|e^+|^a \times (\mu_{e^+} - \mu_\theta) \geqslant |e|^a \times (\mu_e - \mu_\theta) \tag{4.1}$$

Which can be transformed into:

$$|e^+|^a \times \frac{\sum_{i \in e^+}(T(i) - \mu_\theta)}{|e^+|} \geqslant |e|^a \times \frac{\sum_{i \in e}(T(i) - \mu_\theta)}{|e|} \tag{4.2}$$

$$|e^+|^a \times \frac{\sum_{i \in e^+}(T(i) - \mu_\theta)}{|e^+|} \geqslant (|e^+| + |e^-|)^a \times \frac{\sum_{i \in e^+}(T(i) - \mu_\theta) + \sum_{i \in e^-}(T(i) - \mu_\theta)}{|e^+| + |e^-|} \tag{4.3}$$

By construction, we know that $\sum_{i \in e^+}(T(i) - \mu_\theta) \geqslant 0 \geqslant \sum_{i \in e^-}(T(i) - \mu_\theta)$. The rest of the proof follows the same as (Lemmerich et al., 2016a). We deduce that for any subgroup verifying $q^a_{mean}(p) \geqslant 0$, the closure on the positives always leads to a subgroup of equal or higher quality. □

The case of a negative quality subgroup is more complex since the closure on the positives can lead to a decrease in the subgroup quality. We prove that objects which are not in the closure on the positives can never be part of the best subgroup specialization.

Theorem 2. *Let p be an interval pattern, p^+ the closure on the positives of p such that $p^+ \subseteq p$ and $ext(p)^+$ its extent with $|ext(p)^+| > 0$. Let $ext(p)^- = ext(p) \backslash ext(p)^+$ be the set of negative objects of $ext(p)$ not in $ext(p)^+$, and q^a_{mean} a set of quality measures with $q^a_{mean}(p) < 0$: No object in $ext(p)^-$ can be part of the best specialization of p.*

Proof. Let us assume that there exists an object in $ext(p)^-$, denoted i^-, which belongs to the best specialization of p, denoted p_{top}. By construction, $q^a_{mean}(p_{top}) > 0$ (since $|ext(p)^+| > 0$). Let p^+_{top} be the closure on on the positives of p_{top}. By construction, we know that i^- is not part of the extent of p^+_{top} (since i^- doesn't belong to p^+). Yet, according to Theorem 1, we have $q^a_{mean}(p^+_{top}) \geqslant q^a_{mean}(p_{top})$. We deduce that i^- doesn't belong to the best specialization of p. □

4.2.2 Tight Optimistic Estimate

We now introduce a new tight optimistic estimate for the family of quality measures q^a_{mean}. An optimistic estimate is said to be tight, if, for any subgroup of the dataset, there is a subset of objects of the subgroup whose quality is equal to the value of the subgroup optimistic estimate. Note that the subset does not need to be a subgroup. It is possible to derive a tight optimistic estimate for the quality measures q^a_{mean} by considering each object of a subgroup only once.

Definition 20. *Let p be an interval pattern, and $S_i \subseteq ext(p)$ the subset of objects of $ext(p)$ containing the i objects with the highest label value. Then, as defined in (Lemmerich et al., 2016a), a tight optimistic estimate for q^a_{mean} is given by:*

$$bss^a_{mean}(p) = max(q^a_{mean}(S_1), ..., q^a_{mean}(S_{|ext(p)|})), a \in [0, 1]$$

We can derive a better optimistic estimate by focusing on positive objects only.

Theorem 3. *Let p be an interval pattern and $ext(p)^+$ the set of objects from the extent of p whose label value is higher than the mean of the dataset. Let $S_i \subseteq ext(p)^+$ be the subset of objects containing the i objects with the highest label value. A new tight optimistic estimate for q^a_{mean} is given by:*

$$\overline{bss}^a_{mean}(p) = max(q^a_{mean}(S_1), ..., q^a_{mean}(S_{|ext(p)^+|})), a \in [0, 1]$$

Proof. We need to prove that:

$$\overline{bss}^a_{mean}(p) \geqslant bss^a_{mean}(p), a \in [0, 1]$$

In other words, we need to show that: $\forall S_i \subseteq ext(p), q^a_{mean}(S^+_i) \geqslant q^a_{mean}(S_i)$ with S^+_i the subset of positive objects of S_i. In (Lemmerich et al., 2016a), it is proven that no negative object belongs to the best subgroup's subset of objects for the quality measures q^a_{mean}. It follows logically that for any subset S_i, removing the negative objects can not lower its quality. Thus, we have

$$\forall S_i \subseteq ext(p), q^a_{mean}(S^+_i) \geqslant q^a_{mean}(S_i)$$

We deduce that:

$$max(q^a_{mean}(S_1), ..., q^a_{mean}(S_{|ext(p)^+|})) \geqslant max(q^a_{mean}(S_1), ..., q^a_{mean}(S_{|ext(p)|})), a \in [0, 1]$$

Thus, $\overline{bss}^a_{mean}(p)$ is a tight optimistic estimate for q^a_{mean}. □

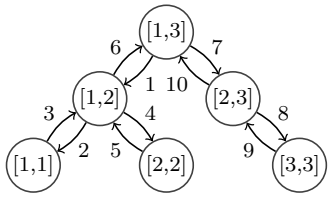

Figure 4.3: Depth-first traversal of D_{m_2} (from Fig. 4.2 (left)) using minimal changes.

4.2.3 Algorithm

We introduce OSMIND, a *depth first search* algorithm for an optimal subgroup discovery. It computes closed on the positives interval patterns coupled with the use of tight optimistic estimates and advanced search space pruning techniques. The pseudocode is available in Algorithm 1.

To guarantee an optimal subgroup discovery, we adopt the concept of minimal change from MinIntChange that ensures an exhaustive enumeration of all interval patterns (see Fig. 4.3 for an example with one attribute). A right minimal change consists in replacing the right bound of an interval by the current value closest lower value in the domain of the corresponding attribute. Following the same logic, a left minimal change consists in replacing the left bound by the closest higher value. The search starts with the minimal interval pattern that covers all the objects of the dataset. The main idea in procedure RECURSION is to apply consecutive left or right minimal changes until obtaining an interval whose left and right bounds have the same value for each interval of the minimal interval pattern. If so, the algorithm backtracks until finding a pattern on which a minimal change can be applied. We leverage the concept of closure on the positives adapted to numerical labels to significantly reduce the number of candidate interval patterns. After each minimal change (Line 4), instead of evaluating the resulting interval pattern, we compute and evaluate the corresponding closed on the positives interval pattern (Line 5). When carrying out an exhaustive search of all closed on the positives interval patterns, a given interval pattern can be generated multiple times. To avoid this redundancy and to ensure the unicity of the pattern generation, a popular solution is the use of a canonicity test. In the case of interval patterns, the canonicity test verifies that the closure operation did not lead to a change on an interval preceding the interval on which the minimal change has been applied (Line 6). However, the successive application of left or right minimal changes on an interval can also lead to multiple generations of the same interval pattern. A solution is to use a constraint on the minimal changes. After a right minimal change, a right or left minimal change can be applied. However, a left minimal change must always be followed by a left minimal change. We also exploit advanced pruning techniques to reduce the size of the search space. This can be done through the use of a tight optimistic estimate of the quality of a closed on the positives interval pattern specializations. For each subgroup, an optimistic estimate is derived (Line 7), and, if it is lower than the best subgroup quality, the search space is pruned by discarding every specialization of this interval pattern. Our second implemented technique is the coupling of *forward checking* and *branch reordering*. Given an interval pattern, the set of all its direct specializations (application of a right or left minimal change on each interval) are computed – forward checking – and those whose optimistic estimate is higher than the best subgroup are stored (Line 8). Branch reordering by descending order of the optimistic

estimate value is then carried out (Line 14). Branch reordering enables the exploration of the most promising parts of the search space first. It also enables a more efficient pruning by raising the minimal quality earlier. While our algorithm provides the guarantee of discovering an optimal subgroup, it is important to note that there can be multiple optimal subgroups with the same quality for a given dataset. Indeed, we can discover subgroups of equal quality, but with differing trade-offs between size of the subgroup (i.e., number of objects), deviation from the global model, and length of the subgroup description. In this situation, it is up to the user to define an order of importance on the criteria and to choose which subgroup will serve him best.

4.3 Empirical Validation

We consider 5 purely numerical datasets described in Table 4.1. Source code of implemented algorithms and used datasets are available at `https://bit.ly/3ilhir5`. `SD-Map*` implementation is available within the VIKAMINE system[1]. The 5 datasets (`Bolt`, `Basketball`, `Airport`, `Body Temp` and `Pollution`) originate from the Bilkent[2] repository. An extended version of these experiments is available in our paper (Millot et al., 2020a). For the

[1] `http://www.vikamine.org/`
[2] `http://funapp.cs.bilkent.edu.tr/DataSets/`

Algorithm 1 `OSMIND` algorithm

```
1: function OSMIND( )
2:     Initialize(minimal_interval_pattern, optimal_pattern)
3:     RECURSION(minimal_interval_pattern, 0)
4:     return optimal_pattern
5: end function
1: procedure RECURSION(pattern, attribute)
2:     for (i = attribute to nb_attributes − 1) do
3:         for (elem in {right, left}) do
4:             pattern ← minimalChange(pattern, i, elem)
5:             closed_pattern ← computeClosureOnThePositives(pattern)
6:             if (isCanonical(closed_pattern)) then
7:                 if (tightOptEst(closed_pattern) > quality(optimal_pattern)) then
8:                 store(closed_pattern, i) end if
9:                 if (quality(closed_pattern) > quality(optimal_pattern)) then
10:                optimal_pattern ← closed_pattern end if
11:            end if
12:        end for
13:    end for
14:    for (each element stored ordered by optimistic estimate value) do
15:        if (tightOptEst(element.pattern) > quality(optimal_pattern)) then
16:        RECURSION(element.pattern, element.attribute) end if
17:    end for
18: end procedure
```

purpose of our experiments, we also need to be able to generate datasets of different sizes. Fortunately, benchmark functions for single-objective optimization have been introduced to evaluate the quality of optimization methods. We consider the `Sphere` function – whose implementation is available at `https://bit.ly/3gYnt3Y` – that implies n descriptive variables and 1 objective to minimize:

$$f(\mathbf{x}) = \sum_{i=1}^{n} x_i^2,\ x_i \in [-5.12, 5.12], \forall i \in \{1, ..., n\}$$

The global minimum of the function is found in $f(0, ..., 0) = 0$. Thanks to this function, we can easily generate synthetic purely numerical datasets to experiment on. Table 4.2 provides a toy dataset made of `Sphere` objects with $n = 4$ attributes and the function to minimize.

Table 4.1: Datasets and their characteristics: number of attributes, number of objects and size of the search space.

Dataset	Attr	Obj	|P|
`Bolt`	8	40	8.7×10^9
`Basketball`	4	96	2.3×10^{11}
`Airport`	4	135	7.1×10^{15}
`Body Temp`	2	130	1.8×10^3
`Pollution`	15	60	1.7×10^{42}

Table 4.2: Example of `Sphere` dataset made of objects with 4 descriptive variables and 1 target to minimize.

G	x_1	x_2	x_3	x_4	f_1
g_1	-2.84	-1.71	0.57	-3.98	27.2
g_2	-5.12	3.98	-5.12	1.71	71.2
g_3	-1.71	-2.84	-2.84	-3.98	34.9
g_4	5.12	3.98	-1.71	-1.71	47.9
g_5	-3.98	3.98	-2.84	3.98	55.7

Performance improvements provided by our contributions are summarized in Table 4.3. Performances of the closure on the positives operator are compared to those of a simple closure operator. For each dataset, we compare the number of evaluated subgroups before finding the optimal one for the quality measure q^a_{mean} with $a = 0.5$ and $a = 1$. In all the cases, the closure on the positives is significantly more efficient. In fact, our method enables us to divide the number of considered subgroups by an average of more than 20. Let us now study the potential performance improvement – in terms of execution time in seconds – provided by our new tight optimistic estimate. We compare it to the tight optimistic estimate from (Lemmerich et al., 2016a) on all the datasets with the same quality measures. Our optimistic estimate is more efficient in all cases and it provides an execution time decrease of up to 30%.

Next, we want to study the scalability of our algorithm when the size of the search space and the size of the dataset increase. To do this, we randomly generate multiple synthetic datasets – using the `Sphere` function – with different number of objects (i.e., 100, 1000, 10000, 100000) and different number of attributes (i.e., 1, 2, 4, 6, 8, 10). In each dataset, each attributes can take 10 different values. We run `OSMIND` on each dataset with an allotted time of 24 hours, using the quality measure q^a_{mean} with both $a = 0.5$ and $a = 1$ and report the results – in seconds – in Table 4.4.

It seems like the running time of the algorithm caps – or barely increases anymore – once a certain dataset size has been reached. Furthermore, we can see that computing the optimal subgroup with $a = 1$ is much faster than with $a = 0.5$, especially as the number of attributes increases. This is due to the fact that as the value of a increases, so does the size of the interesting subgroups. Therefore, when we increase the value of a, we find very

Table 4.3: Comparison: Closure on the positives (COTP) vs Normal closure (NC) and Tight improved (TI) vs Tight base (TB). "-" means execution time >72h.

Dataset	a	COTP	NC	Gain (÷)	TI	TB	Gain (%)
Bolt	0.5	25	118	4.7	0.0062	0.0078	20.5
	1	16	299	19	0.0042	0.0055	23.6
Basketball	0.5	143037	3014506	21	80.5	104	22.6
	1	42548	1121798	26	30.5	39.3	22.4
Airport	0.5	387	12042	35	0.17	0.19	10.5
	1	57	10055	176	0.033	0.037	10.8
Body Temp	0.5	795	1199	1.5	0.53	0.73	27.4
	1	570	865	1.5	0.47	0.53	11.3
Pollution	0.5	100776	-	-	23.9	25	4.4
	1	1289	41662411	32321	0.376	0.408	7.8

Table 4.4: Study of the scalability of OSMIND with regard to the size of the dataset and the number of attributes on synthetic datasets generated with the Sphere function. "-" means execution time >24h.

Dataset \ Features		1	2	4	6	8	10
Sphere_{100}	0.5	0.02	0.03	3.2	50	276	1077
	1	0.01	0.02	1.1	18	40	118
Sphere_{1000}	0.5	0.16	0.41	183	5462	-	-
	1	0.04	0.35	33	904	-	-
Sphere_{10000}	0.5	0.07	0.4	182	5469	-	-
	1	0.04	0.33	33	903	-	-
Sphere_{100000}	0.5	0.06	0.36	182	5471	-	-
	1	0.03	0.31	33	904	-	-

good subgroups in higher levels of the search space. On the contrary, when a is set to lower values, we need to explore the search space more in-depth to find good patterns. We can also see that the number of attributes is by far the parameter with the strongest influence on the running time of OSMIND. Indeed, once we reach 8 attributes, the size of the search space becomes so large that the algorithm is unable to discover an optimal subgroup in the allotted amount of time for datasets with more than 1000 objects. For 10 attributes, it is even worse since we can not find an optimal subgroup for datasets with more than 100 objects. Finally, we want to illustrate the well-known problem of "pattern flooding" in data mining, which is even worse for numerical data. To do this, we retain the smallest dataset on which the algorithm returned an optimal subgroup in the allotted amount of time, i.e., Sphere_{100}. We then generate a second dataset made of 100 objects too, but this time with a domain size of 20 for each attribute. Our goal is to show the influence of the domain size on the running time of the algorithm with both $a = 0.5$ and $a = 1$. Results – in seconds – are available in Table 4.5. Doubling the domain size leads to multiplying the running time by

over 20 when 10 attributes are considered, even though we are only working with a dataset made of 100 objects. If larger datasets, higher number of attributes, or large domains were considered, the results would get even worse. Indeed, when considering $\texttt{Sphere}_{100}$ with a domain size of 10, there are 2.5×10^{17} possible subgroups with 10 attributes. However, when considering $\texttt{Sphere}_{100}$ with a domain size of 20, there are 1.7×10^{23} possible subgroups with 10 attributes. By simply doubling the domain size of the attributes, the size of the search space is multiplied by close to 1 million. Even though our algorithm exploits state-of-the-art techniques to compress and prune the search space, Optimal Subgroup Discovery on datasets with numerous attributes – or with fewer attributes but with large domains of values – is simply not in the realm of possibilities with current knowledge and technology. Notice that in that case, we can go for methods like SD-Map⋆ or the heuristic approach MCTS4DM (Bosc et al., 2018).

Table 4.5: Study of the effects on running time of the attributes domain size as the number of features increases.

		1	2	4	6	8	10
Domain size = 10	0.5	0.02	0.03	3.2	50	276	1077
	1	0.01	0.02	1.1	18	40	118
Domain size = 20	0.5	0.04	0.21	11	598	4533	23920
	1	0.01	0.04	0.59	133	246	895

Let us discuss the added-value of OSMIND w.r.t. SD-Map⋆, i.e., the reference algorithm for an exhaustive strategy with numerical target labels. We compare the quality of the best found subgroup with each method on the 5 datasets of Table 4.1 when using the quality measure q^a_{mean} with $a = 0.5$. Regarding SD-Map⋆, a prior discretization of numerical attributes is needed. To obtain fair results, we evaluate several discretization techniques with different numbers of cut-points (2, 3, 5, 10, 15 and 20) for SD-Map⋆ and we retain only the best solution that is compared to the OSMIND results. Selected discretization techniques are *Equal-Width*, *Equal-Frequency* and *K-Means*. The comparison is in Fig. 4.4. Our algorithm provides subgroups of higher quality for all datasets, and this no matter the applied discretization for SD-Map⋆. We infer that the information loss inherent to the attribute discretization is responsible for the poorer results obtained with SD-Map⋆.
In the following experiments, we consider Sphere datasets made of 10 numerical attributes $\{x_1, ...x_{10}\}$ and 1 objective function f_1 to minimize. Since the goal of our algorithm is to mine for subgroups with a maximized mean label value, we transform the objective function of Sphere datasets into a function to be maximized by multiplying it by -1 and normalizing the resulting values. Next, we compare the run times of OSMIND and SD-Map⋆ to quantify the cost of optimality. We generate datasets – made of Sphere objects – with sizes ranging from 10 to 500 objects. While SD-Map⋆ and OSMIND both find the optimal subgroup in the same amount of time for small datasets, the execution time of OSMIND grows exponentially with the number of objects contrary to that of SD-Map⋆ (>86400 seconds for OSMIND vs <1 second for SD-Map⋆ with 500 objects).

Let us now use the Sphere function to successively generate datasets made of 10, 50, 100, 200 objects and we observe the quality of the best subgroup returned for the quality measure q^a_{mean} when $a = 1$. Regarding SD-Map⋆, we use again the discretization that produces the

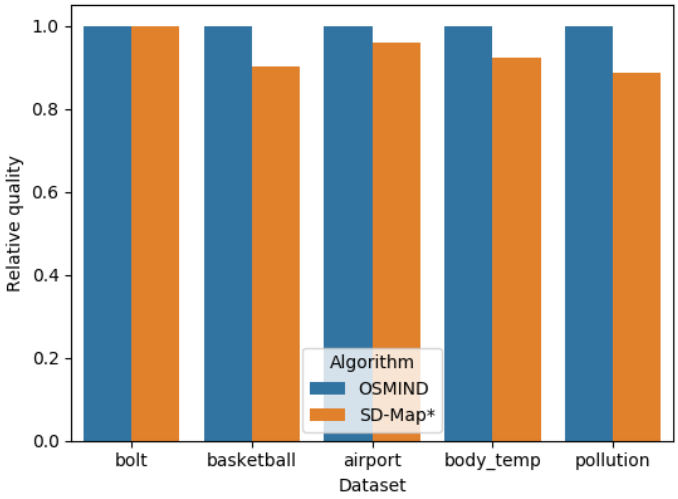

Figure 4.4: Comparison of the best subgroup quality.

best subgroup. Fig. 4.5 depicts the relative quality of the best subgroup returned by each algorithm for different dataset sizes. With a very small dataset, `SD-Map*` finds a close to optimal subgroup despite the use of discretization. However, as datasets get larger, `SD-Map*` returns consistently 50% to 70% worse results. We can conclude that `OSMIND` and `SD-Map*` provide different trade-offs between execution time and quality of the results. With `OSMIND`, we find subgroups of optimal quality, at the expense of running time, while `SD-Map*` provides subgroups of decent quality almost instantly.

Another important qualitative aspect concerns the descriptions of the optimal subgroups found by `OSMIND` and `SD-Map*` with q^a_{mean} when $a = 1$. Table 4.6 depicts these descriptions for our previous dataset made of 100 `Sphere` objects. Besides the higher quality of the

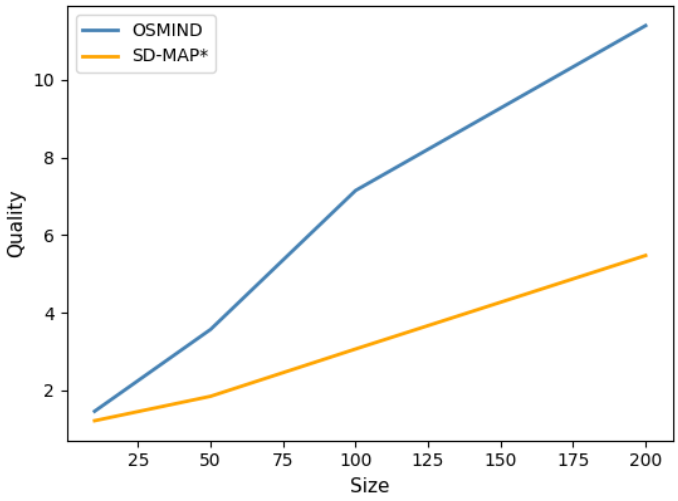

Figure 4.5: Comparison of the best subgroup quality w.r.t. number of objects.

subgroup returned by OSMIND, its description also enables the extraction of much more information than the description obtained with SD-Map⋆. In fact, where SD-Map⋆ only offers a strong restriction on attribute x_6, OSMIND provides actionable information on 7 of the 10 considered attributes.

Table 4.6: Comparison between descriptions of: the overall dataset (DS), the optimal subgroup returned by OSMIND, the optimal subgroup returned by SD-Map⋆. "-" means no restriction on the attribute compared to DS, Q and S denote respectively the quality and size of the subgroup.

Subgroup	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	Q	S
DS	[-5.12,5.12]	[-5.12,5.12]	[-5.12,5.12]	[-5.12,5.12]	[-5.12,5.12]	[-5.12,5.12]	[-5.12,5.12]	[-5.12,5.12]	[-5.12,5.12]	[-5.12,5.12]	0	100
OSMIND	-	[-5.12,3.98]	[-5.12,3.98]	-	-	[-3.98,5.12]	[-3.98,3.98]	[-2.84,5.12]	[-3.98,5.12]	[-3.98,5.12]	7.15	41
SD-Map⋆	-	-	-	-	-	[-1.7,1.7]	-	-	-	-	3.06	31

Let us finally study the relevance of the optimal subgroup found by OSMIND on the Sphere dataset made of 200 objects. We can first check that OSMIND enables the discovery of a subgroup maximizing f_1. Next, we validate the interpretability and actionability of the returned results. Table 4.7 features a comparison between the interval pattern of the overall dataset and that of the optimal subgroup returned by OSMIND. These results illustrate the capacity of OSMIND to discover a subgroup which optimizes f_1 (0.61 vs 0.49). Finally, we can exploit the description of the optimal subgroup to easily generate new objects with more optimized values of f_1.

Table 4.7: OSMIND results: Interval patterns of the overall dataset (DS), the optimal subgroup returned (OS), and average value of f_1 for each subgroup.

Subgroup	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	f_1
DS	[-5.12,5.12]	[-5.12,5.12]	[-5.12,5.12]	[-5.12,5.12]	[-5.12,5.12]	[-5.12,5.12]	[-5.12,5.12]	[-5.12,5.12]	[-5.12,5.12]	[-5.12,5.12]	0.49
OSMIND	[-5.12,3.98]	[-3.98,5.12]	[-5.12,5.12]	[-3.98,3.98]	[-5.12,5.12]	[-3.98,3.98]	[-5.12,5.12]	[-5.12,5.12]	[-5.12,3.98]	[-5.12,5.12]	0.61

Other examples are discussed in Chapter 6.

4.4 Conclusion

We investigate the optimal subgroup discovery with respect to a quality measure in purely numerical data. We motivated the reasons why existing methods achieve suboptimal results by requiring a discretization of numerical attributes. The OSMIND algorithm enables Optimal Subgroup Discovery without such loss of information. The empirical evaluation has illustrated the added-value and the exploitability of the OSMIND algorithm when compared with the reference algorithm SD-Map⋆. From an applicative perspective, this method has already been exploited for plant recipe optimization in controlled environments on synthetic and real-life data in Chapter 6. From an algorithmic perspective, future work concerns the enhancement of OSMIND scalability for high-dimensional datasets. Moreover, it would be interesting to investigate how to exploit some sequential covering techniques for computing not only an optimal subgroup but a collection of non-redundant optimal subgroups.

Chapter 5

Exceptional Model Mining to support Multi-Objective Optimization

Exceptional Model Mining (EMM) is a framework that generalizes subgroup discovery from labeled data. In EMM, we look for subsets of objects – subgroups – whose models deviate significantly from the same models fitted on the whole dataset. Quite different types of models and thus quality measures can be considered. In many MOO application settings, we have at hand numerical and categorical data for descriptive attributes and a number of targets, i.e., the functions to be optimized. In this chapter, we investigate methods that exploit Exceptional Pareto Front Mining (EPFM) in such data. A first method, called Exceptional Pareto Front Deviation Mining (EPFDM) exploits the deviation between the shape of the Pareto front left by the absence of a subgroup of objects compared with the Pareto front on the whole dataset. We also develop an approach called Exceptional Pareto Front Approximation Mining (EPFAM), whose goal is the discovery of models that approximate exceptionally well the true Pareto front. We discuss in detail the design of a generic quality measure for EPFM. Finally, we propose in-depth empirical studies of both EPFDM and EPFAM, and we discuss an application scenario to hyperparameter optimization in Machine Learning.

5.1 Introduction

In EMM, we look for subgroups whose models deviate significantly from the same models fitted on the entire dataset. Where subgroup discovery is inherently limited to a unique target concept, EMM is able to handle data where two or more targets exist, enabling the discovery of more complex interactions between variables.
Examples of complex interactions between variables can be found in multi-objective optimization. Our MOO setting concerns knowledge discovery about the sets of descriptive attributes when we want to optimize simultaneously all the numerical targets. We consider here Pareto optimization, which involves not one, but a set of equal solutions.
An interesting use case for multi-objective optimization concerns the design of better plant growth recipes in controlled environments. Growth optimization is intrinsically an MOO problem. Indeed, in such controlled environments, when trying to optimize the yield, the size, or the taste of plants, other variables like the energy cost have to be taken into account. Therefore, optimizing recipes means finding the best trade-offs between several concurrent objectives. This is difficult: in growth recipe optimization, the underlying model is unknown and experiments are limited due to time and cost constraints, making it impossible to exploit typical MOO approaches. There is a need for methods that would support the discovery of relevant and exploitable information in such MOO problems.
We therefore investigate the cross-fertilization between EMM and MOO by designing a generic model class for Exceptional Pareto Front Mining (EPFM). We discuss the added value of distance-based and volume-based measures for Exceptional Pareto Front Deviation Mining (EPFDM) that enables the discovery of interesting deviation models. We introduce a generic quality measure and investigate different ways to make it as robust as possible. While EPFDM can be used as an exploratory analysis tool to discover interesting nuggets of knowledge, it is not easily usable to generate new and improved solutions. Therefore, we design an original method called Exceptional Pareto Front Approximation Mining (EPFAM). It supports the discovery of subgroups whose Pareto front approximates exceptionally well the true Pareto front, and whose descriptions can be exploited to generate Pareto optimal solutions with higher probability. Although some important changes have to be made to move from exceptional deviation mining towards exceptional approximation mining, most of the results related to pattern relevancy for EPFDM can be easily reused or adapted for EPFAM. The added value of both EPFDM and EPFAM is investigated thanks to in-depth quantitative and qualitative empirical studies. Among others, we discuss a use case about hyperparameter optimization in Machine Learning and show the relevance of our methods in this setting. The actionability of both EPFDM and EPFAM has also been validated for plant growth recipe optimization in Chapter 6.
Part of this chapter (EPFDM) has been published in the Proceedings of the 2021 SIAM International Conference on Data Mining (SDM) (Millot et al., 2021). Most of this chapter is also under review for publication in the Data Mining and Knowledge Discovery (DAMI) journal (submitted in March 2021). For reproducibility purposes, all datasets and source code are made available in `https://bit.ly/3ilhir5`.
The remaining of this chapter is organized as follows. Section 5.2 details our contributions to EPFDM. We then introduce our contributions to EPFAM in Section 5.3. A generic and robust quality measure is introduced in Section 5.4. In Section 5.5, we detail in-depth empirical studies of EPFDM and EPFAM. Finally, Section 5.6 concludes.

5.2 Mining Exceptional Pareto Front Deviations

5.2.1 Approach

We consider here typical datasets for EMM that are composed of a set of attributes, and several numerical targets, akin to the one presented in Section 2.6. To illustrate our work and its related concepts, we consider the `Fonseca` dataset introduced in Chapter 2.
We want to build a model class for EMM in a MOO setting: we propose to look for exceptional Pareto front deviations. In a given dataset, we define the true Pareto front – denoted $PF_{dataset}$ – as the set of all non-dominated objects over the whole data. In typical EMM approaches, an exceptional model is computed directly on the objects of the subgroup. Then a quality measure is used to measure the deviation between the model built on the subgroup and the same model built on the whole dataset. We assume that we have to work with objective minimization only. When a maximization problem occurs, it is transformed into a minimization one by multiplying the function by -1.

Our goal hereafter is to capture subgroups representing local phenomena with the highest influence on the shape of $PF_{dataset}$, meaning that we need to measure the effects on $PF_{dataset}$ of removing these objects from the data. Therefore, when a subgroup is generated, we remove all its objects from the dataset. Then, we compute the new Pareto front PF_{model} on the remaining data, i.e., the *complement* of the subgroup. Finally, we can compute the deviation between $PF_{dataset}$, the Pareto front for the dataset, and PF_{model}. This first approach to EPFM, based on the discovery of subgroups creating large deviations in the shape of the true Pareto front, is called Exceptional Pareto Front Deviation Mining (EPFDM).

Let us first define which objects of each Pareto front are taken into account when computing distances between Pareto fronts.

Definition 21. *Given two Pareto fronts PF_{target} and $PF_{reference}$, the Partial Pareto Front $PPF(PF_{target}, PF_{reference})$ is equal to:*

$$\{x \in PF_{target} | \nexists y \in PF_{reference}, x = y\}$$

The PPF is defined as the subset of objects of a Pareto front that are not in the set of objects of the other Pareto front. A PPF can be computed either for $PF_{dataset}$ by keeping its objects which are not in PF_{model} or for PF_{model} by keeping its objects which are not in $PF_{dataset}$. Figure 5.1 depicts the PPFs of PF_{model} (left) and $PF_{dataset}$ (right). In Figure 5.1 (left), the PPF of the model – denoted by PPF_model – is the set of objects of PF_model (i.e., the Pareto front of the data that is left once the subgroup has been removed) which do not belong to the Pareto front of the dataset – denoted by $PF\_dataset$. Conversely, in Figure 5.1 (right), the PPF of the dataset – denoted by $PPF\_dataset$ – is the set of objects of $PF\_dataset$ which do not belong to the Pareto front of the model – denoted by PF_model. In our figures, ND stands for normal data point, SG denotes a subgroup, $PF\_dataset$ represents the best known Pareto front and PF_model represents the Pareto front of a subgroup.

5.2.2 Designing Quality Measures for EPFDM

Measuring Deviations between Pareto fronts

Multi-objective optimization requires algorithms that approximate as well as possible the true Pareto front for any given problem. Many quality measures have been introduced to estimate the quality of the computed Pareto front compared to the true Pareto front or to an ideal point (Li and Yao, 2019). Thanks to some of these measures, the distance between two Pareto fronts can be computed. In traditional multi-objective optimization measures, only the non-symmetrical distance from either the true Pareto front to the approximate Pareto front (e.g., Inverted Generational Distance (Li and Yao, 2019)) or from the approximate Pareto front to the true Pareto front (e.g., Generational Distance (Li and Yao, 2019)) is computed. However, (Schutze et al., 2012) shows that taking into account both distances provides measures that are more resilient to outliers and uncommonly shaped Pareto fronts. Therefore, we consider measures that consider both the distance between the partial Pareto front of the subgroup PPF_{model} and the Pareto front of the overall dataset $PF_{dataset}$, and the distance between the partial Pareto front of the overall dataset $PPF_{dataset}$ and the Pareto front of the subgroup PF_{model}. Then, the largest one is kept as the true distance. It is important to normalize each of the objectives such that they contribute equally to the measure. We normalize each of them to get a value between 0 and 1 using the standard scaling $x_j^{'} = (x_j - min_j)/(max_j - min_j)$, where min_j and max_j are respectively the minimum and maximum of Objective j.

Our measures are based on the popular Hausdorff Distance that estimates how far two subsets of a metric space are from each other using Euclidean distances: informally, it is defined as the largest of all the distances from a point in one subset to its closest point in the other subset.

Definition 22. *The Hausdorff Distance (HD) between PF_{model} and $PF_{dataset}$ is defined as:*

$$HD(PF_{model}, PF_{dataset}) = max(max(mind(PPF_{model}, PF_{dataset})), \\ max(mind(PPF_{dataset}, PF_{model})))$$

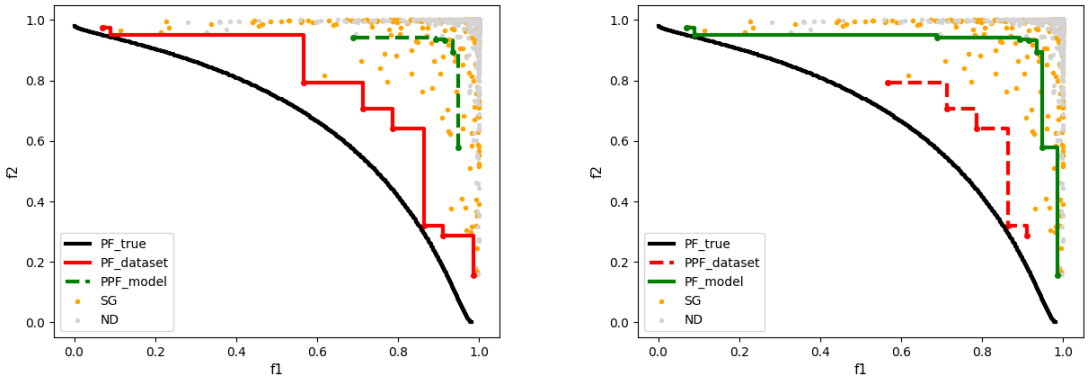

Figure 5.1: Partial Pareto fronts of PF_{model} (left) and $PF_{dataset}$ (right) in `Fonseca`.

The Median Hausdorff Distance (MHD) between PF_{model} and $PF_{dataset}$ is defined as:

$$MHD(PF_{model}, PF_{dataset}) = max(med(mind(PPF_{model}, PF_{dataset})), \\ med(mind(PPF_{dataset}, PF_{model})))$$

where mind computes the minimal Euclidean distance from each point of the partial Pareto front to the other Pareto front, max returns the largest value in a set of distances and med returns the median value in a set of distances.

Let us now consider a modified version of the Averaged Hausdorff Distance (AHD) from (Schutze et al., 2012).

Definition 23. *The Averaged Hausdorff Distance $AHD(PF_{model}, PF_{dataset})$ between PF_{model} and $PF_{dataset}$ is:*

$$max\left(\frac{1}{N}\sum_{i=1}^{N}(mind(PPF^{i}_{model}, PF_{dataset})), \right. \\ \left. \frac{1}{M}\sum_{i=1}^{M}(mind(PPF^{i}_{dataset}, PF_{model}))\right)$$

where N is the number of objects of PPF_{model} and M is the number of objects of $PPF_{dataset}$. mind computes the minimal Euclidean distance from object i of the partial Pareto front to the other Pareto front. The average of all minimal distances is then computed. Finally, max takes the largest distance of the two.

Although our work has lead us to investigate measures that consider the distance between solution sets, MOO literature contains numerous ways of estimating the quality of a solution set, including dominance-based, region-division based, and volume-based quality indicators (Li and Yao, 2019).
We propose to exploit a new volume-based measure taken from the MOO literature, the so-called Hypervolume (HV). Contrary to previously introduced distance-based measures, HV does not need a reference set, but a reference point to compute the quality of a given Pareto front. In other terms, the concept of Partial Pareto Front is only relevant for distance-based measures, and will not be used with HV. Its formal definition can be found in Section 3.4. Informally, the Hypervolume value of a Pareto front can be seen as the volume of the area enclosed by the Pareto front and the specified reference point. HV usually takes values between 0 and 1. Typically, the reference point corresponds to the Nadir point – i.e., the vector of the worst possible value of each objective in the optimal true Pareto front – which is impossible to precisely estimate in most scenarios. Figure 5.2 (left) depicts an example of the Nadir point defined according to the Pareto front in `Fonseca`. The figure also depicts an example of HV computed between the Pareto front of the dataset and the previously defined Nadir point. One issue arises from estimating the reference point this way: numerous objects of the dataset lie outside the area enclosed by the Pareto front and the Nadir point. This is problematic for us since when we mine for subgroups, any object could be part of the Pareto front of a model, even those that lie outside the enclosed area in Figure 5.2. For this reason, we define our own version of a reference point, that ensures that no object lies outside the enclosed area.

Definition 24. *The reference point $r((G, M, T))$ of a given dataset is defined by:*

$$r((G, M, T)) = \langle max(T_i) \rangle_{i \in \{1,\ldots,|T|\}}$$

Informally, the reference point of a dataset is the vector composed of the worst value for each objective in the overall dataset. Figure 5.2 (right) depicts a comparison between our reference point and the Nadir point. This novel reference point ensures that the HV can be properly computed for any subset of a given dataset. Next, we detail how the HV of a given subgroup is computed in EPFDM. We look for subgroups whose removal produces exceptional deviations of the Pareto front. Therefore, we need to look for subgroups that create the largest differences between the HV of the dataset, and the HV of the complement of the subgroups.

Definition 25. *The HV of a given subgroup, denoted by HV_{dev}, is defined as:*

$$HV_{dev}(PF_{model}, PF_{dataset}) = 1 - \frac{HV(PF_{model})}{HV(PF_{dataset})}$$

This way, higher values of HV_{dev} mean larger deviations of the Pareto front, and the measure is normalized with values between 0 and 1.

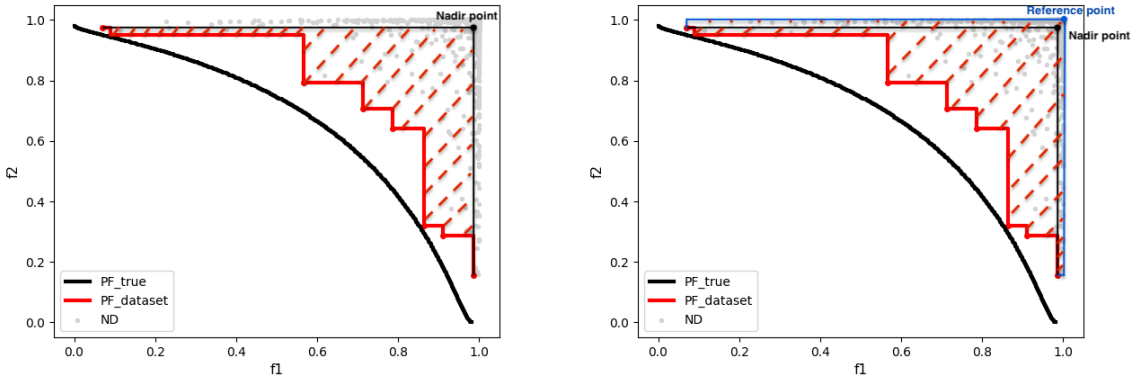

Figure 5.2: Hypervolume of `Fonseca` with the Nadir point (left) and our reference point (right).

5.2.3 Algorithm

Our enumeration algorithm is based on a top-K beam search (Duivesteijn et al., 2016). The pseudocode is available in Algorithm 2. In a simple implementation of beam search, subgroups can be evaluated multiple times due to its candidate generation process. In our beam search, candidates in the beam can only be evaluated once, leading to a small gain in efficiency. The evaluation part of the process is by far the most costly here. To compute the Pareto front of a subgroup, we employ a greedy approach where each object not in the subgroup is compared to all the objects not in the subgroup to check whether it is dominated by at least one other object. If it is not dominated by any other object, we add it to the Pareto front. Finally,

we implement a simple pruning technique that leads to a large reduction in the number of subgroups that need to be evaluated. For a subgroup to be interesting, its removal has to create a deviation in the shape of the true Pareto front. Due to the nature of the dominance relation, the removal of any object not on the true Pareto front cannot lead to a change in the Pareto front. It means that only subgroups that contain at least one object that belongs to the true Pareto front are of interest. As a result, during our search, we ignore any subgroup and their specializations if it does not contain an object that belongs to the true Pareto front.

Algorithm 2 Beam search for Top-K EPFDM

Input: Dataset D, quality measure q, beam width w, search depth dp, result set size k, global pareto front pf

Output: Priority queue Q

```
1: current_depth ← 0
2: candidate_queue ← new queue()
3: candidate_queue ← candidate_queue.insert({})
4: Q ← new priority_queue(k)
5: while (current_depth < dp) do
6:     beam ← new priority_queue(w)
7:     while (candidate_queue ≠ ∅) do
8:         lst_candidates_lvl ← specialize(candidate_queue.dequeue())
9:         for (pattern ∈ lst_candidates_lvl) do
10:            extent ← computeExtent(pattern)
11:            if (extent.isNotDuplicate() and extent.hasObj(pf)) then
12:                complement_extent ← computeComplement(extent)
13:                pareto_front_extent ← computeParetoFront(complement_extent)
14:                quality_extent ← q(pareto_front_extent)
15:                if (quality_extent > worstPattern(beam)) then
16:                    beam.insert(extent, quality_extent, pareto_front_extent)
17:                end if
18:                if (quality_extent > worstPattern(Q)) then
19:                    Q.insert(extent, quality_extent, pareto_front_extent)
20:                end if
21:            end if
22:        end for
23:    end while
24:    while (beam ≠ ∅) do
25:        candidate_queue.insert(beam.dequeue())
26:    end while
27:    current_depth = current_depth + 1
28: end while
29: return Q
```

5.3 Mining Exceptional Pareto Front Approximations

5.3.1 Approach

Although EPFDM can be used as an exploratory data analysis tool to discover interesting pieces of knowledge, such as (i) subspaces of the current Pareto front where data could be missing, (ii) subsets of better or worse solutions of the Pareto front, (iii) anomalous parts of the Pareto front, it lacks the capability of providing information that directly enables the design of better solutions. Therefore, we would like a method that can better support the discovery of actionable insights to generate higher-quality solutions for MOO problems. Therefore, we investigate the discovery of exceptionally good approximations of the true Pareto front, called Exceptional Pareto Front Approximation Mining (EPFAM). It provides a nice solution to our problem: with exceptional approximations supported by subgroups and their understandable descriptions, we can generate new, close to Pareto optimal, solutions for a given MOO problem. When we lack expertise, instead of exploring new solutions more or less randomly, hoping for them to offer good trade-offs, we can exploit a given subgroup description to generate high-quality solutions with a higher probability.
Our goal hereafter is to discover subgroups whose Pareto front shape is as similar as possible to that of $PF_{dataset}$. To do this, when a subgroup is generated, we compute its Pareto front PF_{model}. Then, we can assess how good an approximation PF_{model} is with regard to $PF_{dataset}$. Now that we have defined how models are computed in EPFAM, we need measures to assess their quality.

5.3.2 Designing Quality Measures for EPFAM

Comparing the Shape of Pareto Fronts

While the use of distance-based measures makes sense in the case of EPFDM, it is not always relevant for EPFAM. Indeed, in critical cases where the Pareto front of the subgroup lies entirely on the true Pareto front, the computed distance between the two would either be 0 (e.g., if we do not use the concept of Partial Pareto Front) or it would be an irrelevant value (e.g., in the case where we use Partial Pareto Fronts) non-representative of the actual distance between the fronts. Therefore, we forget distance-based measures and we focus on volume-based measures like HV, which can better represent how similar two Pareto fronts are. The HV of the true Pareto front and its reference point are calculated as detailed in Section 5.2.2. Let us detail how the HV of a given subgroup is computed in EPFAM. We now look for subgroups whose Pareto front is an exceptional approximation of the true Pareto front. Therefore, we need to look for subgroups whose HV is as close as possible to that of the dataset.

Definition 26. *The HV of a given subgroup, denoted HV_{approx}, is defined as:*

$$HV_{approx}(PF_{model}, PF_{dataset}) = \frac{HV(PF_{model})}{HV(PF_{dataset})}$$

This way, higher values of HV_{approx} mean better approximations of the Pareto front, and the measure is normalized with values between 0 and 1.

5.3.3 Algorithm

For the computation of top-K EPFAM, a slightly modified strategy from the one introduced in Section 5.2 for EPFDM can be used. The pseudocode is available in Algorithm 3. First, instead of computing the Pareto front of the complement for each subgroup, we compute the Pareto front of the subgroups themselves. Second, in EPFAM, the pruning of the subgroups which do not contain any object that belongs to the true Pareto front is only applied if a minimum support constraint is used.

Algorithm 3 Beam search for Top-K EPFAM

Input: Dataset D, quality measure q, beam width w, search depth dp, result set size k, global pareto front pf
Output: Priority queue Q

```
1: current_depth ← 0
2: candidate_queue ← new queue()
3: candidate_queue ← candidate_queue.insert({})
4: Q ← new priority_queue(k)
5: while (current_depth < dp) do
6:     beam ← new priority_queue(w)
7:     while (candidate_queue ≠ ∅) do
8:         lst_candidates_lvl ← specialize(candidate_queue.dequeue())
9:         for (pattern ∈ lst_candidates_lvl) do
10:            extent ← computeExtent(pattern)
11:            if (extent.isNotDuplicate()) then
12:                if (extent.hasObj(pf) or (!extent.hasObj(pf) and !useMinSup())) then
13:                    pareto_front_extent ← computeParetoFront(extent)
14:                    quality_extent ← q(pareto_front_extent)
15:                    if (quality_extent > worstPattern(beam)) then
16:                        beam.insert(extent, quality_extent, pareto_front_extent)
17:                    end if
18:                    if (quality_extent > worstPattern(Q)) then
19:                        Q.insert(extent, quality_extent, pareto_front_extent)
20:                    end if
21:                end if
22:            end if
23:        end for
24:    end while
25:    while (beam ≠ ∅) do
26:        candidate_queue.insert(beam.dequeue())
27:    end while
28:    current_depth = current_depth + 1
29: end while
30: return Q
```

5.4 A Generic Quality Measure

Being able to measure the deviation from the true Pareto front may not be enough to mine interesting subgroups. In the literature about EMM quality measures, we usually get measures with the following form: the quality of the subgroup is multiplied by its generality. Indeed, in typical EMM, discovering unusual distributions is easily achieved with small subgroups, therefore there is a need to optimize the generality (i.e., cover) of the discovered subgroups. In the context of EPFDM, we face the opposite problem: unusual distributions are easily achieved with large subsets of the data (e.g., if we find a subgroup covering 80% of the data, it is very likely that its removal will create a large deviation in the Pareto front). Despite their high quality, such subgroups are not interesting. Figure 5.3 (left) depicts an example of this phenomenon. Therefore, we need to optimize the locality of the subgroups. Furthermore, small subgroups that modify only a small part of the true Pareto front when removed are also not desirable. Indeed, an issue can arise when either outliers are a part of the true Pareto front or when the density of objects is very low close to some part of the Pareto front. In such cases, the removal of subgroups with very few objects on the true Pareto front can create unwanted large deviations in the Pareto front of the model leading to overfitting and trivial subgroups. Figure 5.3 (right) depicts an example of this phenomenon. Therefore, in EPFDM, we might also be interested in the optimization of the generality of the subgroup with regard to the true Pareto front, i.e., the *generality* of the model. To summarize, given the previously defined deviation measures, we can get either very large or very small subgroups.

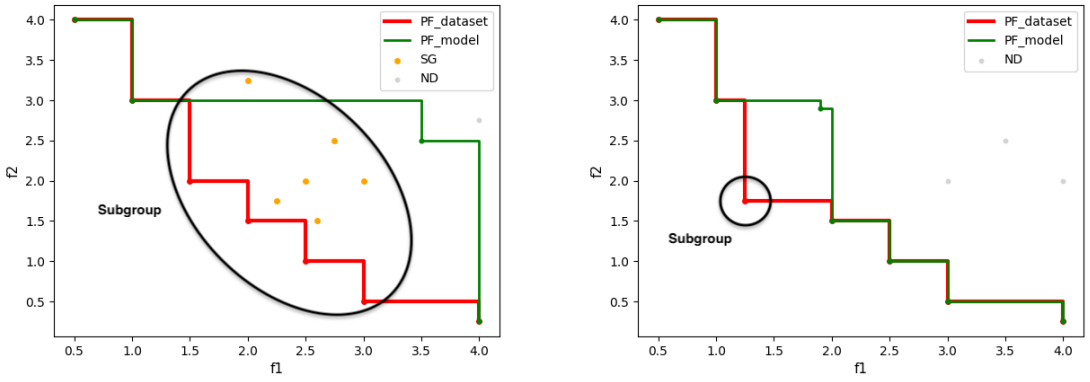

Figure 5.3: Low entropy (left) and large (right) subgroups.

To deal with the first issue (i.e., unwanted large subgroups), let us introduce a locality constraint.

Definition 27. *The locality constraint of a subgroup p is:*

$$Locality(p) = 1 - \left(\frac{n}{N}\right)$$

where N is the total number of objects of the dataset and n is the number of objects of p.

This locality constraint favors smaller subgroups over larger ones. It is especially useful for cases where objects can be removed from a subgroup without modifying the Pareto fronts.

However, this constraint might be too strict in some application cases where larger subgroups might be more desirable. Therefore, we add a factor that tunes the importance of the locality constraint.

Definition 28. *The locality constraint of a subgroup p, with its importance factor, is:*

$$Locality^a(p) = \left(1 - \left(\frac{n}{N}\right)\right)^a,\ a \in [0,1]$$

where N is the total number of objects of the dataset, n is the number of objects of p, and a an importance factor.

To deal with the second issue, we propose several solutions. First, let us use the entropy of the split between the objects of the Pareto front which are not part of the subgroup, and those who are. We also want control over the importance of the entropy, therefore we introduce a factor that tunes its importance.

Definition 29. *The entropy of a subgroup p is:*

$$Entropy^b(p) = \left(-\frac{n}{N}\lg\left(\frac{n}{N}\right) - \frac{N-n}{N}\lg\left(\frac{N-n}{N}\right)\right)^b,\ b \in [0,1]$$

where lg *denotes the binary logarithm, N is the total number of objects on the true Pareto front, n is the number of objects of p that belong to the true Pareto front, and b an importance factor.*

The entropy favors balanced splits over unbalanced ones. It returns 0 when the subgroup has no point on the true Pareto front or the subgroup covers the whole true Pareto front. It returns 1 when a perfect 50/50 split is achieved. This way, our quality measure is driven toward finding more relevant subgroups with enough objects on the true Pareto front. Notice that it introduces a bias against subgroups that cover most of the true Pareto front (or the whole Pareto front) although it can be controlled by tuning the importance factor b.
Next, as a second way, let us consider how to use the coverage of the subgroup with regard to the global model. Informally, we compute the percentage of objects of the true Pareto front which are covered by the subgroup. Again, an importance factor can be used to control the weight of the generality.

Definition 30. *The coverage of a subgroup p is:*

$$Coverage^c(p) = \left(\frac{n}{N}\right)^c,\ c \in [0,1]$$

where N is the total number of objects on the true Pareto front, n is the number of objects of p that belong to the true Pareto front, and c an importance factor.

Finally, we can suggest a third way to take into account the generality of the model. We can exploit a minimum support for the percentage of objects of the true Pareto front which are covered by the subgroup. If the minimal support constraint is not satisfied, the subgroup should be discarded.

Definition 31. *A subgroup p is valid with regard to a minimum support $minSupp$ if and only if:*

$$\frac{n}{N} \geqslant minSupp$$

where N is the total number of objects on the true Pareto front, n is the number of objects of p that belong to the true Pareto front, and $minSupp$ is the user-defined minimum support.

We can now define an aggregated measure to take into account the quality of the model, the locality of the subgroup, and the generality of the model.

Definition 32. *Our aggregated quality measure q_{EPFDM} for a subgroup p is defined as:*

$$q_{EPFDM}(p) = Deviation(p) \times Locality^{a}(p) \times Generality(p)$$

where $Deviation(p)$ can be any measure of the deviation quality of p with regard to the true Pareto front, $Locality^{a}(p)$ denotes the locality constraint, and $Generality(p)$ denotes the chosen constraint for the generality of the model (i.e., Entropy, Coverage, or a minimal support constraint). Note that if the minimum support is chosen, then $Generality(p) = 1$.

The generic quality introduced for EPFDM applies to EPFAM, provided that we use an approximation measure instead of a deviation measure in the aggregated measure.

Definition 33. *Our aggregated quality measure q_{EPFAM} for a subgroup p is defined as:*

$$q_{EPFAM}(p) = Approximation(p) \times Locality^{a}(p) \times Generality(p)$$

where $Approximation(p)$ can be any measure of the approximation quality of p with regard to the true Pareto front, $Locality^{a}(p)$ denotes the locality constraint, and $Generality(p)$ denotes the chosen constraint for the generality of the model (i.e., entropy, coverage, or minimal support).

Although we chose to take an interest in distance-based and volume-based measures for the exceptionality of the Pareto fronts in both EPFDM and EPFAM, other quality indicators from the MOO literature, like the dominance-based C indicator (Zitzler and Thiele, 1999) could be considered as well.

5.5 Experiments

Let us now consider experiments on both synthetic and real-life datasets. In the following experiments and unless specified otherwise, the beam width was set to 10, the search depth to 5, the minimum support to 0.1, and the locality factor to 1. These parameters were chosen to explore the search space as much as possible while favoring the small subgroups and keeping the running times in an acceptable range. In the figures, both red and orange objects belong to the best subgroup. When it comes to discretization of the numerical attributes, we apply equal-width discretization using 2, 3, and 5 bins on each dataset and we retain only the one that leads to the best models. Please note that in the application scenario to hyperparameter optimization for Machine Learning, we used a different type of discretization technique which is described in the corresponding section.

5.5.1 Relevance of EPFDM

The goal of this experiment is to show the relevance of our approach to discover exceptional Pareto front deviations. Here, it means finding subgroups whose descriptions in the attribute space provide insights into interesting local parts of the Pareto Front. Let us first use the synthetic dataset `Fonseca` introduced in Chapter 2. We compute the best subgroup found by our algorithm with HV_{dev}, HD, AHD, and MHD. Figure 5.4 depicts the best model for each measure. The best deviation is almost the same for all measures, although the size of the subgroup is quite different. HD finds a model with a large deviation supported by a small subgroup whose description is $\langle x1 \in [-0.8, 0.8], x2 \in [-0.8, 0.8], x3 \in [-0.8, 0.8] \rangle$. Exploiting this subgroup allows for the generation of new objects with a good trade-off between both functions. Since HD, AHD, and MHD mine very similar models – likely due to the fact that all 3 measures are based on the Hausdorff distance –, we only report the best models found with HV_{dev} and HD for the other experiments.

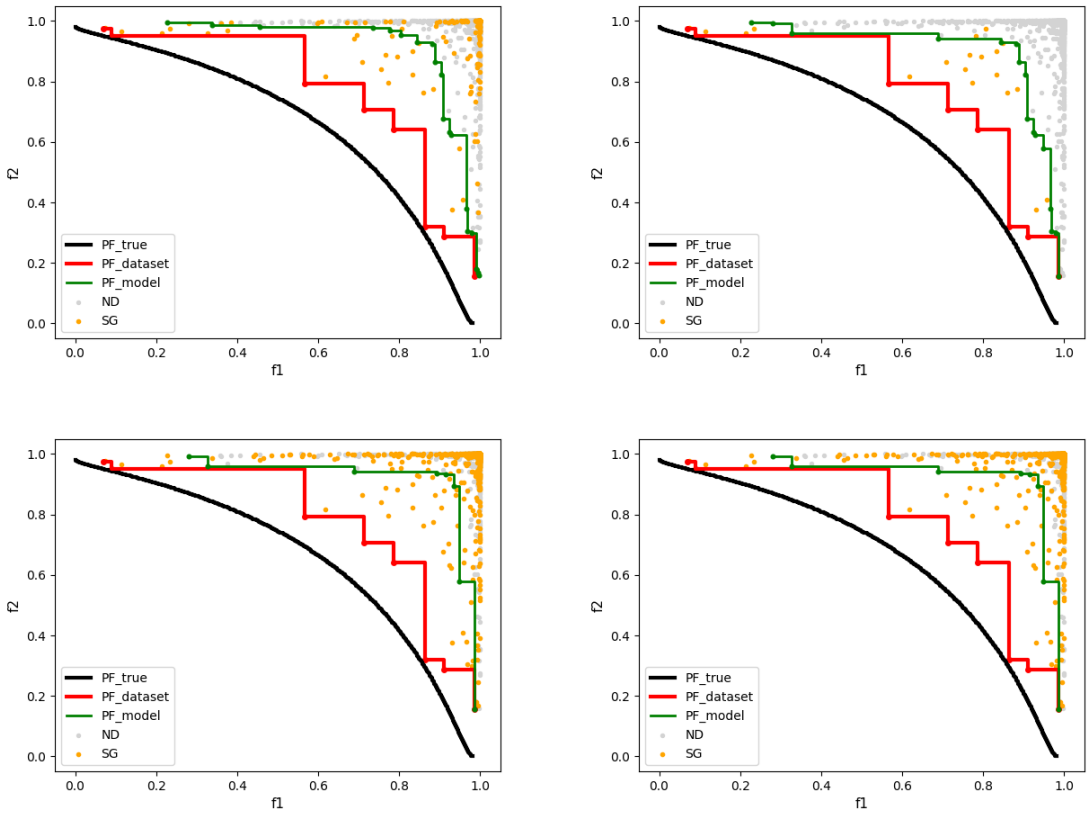

Figure 5.4: EPFDM best deviations on `Fonseca` with respectively HV_{dev}, HD, AHD and MHD.

Let us now consider use cases that are less familiar to the MOO community. Here, the data is limited to the available one (i.e., it cannot be easily extended) and the underlying model is unknown, making it impossible to run something else than a Pareto front computation.

The first dataset – named `Obesity` – concerns data about eating habits and physical conditions of people from Mexico, Peru and Colombia. It was extracted from the UCI repository[1]. It is made of 2111 observations, 14 descriptive variables and 2 objective variables to be optimized: the height that needs to be minimized and the weight that needs to be maximized. In doing so, we want to identify individuals with the worse height-weight trade-off. We compute the best models found with HV_{dev} and HD, and we report the results in Figure 5.5. The deviations found by the two measures look relatively different, although the objects from their subgroup are similar. It can be seen when looking at their respective subgroup descriptions:

$\langle$ Number_main_meals = 3, Frequency_consumption_vegetables = 3, Age $\in$ [13.953,23.4], Family_history_with_overweight = 'yes', Consumption_alcohol = 'Sometimes' $\rangle$

for HV_{dev}, and

$\langle$ Number_main_meals = 3, Frequency_consumption_vegetables = 3, Transportation_used = 'Public_Transportation', family_history_with_overweight = 'yes', Consumption_alcohol = 'Sometimes' $\rangle$

for HD.
Indeed, we notice that the two descriptions differ on only one attribute that can create a large difference in the Pareto front deviation models. We can summarize this difference as follows: the first subgroup represents young people with bad height/weight trade-offs, while the second subgroup might represent poor people who use public transportation and have a worse height/weight trade-off than the rest of the population.

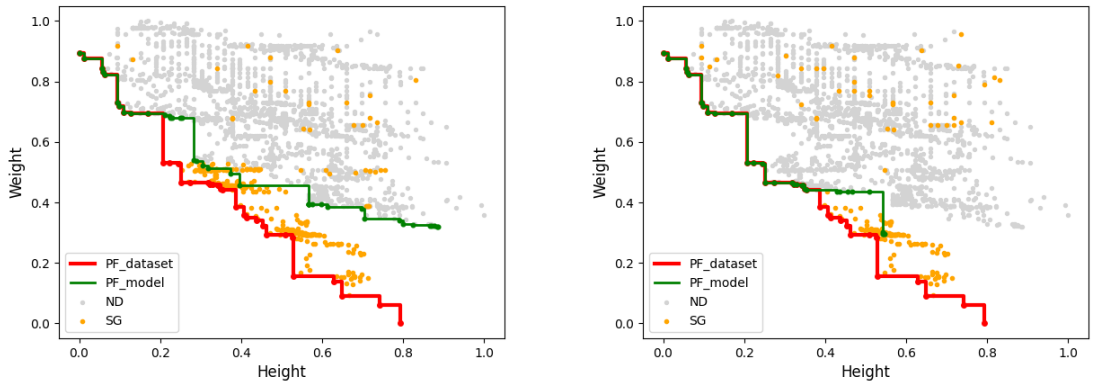

Figure 5.5: EPFDM best deviations on `Obesity` with HV_{dev} (left) and HD (right).

Let us now consider a third experiment about the trade-off between physical and chemical defense in plant seeds. The dataset named `Plants` is made of 163 observations. It was extracted from the Datadryad website[2]. Each observation is described by the family and the

[1] `https://archive.ics.uci.edu/ml/datasets.php`
[2] `https://datadryad.org/stash`

mass of the plant seed. The objective variables are the fiber – physical defense – and the tannin contents – chemical defense – that both need to be maximized. Again, we compute the best subgroup found with the same measures as for `Obesity`. The results, reported in Figure 5.6, show two significantly different deviation models, supported by subgroup descriptions. For HV_{dev}, the subgroup description is $\langle family = `Combretaceae' \rangle$, while for $HD$, the description is $\langle family = `Melastomataceae' \rangle$. Both subgroups represent plant families that seem to provide good local trade-offs between physical and chemical defenses.

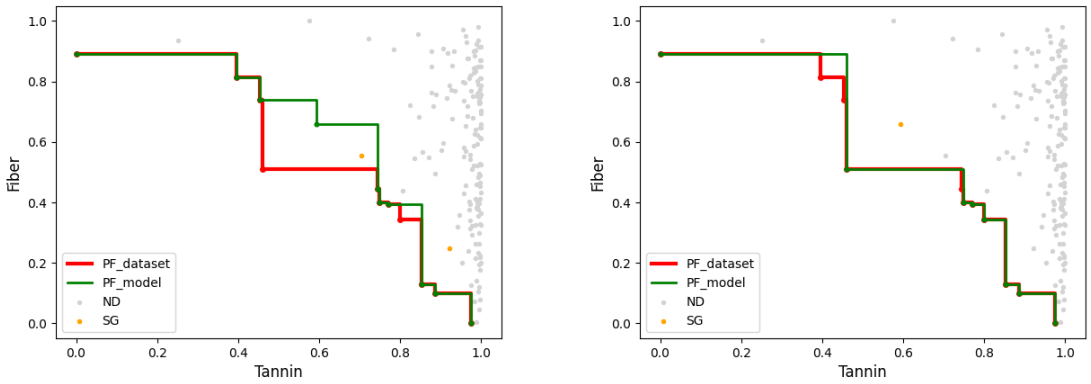

Figure 5.6: EPFDM best deviations on `Plants` with HV_{dev} (left) and HD (right).

The last dataset named `RealEstate` has been extracted from the UCI repository[3]. It concerns over 400 sales of houses in Taiwan between 2012 and 2013. It is made of 4 descriptive variables (latitude, longitude, house age, and number of convenience stores in the living circle on foot) and 2 objective variables: the price of the house and the distance to the closest massive rapid transit station that both need to be minimized. We compute the best subgroup found by our algorithm with HV_{dev} and HD, and we report the results in Figure 5.7. Both measures find the same deviation model, although supported by different subgroups. The subgroup description for HV_{dev} is $\langle latitude \in [24.949, 24.965], longitude \in [121.529, 121.548] \rangle$ while the subgroup description for HD is $\langle latitude \in [24.949, 24.965], number\_of\_convenience\_stores \in [4, 6] \rangle$. The exploitation of these subgroups can lead to finding houses (including their location and characteristics) that offer an interesting trade-off between price and distance to the nearest transport station.

5.5.2 Quantitative Evaluation of EPFDM

As the quality measure introduced offers multiple degrees of freedom, it makes sense to investigate the running time of our process. We first carry out a running time comparison of EPFDM between the 4 proposed deviation measures. To do this, we run our algorithm with standard parameters on the four previous datasets for each measure and we report the results in Table 5.1. The difference in running time between quality measures is small to non-existent on small datasets. However, on a larger dataset like `Obesity`, HV_{dev} seems to be faster than other measures, while MHD and AHD are closer in running time and

[3] `https://archive.ics.uci.edu/ml/datasets.php`

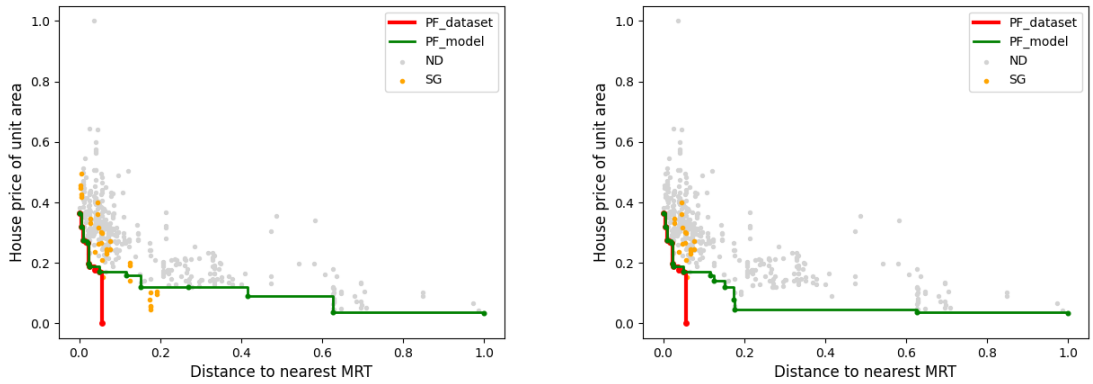

Figure 5.7: EPFDM best deviations on `RealEstate` with HV_{dev} (left) and HD (right).

HD has the highest execution time by a decent margin. These results can be explained in two ways. First, the subgroups returned using HD seem to be smaller than those returned with other measures. Indeed, the most expensive part of the process is the Pareto front computation of each subgroup. Since the Pareto front is computed on the complement of the subgroup, smaller subgroups mean larger complements, leading to higher computation time. Second, the running time differences between measures seem to be pretty small, might not be statistically significant, and could simply be related to more or less efficient implementations. To conclude, choosing one measure or another should not be made according to expected running time efficiency.

Table 5.1: Running time comparison (in seconds) of EPFDM on 4 deviation measures.

Dataset	HV_{dev}	HD	AHD	MHD
`Fonseca`	175	176	176	176
`Obesity`	15426	17866	16690	16215
`Plants`	2.4	2.4	2.4	2.4
`RealEstate`	40	42	42	42

Let us now discuss the running time efficiency when looking for the locality and the generality. Here, we carry out a comparison on our 4 datasets. For each dataset, we use different configurations for the evaluation of both the locality and the generality. For the locality of the subgroup, possible values for the importance of the factor are taken in {0.1,0.5,1}. It is expected that lower (resp. higher) values for the locality factor favor large (resp. small) subgroups. Regarding the minimum support constraint for the generality of the model, possible values are taken in {0.1,0.3,0.5}. When *Coverage* or *Entropy* is selected instead of a minimum support, the values for the factor that controls the importance of the generality of the model are taken in {0.1,0.5,1}. Results of the empirical study are in Table 5.2 where *loc* denotes *Locality*. First, we can see that using *Entropy* or *Coverage*, regardless of their factor value, seem to yield the worse results in terms of running time. Furthermore, for larger datasets like `Obesity`, the execution could not finish within 24 hours when using either *Entropy* or *Coverage*. Second, it seems that there is no running time difference between algorithm con-

figurations that use either *Entropy* or *Coverage*. Configurations that use a minimum support of 0.5 yield the fastest execution times: this is indeed expected because the number of potential subgroups to explore gets lower when the minimum support value goes up. We find no notable running time differences between configurations of the locality factor for small datasets. However, with a larger dataset like `Obesity`, we can see that depending on the chosen minimum support, different values for the locality factor yield significant running time disparities.

Table 5.2: Running time comparison (in seconds) of quality measure parameters on 4 datasets using HV_{dev}. "-" means that the execution was not completed after 24 hours (86400 seconds).

Dataset \ Gen.		Minimum Support			Entropy			Coverage		
		0.1	0.3	0.5	0.1	0.5	1	0.1	0.5	1
`Fonseca`	$loc^{0.1}$	179	128	128	178	177	177	177	177	177
	$loc^{0.5}$	177	127	127	177	177	177	177	177	177
	loc^{1}	182	128	128	178	178	177	177	179	178
`Obesity`	$loc^{0.1}$	23189	11188	5213	-	-	-	-	-	-
	$loc^{0.5}$	16595	11021	6350	-	-	-	-	-	-
	loc^{1}	15426	13159	6443	-	-	-	-	-	-
`Plants`	$loc^{0.1}$	2.4	0.5	0.5	6.7	6.7	6.6	6.6	6.6	6.7
	$loc^{0.5}$	2.4	0.5	0.5	6.6	6.6	6.6	6.6	6.6	6.6
	loc^{1}	2.4	0.5	0.5	6.7	6.7	6.6	6.6	6.6	6.7
`RealEstate`	$loc^{0.1}$	41	6	3	108	109	109	113	113	112
	$loc^{0.5}$	41	6	3	108	109	109	113	113	112
	loc^{1}	41	6	3	100	107	109	108	113	113

Next, we want to investigate the impact of discretization on the quality of the discovered deviation models. To do this, we exploit the `Fonseca` dataset made of 5000 objects. We generate several datasets by using discretization techniques on the main dataset. We used equal-width and equal-frequency, two of the most well-known discretization techniques and for each technique, we tried respectively 2, 3, 5, 10, 15, and 20 bins. It leaves us with 12 datasets on which we can experiment with our method to study the effect of discretization on the quality of the discovered models. We run our algorithm with HV_{dev}, HD, AHD, and MHD and only retain the best subgroup found for each run. The results can be found in Table 5.3. The overall best model found for each distance measure is highlighted in red. Although the discretization technique seems to have a small impact on the quality of the best exceptional models (equal-width best model for HV_{dev}, and equal-frequency best model for HD, AHD and MHD), the main driver w.r.t. quality seems to be the number of bins. Indeed, in all cases, the quality of the best model peaks at 3 bins and then decreases as the number of bins increases. Finding large enough subgroups to create significant deviations in the Pareto front indeed becomes harder as the number of values that can be taken by each attribute grows.

Table 5.3: Impact of discretization on the quality of the best model for different measures.

Measure \ Disc. tech.	Equal-Width						Equal-Frequency					
	2	3	5	10	15	20	2	3	5	10	15	20
HV_{dev}	0.17	0.92	0.66	0.27	0.24	0.19	0.18	0.90	0.72	0.31	0.25	0.21
HD	0.31	0.43	0.31	0.29	0.25	0.25	0.31	0.45	0.33	0.29	0.25	0.25
AHD	0.28	0.31	0.21	0.21	0.19	0.19	0.28	0.32	0.25	0.23	0.19	0.19
MHD	0.298	0.297	0.20	0.21	0.19	0.19	0.29	0.30	0.28	0.24	0.19	0.19

5.5.3 Relevance of EPFAM

The goal here is to investigate the relevance of EPFAM on the same datasets as for Section 5.5.1 using HV_{approx}. For each dataset, we report the best approximation of the true Pareto front found according to the algorithm configuration. The best model found for each dataset is depicted in Figure 5.8. On `Fonseca`, we can see that the approximation found fits almost perfectly the true Pareto front and the subgroup is very small. Furthermore, the subgroup description which is $\langle x1 \in [-0.8, 0.8],\ x2 \in [-0.8, 0.8],\ x3 \in [-0.8, 0.8] \rangle$ supports the easy generation of very high quality solutions close to the true Pareto front.

Regarding `Obesity`, we also find a very good approximation of the true Pareto front, supported by the following description:

> $\langle$ Gender = 'Female', Frequency_consumption_vegetables = 3, Age $\in$ [13.953, 23.4], family_history_with_overweight = 'yes', Consumption_alcohol = 'Sometimes' $\rangle$.

This approximation corresponds to young women with a family history of obesity and alcohol consumption despite their young age. It is however interesting to note that these women have a high frequency of vegetable consumption.

When looking for the best approximation in `Plants`, we find a good approximation of the Pareto front, supported by the description $\langle family = `Combretaceae' \rangle$. Therefore, the family of plants known as $`Combretaceae'$ appears as representative of a high-quality trade-off between physical and chemical defense in plant seeds.

Finally, we study the best model found in `RealEstate`. We again find a very good approximation of the true Pareto front, supported by a small subgroup whose description is $\langle latitude \in [24.949, 24.965],\ number\_of\_convenience\_stores \in [4, 6] \rangle$.
Exploiting such a subgroup can allow for the easy discovery of houses that offer more interesting trade-offs between price and distance to the nearest transport station. It is interesting to note that both EPFDM and EPFAM can find the same exceptional model, but for different reasons. Indeed, it sometimes happens that the subgroup which creates the largest deviation of the true Pareto front is also the subgroup that best approximates it. Please note that this could be due the use of the same locality and generality parameters for both EPFDM and EPFAM. While this makes sense for fairness of comparison and working without a priori knowledge, using configurations of EPFAM where the generality of the model needs to be maximized could lead to much different results from those found with EPFDM.

Next, we want to investigate the impact of discretization on the quality of the discovered approximation models. To do this, we use the same 12 datasets used to study the impact of discretization on EPFDM in Section 5.5.2. We run our algorithm with HV_{approx} on each dataset and only retain the best subgroup found for each run. The results can be found in

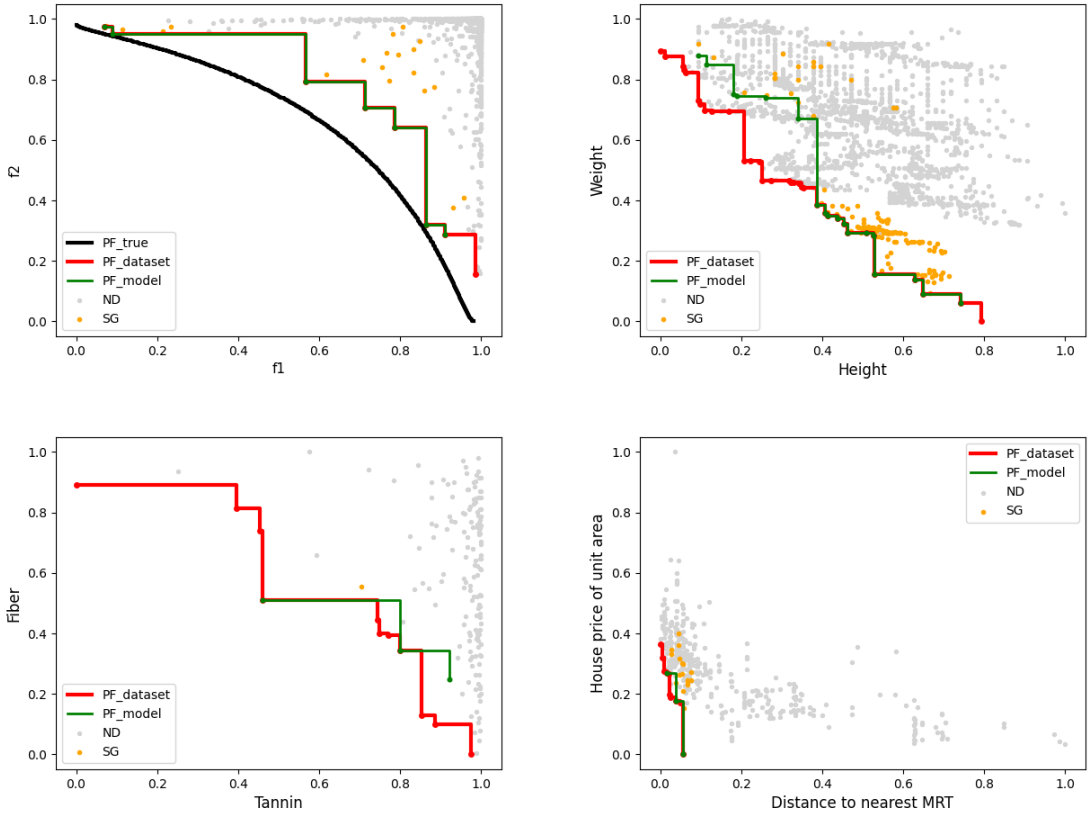

Figure 5.8: EPFAM best approximations with HV_{approx} on `Fonseca`, `Obesity`, `Plants` and `RealEstate` respectively.

Table 5.4: Impact of the discretization on the quality of the best model for EPFAM.

Disc. tech. / Measure	Equal-Width						Equal-Frequency					
	2	3	5	10	15	20	2	3	5	10	15	20
HV_{approx}	0.48	0.96	0.986	0.77	0.87	0.78	0.58	0.96	0.983	0.80	0.87	0.77

Table 5.4. The overall best model found is highlighted in red. Once again, the discretization technique seems to have little impact on the quality of the best exceptional models. Furthermore, the number of bins seems to be the main factor influencing the quality of the returned subgroups. Indeed, the quality of the best model peaks at 5 bins and then decreases for larger numbers of bins. While both EPFDM and EPFAM seem to be affected the same way by the used discretization techniques, their respective qualities peak at different numbers of bins.

5.5.4 Quantitative Comparison of EPFDM and EPFAM

Since different configurations of the algorithm have already been studied in Section 5.5.1 for several datasets, the same study is not needed for EPFAM. However, studying the running time of HV_{approx} on different datasets and comparing it to EPFAM using HV_{dev} is highly relevant. The results of these evaluations are available in Table 5.5. EPFAM is 2 to 7 times

faster than EPFDM on all datasets. It makes sense as the most expensive part of the process is the Pareto front computation of each subgroup, and the Pareto fronts in EPFAM are typically way easier to compute than in EPFDM. Indeed, in EPFDM, the Pareto front is computed on the complement of the dataset once the subgroup has been removed, while for EPFAM the Pareto front is computed on the subgroup itself. Since we generally favor small subgroups, the complement is way larger than the subgroup, hence the computation is faster for EPFDM than for EPFAM.

Table 5.5: Running time comparison (in seconds) of EPFDM and EPFAM.

Dataset	HV_{dev}	HV_{approx}
`Fonseca`	175	23.5
`Obesity`	15426	2320
`Plants`	2.4	0.9
`RealEstate`	40	7.7

5.5.5 A Use Case: Hyperparameter Optimization for Machine Learning.

Plant growth recipe optimization in urban farms is the application scenario that motivated this work, and the actionability of both EPFDM and EPFAM for this setting is studied on synthetic and real-life data in Chapter 6. We now propose to apply our algorithm when one needs to optimize multiple metrics at the same time for a machine learning task (e.g., precision and recall, bias and variance, quality and runtime, accuracy and interpretability). A metric can be any measure that needs to be optimized (e.g., a quality measure or the complexity of a model). Since multiple metrics need to be optimized at the same time, a trade-off has to be found. It has already been shown in the literature that one metric can often not be enough to assess the quality of a model (Caballero et al., 2010, Shi et al., 2012). We want to show how we can discover exceptional deviation and approximation models which can be exploited to both discover interesting nuggets of information and to generate new models that offer better trade-offs between the different objectives.

Multi-Label Classification.

We first consider the optimization of hyperparameters for a multi-label classification task using a random forest classifier. We use the popular `Yeast` dataset from the OpenML[4] repository. It is made of 2416 observations, 103 descriptive variables, and 14 binary labels to classify. We use 5 hyperparameters that are discretized into a list of values to sample from. For each run of the classifier, we select random values from the list of each hyperparameter. We run the classifier 200 times with different sets of hyperparameter values for each run and we assess the quality of the model by computing the recall and the precision of each model. Indeed, it is known that both precision and recall are important in classification tasks and that a trade-off between the two measures has to be found. Indeed, to do this, the F1 measure has been proposed. However, simplifying the problem of optimizing both measures into optimizing only one is a well-known trope in multi-objective optimization and it has

[4] `https://www.openml.org/`

been shown that it can lead to suboptimal results and loss of information. In other terms, we argue that optimizing both precision and recall at the same time is better than trying to optimize the F1 measure only. We finally build a dataset made of the 200 runs of the classifier with 5 descriptive variables – the hyperparameter values – and 2 objectives – the precision and recall – that need to be maximized. We then first use our algorithm to mine for the most exceptional deviation models in the data using HV_{dev} and HD, and we get the results reported in Figure 5.9.

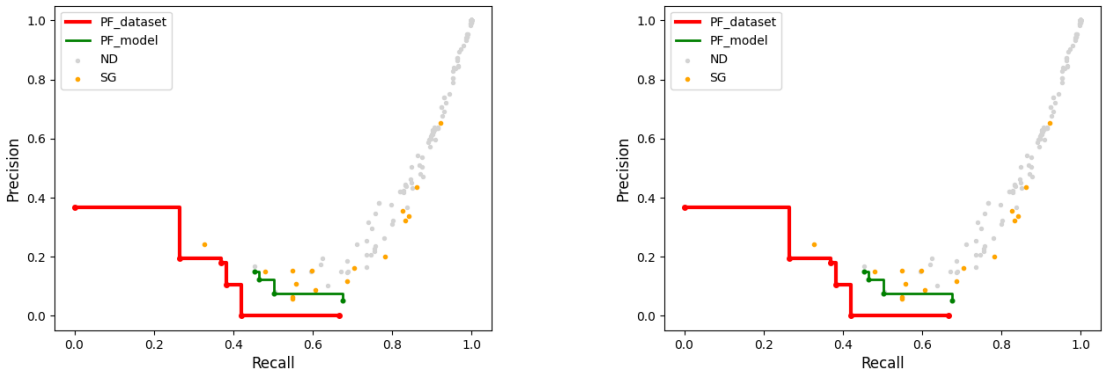

Figure 5.9: Best EPFDM models in `Yeast` with HV_{dev} (left) and HD (right).

The best discovered subgroup, described by $min\_samples\_leaf = 0.01$ is the same for both HV_{dev} and HD. Its removal creates a deviation of the whole true Pareto front. The objects of PF_true that belong to the subgroup not only have a common description, but also offer an overall excellent trade-off between recall and precision. Therefore, the description of the subgroup can be used not only to prune the hyperparameter search space for further optimization of the classifiers, but also to build good multi-label classifiers with an interesting trade-off between recall and precision.
Next, we want to exploit EPFAM to find a good approximation of the global Pareto front. We run our method and report the results in Figure 5.10. The subgroup description is clear and concise: $n\_estimators = 900.0$ *and* $min\_samples\_leaf = 0.01$. As can be seen in the figure, it seems like an excellent approximation of the true Pareto front, which is confirmed by its high quality of 0.96. Therefore, we can easily exploit its description to generate new models of higher quality.

So far, we have considered two objectives only. Our approach can however be generalized to more objectives. For instance, let us consider the same settings as with the previous example though adding the running time of each classifier as a third objective. Our goal is to look for models that maximize both precision and recall and at the same time minimize the running time. Since we have now a larger objective space, we increase the number of multi-label classifier executions to 400 to make sure that the dataset provides a good enough coverage of the objective space. After building our new dataset made of 400 observations, 5 descriptive variables, and 3 objectives to be optimized. We first run our EPFDM method with HV_{dev} and HD and return the best computed model for each measure.
We first report the results obtained with HV_{dev} in Figure 5.11. When dealing with Pareto fronts which are more than two-dimensional, one way to study their characteristics is to use

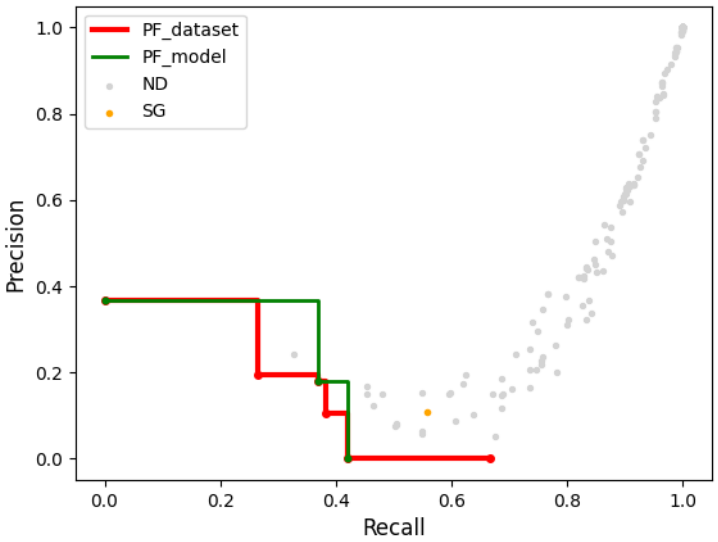

Figure 5.10: Best EPFAM model in `Yeast`.

scatter plots and visualize the pair-wise relationship of objectives. As it can be seen on each of the 3 scatter plots, the removal of the subgroup leads to a large deviation in all 3 pair-wise relationships that compose the overall Pareto front. The corresponding subgroup – which is relatively large – is described by $min\_samples\_split = 0.02$. From Figure 5.11, we can infer that the objects which compose the subgroup highly optimize the recall and correspond to overall good precision values, but show poorer results on execution time. This information as well as the subgroup description can be used to investigate the reasons why optimizing the recall leads to overall higher execution times, while the same relationship is not as clear between precision and execution time. Next, if concessions can be made on the degree of optimization of the execution time (i.e., we still want solutions on the Pareto front but other objectives can be prioritized when a conflict occurs), the subgroup can be exploited to further optimize the classifiers by looking for solutions which both optimize the recall and precision while keeping the execution time as low as possible.

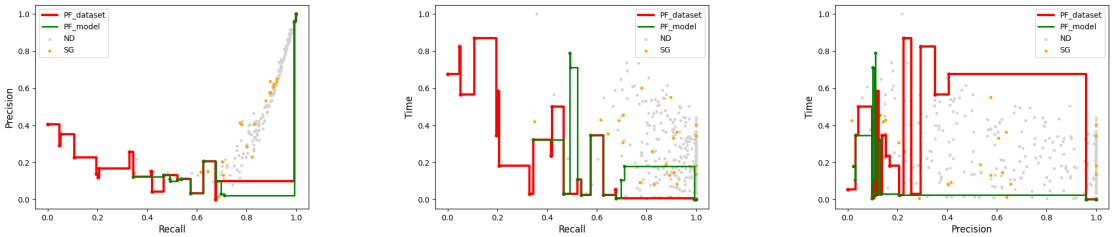

Figure 5.11: Scatter plots of the best computed EPFDM model with HV_{dev} showing the pair-wise relationship between objectives.

Next, we study the best subgroup returned with the HD measure. The 3 corresponding

scatter plots can be found in Figure 5.12. As can be seen in the plots, the HD measure offers a slightly different variation of the model found with HV_{dev}, composed of very few objects. In this case, the objects which compose the subgroup optimize even more the recall, at the cost of slightly worse trade-offs with both precision and time compared to the previous subgroup. The description of the subgroup, $min\_samples\_leaf = 0.01$ *and* $min\_samples\_split = 0.02$, confirms the relationship with the subgroup found with HV_{dev}, although it contains one different restriction from it. This subgroup could be exploited to generate models with highly optimized recall, decent precision, and relatively bad running times.

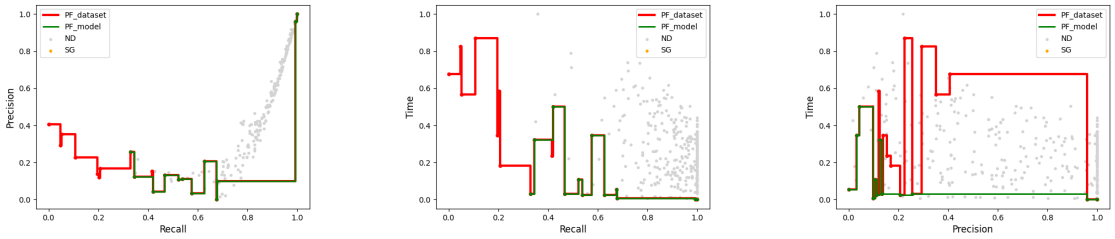

Figure 5.12: Scatter plots of the best computed EPFDM model with HD showing the pair-wise relationship between objectives.

Finally, we want to exploit EPFAM to find a model representing a very good approximation of the global Pareto front. We run our method and report the results in Figure 5.13. As can be seen in all 3 figures, we find a very good representative of the global model, even though we are now working on a problem with 3 objectives for which good approximations described by a subgroup would be expected to be hard to find – if they even exist. Furthermore, the subgroup is made of few points (less than 10% of the dataset), has a high quality (0.89), and possesses a clear description – $min\_samples\_split = 0.02$ – which can be exploited to generate new models with good overall trade-offs between the 3 objectives. It is interesting to note that we find the same subgroup that we found using EPFDM with HV_{dev}. In this particular case, the subgroup whose absence creates the largest deviation of the Pareto front is also the one that best approximates it.

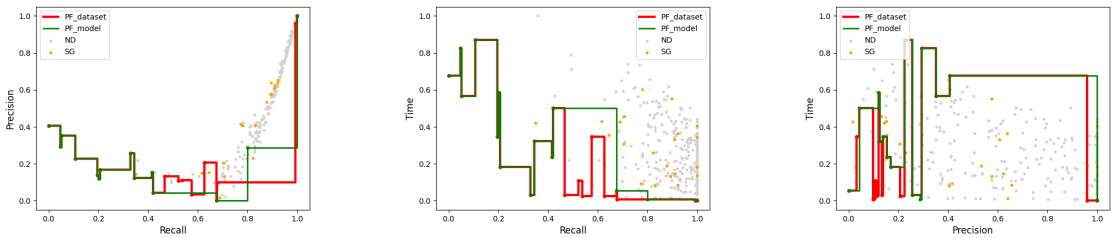

Figure 5.13: Scatter plots of the best computed EPFAM model showing the pair-wise relationship between objectives.

Regression.

Let us now consider the optimization of the hyperparameters of a neural network, and more precisely of a multi-layer perceptron regressor. We use the `California Housing` dataset from the scikit-learn library[5]. It is made of 20640 observations, 9 variables and the goal is to predict the sale price of each house. We retain 9 hyperparameters and discretize each of them into a list of values to sample from. For every run of the neural network, we sample random values from the list of each hyperparameter. To evaluate the quality of the neural network depending on the hyperparameter values, we run the model 200 times and, for each run, we compute the maximum residual error and the explained variance of the model. Finally, we build a dataset made of 200 observations with 9 descriptive variables – the hyperparameter values of each run – and 2 objective variables – the maximum residual error and the explained variance – which both need to be minimized.
We first want to use EPFDM to discover exceptional deviation models. To do this, we run our method with the HV_{dev} measure and return the top-3 models found. The results are reported in Figure 5.14. The best subgroup seems to create a large deviation of most of the true Pareto front. Its description is $learning\_rate\_init = 0.01$ and applying this restriction seems to lead to good overall trade-offs between explained variance and maximum residual error for the neural network (i.e., removing those objects leaves neural networks with poorer trade-offs). This is interesting: we now know that running our neural network with a learning rate of 0.01 will lead to good overall quality models. It can also be used as a basis for further hyperparameter optimization. The second best subgroup found creates a deviation in the left part of the Pareto front. This part of the Pareto front corresponds to very good values of explained variance and less optimized values of residual error. Its description is different from that of the previous subgroup: $learning\_rate\_init = 0.01$ *and* $max\_iter = 100$. This is also an interesting subgroup, it is made of very few objects (<5% of the dataset) and it represents a very specific part of the Pareto front. It can be exploited to generate models with good explained variance (at the cost of a worse residual error), or it can be used to prune these same solutions from the search space. Furthermore, we can observe that there are very few points around that part of the Pareto front in the objective space. It could be interpreted as a problem of missing data. Indeed, since most models generated seem to optimize the residual error at the cost of the explained variance, we have access to few models with optimized variance, which leads us to find this subgroup. Therefore, the user can see this as an indication that more data is needed in that part of the objective space to make informed decisions. Finally, as can be seen in the figure, the third best subgroup produces the same deviation of the true Pareto front than the second subgroup, although it is made of many more objects (≈20% of the dataset). The subgroup is therefore less interesting since creating an equivalent deviation with a much larger subset of the dataset is easier.

So far, we have shown that EPFDM can be used as an exploratory data analysis tool to find interesting parts of the true Pareto front. Next, we want to show how our second method, EPFAM, can be exploited to iteratively optimize the trade-offs of the generated models.
To achieve this, we first run our method on the original `California Housing` dataset and retrieve the best approximation of the overall Pareto front. The results are illustrated in Figure 5.15 (left). As can be seen, we find a subgroup whose Pareto front fits almost perfectly the true Pareto front, although it is relatively large (≈20% of the dataset). It is

[5] `https://bit.ly/38UGJe9`

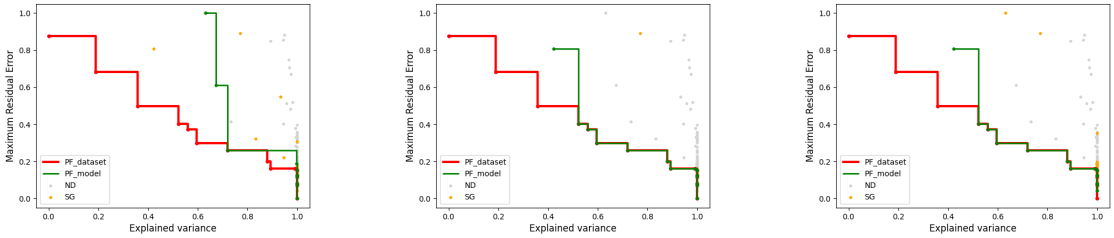

Figure 5.14: Top-3 best EPFDM models in `California Housing` with HV_{dev}.

described by *learning_rate_init* = 0.01. This is clear and concise and it can be exploited to produce new, better models. To do that, we generate 200 new neural network models with random hyperparameter values, with the important difference that for these models, the domain of values of the learning rate hyperparameter is restricted to 0.01, i.e., following the description of the subgroup. By doing this, we increase our chances of generating models with good trade-offs between maximum residual error and explained variance. We then build a new dataset out of these 200 models. The new models can be observed in Figure 5.15 (right), where we can clearly see that the Pareto front of the new data – PF_it_2 – is better than that of the original data, i.e., PF_it_1. This is confirmed by the numbers available in Table 5.6. The new dataset of models optimizes significantly better the explained variance, at a small cost for the residual error. The hypervolume of the new set of models is also significantly better than that of the original models (0.61 vs 0.49). Please note that for a fair comparison, the hypervolume of each dataset has to be recalculated after each iteration, since the reference point (built out of the worst values found for each objective out of all the objects encountered) can change at each generation of a new dataset.

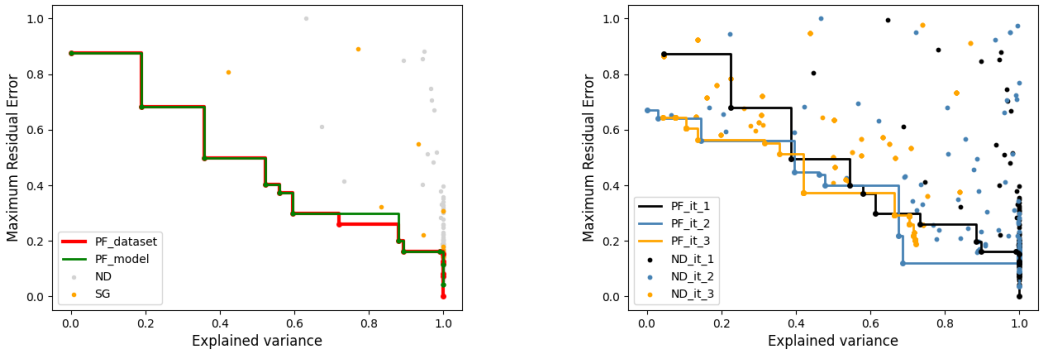

Figure 5.15: Best EPFAM model in `California Housing` (left) and comparison of the datasets generated for the 3 iterations of our process exploiting (right).

Next, we run EPFAM on the set of models of the second iteration to find the best approximation of the new Pareto front. We find a small subgroup (<4% of the dataset) whose description is *alpha* = 0.0005 *and epsilon* = 0.000001. Once again, we can exploit the description of the subgroup to apply restrictions on the *alpha* and *epsilon* hyperparameters before generating new, better models. We generate 200 new models – using all the restrictions

encountered until now – and build a dataset out of them. The new dataset can be observed in Figure 5.15 (right), where we can see that the Pareto front of the new data – PF_it_3 – is as good if now better than that of the second iteration, i.e., PF_it_2. This is confirmed by the numbers available in Table 5.6. The dataset of this third iteration is composed of models with a much better average variance, at some cost on the average residual error. The hypervolume, however, stays the same, signaling that it might not be possible to optimize the trade-offs between quality measures further in this case. In this scenario, we showed that EPFAM can be used as a relevant and actionable tool to iteratively improve machine learning models through hyperparameter optimization.

Table 5.6: Comparison of the average, median, standard deviation values of both the explained variance and the maximum residual error, and comparison of the hypervolume between the original models and the new sets of models generated during the different iterations of our process.

	$Variance_{avg}$	$Error_{avg}$	$Variance_{med}$	$Error_{med}$	$Variance_{std}$	$Error_{std}$	$Hypervolume$
Original models	0.97	0.22	1	0.16	0.12	0.17	0.49
Iteration 2	0.91	0.25	1	0.16	0.20	0.20	0.61
Iteration 3	0.83	0.28	0.98	0.17	0.26	0.22	0.61

5.6 Conclusion

To investigate the cross-fertilization between Exceptional Model Mining and Multi-objective Optimization, we built a new model class called Exceptional Pareto Front Mining. While other approaches that link pattern mining to MOO work at the pattern level, EPFM is able to find relevant patterns at the object level. Our first approach EPFDM looks for deviations in the shape of the Pareto front created by the absence of a subgroup of objects, compared to the same Pareto front computed on the whole dataset. Our second approach EPFAM looks for subgroups whose Pareto front approximates exceptionally well the true Pareto front.
Thanks to experiments on both synthetic and real-life data, we discussed the relevance of our methods on different use cases. On typical multi-objective optimization scenarios, EPFDM can be used as an exploratory analysis tool to identify key features in the description space leading to better or worse trade-offs between objectives. EPFAM can be exploited to find exceptionally good approximations of the true Pareto front. In other terms, EPFAM enables the generation of Pareto optimal solutions with a higher probability.
In scenarios with limited data and unknown underlying models, the methods can be used (i) to identify a subspace of the current Pareto front where data might be missing, (ii) to select a subset of better or worse solutions of the Pareto front with an explicit and concise description in the attribute description space, (iii) to identify anomalous parts of the Pareto front, (iv) and to find exceptionally good approximations of the Pareto front, that can be exploited to generate higher quality solutions.
Furthermore, the relevance of the integration of EPFAM in an iterative optimization framework has been validated on a use case for hyperparameter optimization in Machine Learning. It is worth noting that this method has also been exploited for plant recipe optimization in controlled environments (see Chapter 6). The integration of our method in a proper EMM-

based iterative optimization process seems like a logical next step to fully exploit its potential in multi-objective optimization.

Chapter 6

Plant Growth Recipe Optimization in Controlled Environments

In this chapter, we investigate the actionability of our contributions to SD and EMM for plant growth recipe optimization in controlled environments, the real-life setting that has motivated our research. We first introduce important concepts, such as urban farms and plant growth recipes. We detail how an existing crop growth simulator can be exploited to generate synthetic recipes that replicate a controlled environment and support the empirical validation of our work. The relevance of our methods to improve plant growth in both single and multi-objective optimization settings is validated thanks to these synthetic yet realistic recipes. Finally, thanks to temporary access to an urban farm, we detail preliminary results of the application of our methods to basil growth optimization.

6.1 Introduction

Nowadays, conventional farming methods have to face many tough challenges like, for instance, deforestation, soil erosion, and groundwater depletion. Furthermore, crucial problems related to the climate crisis also stimulate the need for new production systems. The concept of vertical urban farms (see, e.g., AeroFarms, Infarm, Bowery Farming[1]) can be part of a solution. Urban farming enables the growth of plants in fully controlled environments close to the place where consumers are (Harper and Siller, 2015). These farms allow for the removal of pesticides and a significant reduction in water consumption, while being able to optimize both the quantity and quality of plants (e.g., improving the flavor (Johnson et al., 2019) or their chemical proportions (Wojciechowska et al., 2015)). For most new technologies and innovations, the barrier for entry into the mainstream lies in proving their cost-effectiveness compared to existing methods. Vertical urban farms are no stranger to this problem, with most of their critics arguing that operating and infrastructure costs will always keep them from being profitable at large scale. In this context, the development of computer-based methods allowing the optimization of urban farming processes is a big step toward urban farms becoming successful. Figure 6.1 depicts an example of a real urban farm where plants grow in vertically stacked layers.

Figure 6.1: Inside image of the FUL vertical urban farm

Urban farms can generate large amounts of data that can be pushed towards a cloud environment such that various machine learning and data mining methods can be used. We may then provide new insights about the plant growth process itself (discovering knowledge about not yet identified/understood phenomena) but also offer new services to farm owners. The number of parameters influencing plant growth can be relatively important (e.g., temperature, hygrometry, water pH level, nutrient concentration, LED lighting intensity, CO_2 concentration). In urban farm environments, these parameters can all be controlled from the moment the plants are planted up to the day of harvesting. Not only experts can specify

[1] `https://aerofarms.com/`, `https://infarm.com/`,`https://boweryfarming.com/`.

a priori the expected values for these descriptive attributes but also sensors can collect the true values during the whole plant growth process. There are numerous ways of measuring the relevance of the crop end-product (e.g., cost, yield, size, flavor, or chemical properties). In other terms, we can retrieve several targets that can be used to evaluate the value of a given crop, though we consider only numerical ones for the moment.
In general, for a given type of plants, expert knowledge exists that concerns the available sub-systems (e.g., to model the impact of nutrient on growth, the effect of LED lighting on photosynthesis, the energy consumption w.r.t. the temperature instruction) but we are far from a global understanding of the interaction between the various underlying phenomena. In other terms, setting the optimal instructions for the diverse set of parameters given an optimization task remains an open problem.
We want to address such an issue by means of data mining techniques. Can we learn from available recipe records to suggest new ones that should provide better results w.r.t. the selected target attributes? It is worth noting that in the context of plant growth optimization in urban farms, the underlying model is unknown, preventing the use of traditional genetic or evolutionary methods. Moreover, experiments have to be executed in batch and in very limited occurrences due to time and cost constraints.
Please note that for reproducibility purposes, all datasets and source code are made available in `https://bit.ly/3ilhir5`.
The remaining of this chapter is organized as follows. In Section 6.2, we explain the concept of plant growth recipe. We then introduce our synthetic recipe generator in Section 6.3. A generic framework for actionable SD is introduced and applied to recipe optimization in Section 6.4. In Section 6.5, multi-objective plant growth recipe optimization is studied on synthetic data. Then, a real-life application to plant growth optimization is considered in Section 6.6. Finally, Section 6.7 concludes.

6.2 Plant Growth Recipes

In urban farms, plants grow in an environment where the growth conditions can be controlled and collected throughout their development. From agronomic knowledge, we also know that plants go through different stages during their growth, from the time they are planted to their harvesting. Figure 6.2 illustrates the growth process of plants.

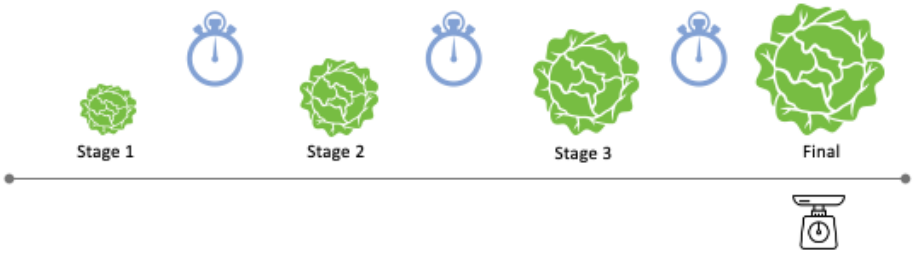

Figure 6.2: Growth process

Given the different growth stages of a plant, the concept of growth recipe consists in the aggregation of the growing conditions set at each stage, and it can be evaluated by one

or several numerical objectives. A growing condition can be defined as any variable which affects the way plants grow. We now formalize the notion of growth recipes.

Definition 34. *A plant growth recipe (M, P, T) is given by a set of numerical parameters M specifying the growing conditions thanks to intervals on numerical values, a numerical value P representing the number of stages of the growth cycle, and a numerical target label T to quantify the recipe quality. In a given recipe, each parameter of M is repeated P times s.t. we have $|M| \times P$ numerical attributes.*

Examples of such recipes can be found in Table 6.1.

Table 6.1: Examples of generic plant growth recipes.

Light^{P1}	Temperature^{P1}	Light^{P2}	Temperature^{P2}	Light^{P3}	Temperature^{P3}	T
50	25	50	21	20	23	0.3
70	18	80	24	30	12	0.45
60	30	70	18	80	17	0.95
30	32	60	26	50	31	0.75
90	27	30	16	70	18	0.6

Our goal is to discover the characteristics (i.e., the growing conditions) of an optimized growth. In expert hands, such characteristics can be exploited to define better recipes. The design of recipes can be done according to (i) statistical and mathematical methods (ii) empirical studies (iii) expert knowledge (iv) a combination of all of those. Thanks to expert knowledge and the state of the art, we can design recipes that take into account already studied complex interactions between variables, such as the connection between air circulation and humidity. Furthermore, statistical methods support the creation of experimental recipes that cover as much as possible the search space, which can help in the discovery of previously unknown information, and also helps in avoiding the creation of multiple redundant recipes.

6.3 A Synthetic Data Generator

This research is partially funded by a project on urban farm recipe optimization. Even though we do not have access to real farming data yet, we found a way to support an empirical study on multi-objective recipe optimization using EMM thanks to inexpensive experiments enabled by the PCSE[2] simulator. Originally, the PCSE environment was designed to build crop simulation models for conventional farming (i.e., crops growing outside in fields, in non-controlled environments). The PCSE process is depicted in Figure 6.3.

To simulate the growth of a given crop, the model needs several files as input. First, the *soil file* which contains information on the physical properties of the soil, such as water retention, hydraulic conductivity, and soil workability. Second, it needs a *crop file* that defines which crop is simulated: it describes the crop growth process according to numerous parameters. Notice that we selected the sugar beet as reference crop for all our experiments hereafter. Next, it needs an *agromanagement file*, that contains the schedule for agromanagement processes such as irrigation, weeding, nutrient application, and pest control. Finally and most importantly for us, it requires a *weather file* that provides daily values for several

[2] `https://pcse.readthedocs.io/en/stable/index.html`

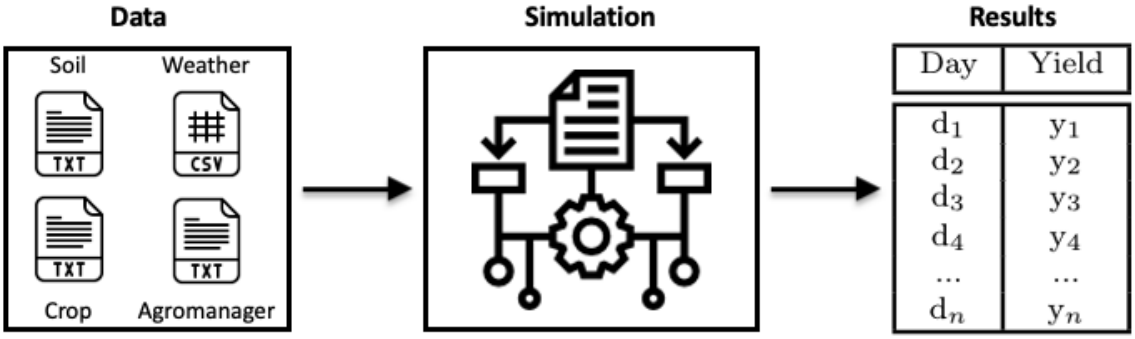

Figure 6.3: The PCSE simulation process

weather variables. This file defines important growing conditions of the plants day by day. Normally, it would contain real weather data extracted from one of the sources supported by PCSE. Since we have full control on the weather file that is used as input of the simulator, we can set our own values for each variable and each day, making it possible to simulate plant growth in controlled conditions. The complete list of variables that can be used to control the environment can be found in Table 6.2.

Table 6.2: Weather file variable description.

Name	Description	Unit
RAIN	Precipitation (rainfall or water equivalent in case of snow or hail)	$cmday^{-1}$
IRRAD	Daily global radiation	$Jm^{-2}day^{-1}$
WIND	Mean daily wind speed at 2 m above ground level	$msec^{-1}$
VAP	Mean daily vapour pressure	hPa
TMIN	Daily minimum temperature	$°C$
TMAX	Daily maximum temperature	$°C$

We can control some of the most important variables that drive the plant growth process. We need to be able to set values for each variable and each day of the plant development. After in-depth investigation of each variable according to the PCSE documentation and taking into account basic agronomic knowledge, we choose values for each variable as follows: IRRAD takes values in [10000,30000], RAIN takes values in [5,30], WIND is taken in [0,20], VAP takes values in [1.1,1.6], TMIN is taken in [15,22] and finally, TMAX takes values in [23,30]. Please note that we only use these domains of values for application cases where the 6 variables can be exploited at the same time. For example, when using exhaustive methods like `OSMIND` where the number of attributes has to be limited, we restrict the number of variables to 3 and the previously defined domain of values are not relevant anymore. It is also worth noting that when sampling new values for each of the variables that will compose a given recipe, we do not carry out a completely random sampling in the domains of values, but instead we draw random values from a list of 15 evenly spaced numbers over each domain of values. For example, given variable IRRAD with value domain [10000,30000], we split the domain into 15 evenly spaced numbers and randomly draw values from these numbers. This way, we avoid having to use any kind of discretization in some important use cases where the search space would be too much to handle if we did not use this technique, e.g., when using

OSMIND. Table 6.3 depicts an example of recipe with as many growing stages as there are days between the planting of the crop and its harvest.

Table 6.3: Example of Weather file: growing conditions for a given plant day by day.

Day	RAIN	IRRAD	WIND	VAP	TMIN	TMAX
d_1	10	23250	15	1.2	15	27
d_2	12	18250	12	1.4	16.5	23.4
d_3	14	24560	7	1.35	17.8	21.5
...	...	...	...	...	...	...
d_n	8	14950	22	1.1	21.1	29.9

Having that many growing stages does not make sense since (i) the growth process for most plants takes weeks to months, (ii) we know from expert knowledge that the growth process of many plants can be split into 3 stages only. For this reason, we have been splitting the Weather file to consider 3 stages using the following method: (1) we define the number of days of the growth process until harvesting, (2) we divide this number by 3 (i.e., 3 stages) which defines the length of a stage, (3) for each stage, we define a unique value for each variable, that will be repeated for as many days as needed in the weather file. An example of the end result of this process for a crop whose growth process takes 300 days is available in Table 6.4.

Table 6.4: Example of Weather file: growing conditions for a given plant stage by stage.

Stage	RAIN	IRRAD	WIND	VAP	TMIN	TMAX
P_1 (d_1 - d_{100})	11	13250	19	1.39	18	26
P_2 (d_{101} - d_{200})	14	15976	9	1.26	15	21
P_3 (d_{201} - d_{300})	24	28390	18	1.42	19	29

The PCSE process outputs as result the state of the plant day by day from the time of planting (d_1,y_1), up to its harvest (d_n,y_n) – see Figure 6.3. However, while the simulator provides us with the yield, it was not built to output the cost of a given crop. We decide to consider the cost as being an energy cost for each recipe. Thanks to expertise from our partner FUL, we have studied the detailed energy consumption of their pilot urban farm for each environment variable. From this data, we were able to find the percentage of total energy consumption of each environment variable. This is however a piece of confidential information that cannot be detailed here. It was then possible to define an approximate percentage of total energy consumption for each variable of the PCSE model. The results are as follows: RAIN represents 24.61% of the energy cost of a recipe, IRRAD 49.22%, WIND 5.15%, VAP 10.74%, TMIN 5.14%, and TMAX 5.14% too. The cost of a recipe is then computed the following way: (i) we normalize variables so that their values fall between 0 and 1, (ii) each variable of the recipe is multiplied by its share of the total energy consumption, (iii) we add the values obtained for each variable and divide the total by the number of stages, such that the final cost of the recipe falls between 0 and 1.

6.4 Subgroup Discovery for Urban Farm Optimization

6.4.1 Context of Recipe Optimization

Designing, selling, and/or exploiting connected vertical urban farms is now receiving a lot of attention. Looking for innovative ideas to optimize recipes, we investigate the use of an optimal subgroup discovery method from purely numerical data. It concerns here the computation of subsets of recipes whose labels (e.g., the yield) show an interesting distribution according to a quality measure. When considering optimization, e.g., maximizing the yield, our virtuous circle optimization framework iteratively improves recipes by sampling the discovered optimal subgroup description subspace. We provide our preliminary results about the added value of this framework thanks to the plant growth simulator introduced in Section 6.3 that enables inexpensive experiments.

The material of this section has been published in the Proceedings of the 2020 International Symposium on Intelligent Data Analysis (IDA) (Millot et al., 2020c). For reproducibility purposes, all datasets and source code are made available in `https://bit.ly/3ilhir5`.

Our goal is to optimize recipes and we want to discover actionable patterns in the sense that delivering such patterns will support the design of new growing conditions and thus recipes. An optimization measure f quantifies the quality of an iteration. We are interested in the mean of the target label of the objects of the optimal subgroup after each iteration. The measure is given by $f_{mean} = \frac{\sum_{i \in ext(p)} T(i)}{|ext(p)|}$ where $T(i)$ is the value of the target label for object i.

Designing recipes that optimize a given target attribute (e.g., the mass, the energy cost) is often tackled by domain experts who exploit the scientific literature. However, in our setting, it has two major drawbacks. First, most of the literature remains oriented towards conventional growing conditions and farming methods. In urban farms, more parameters can be controlled. Secondly, the amount of knowledge about plants is unbalanced from one plant to another. Therefore, relying only on expert knowledge for plant recipe optimization is not sufficient. We have an optimization problem and the need for a limited number of iterations. Indeed, experimenting with plant growth recipes is time-consuming (i.e., asking for weeks or months). Therefore, we have to minimize the number of experiments that are needed to optimize a given recipe.

There are two main families of methods addressing the problem of optimizing a function over numerical variables: *direct* and *model-based* (Rios and Sahinidis, 2013). For *direct* methods, the common idea is to apply various strategies to sequentially evaluate solutions in the search space of recipes. However such methods do not address the problem of minimizing the number of experiments. For *model-based* methods, the idea is to build a model simulating the ground truth using available data and then to use it to guide the search process. For instance, (Johnson et al., 2019) introduced a solution for recipe optimization using this type of method with the goal of optimizing the flavor of plants. Their framework is based on using a surrogate model, in this case, a Symbolic Regression (Koza, 1992). It considers recipe optimization by means of a promising virtuous circle. However, it suffers from several shortcomings: there is no guarantee on the quality of the generated models (i.e., they may not be able to model correctly the ground truth), the number of tested parameters is small (only 3), and the ratio between the number of objects and the number of parameters in the data needs to be at least ten for Symbolic Regression (Jones et al., 1998). Clearly, it would restrict the search to only

a few parameters.

6.4.2 Leveraging Subgroups to Optimize Recipes

A virtuous circle. Our optimization framework can be seen as a virtuous circle, where each new iteration uses information previously gathered to iteratively improve the targeted process. First, a set of recipe experiments – which can be created with or without the use of expert knowledge – is created. With the use of expert knowledge, values or domain of values are defined for each attribute and then recipes are produced using these values. When generating recipes without prior knowledge, we create recipes by randomly sampling the values of each attribute. Secondly, we use subgroup discovery to find the best subgroup of recipes according to the chosen quality measure (e.g., the subgroup of recipes with the best average yield). Then, we exploit the subgroup description – i.e., we apply new restrictions on the range of each parameter according to the description – to generate new, better, recipe experiments. Finally, these recipes are in turn processed to find the best subgroup for the new recipes, and so on until recipes cannot be improved anymore. This way, we sample recipes in a space that gets smaller after each iteration and where the ratio between good and bad solutions gets larger and larger. Fig. 6.4 depicts a step-by-step example of the process behind the framework. Our framework makes use of several hyperparameters that affect runtime efficiency, the number of iterations, and the quality of the results.

Convergence. The first hyperparameter is the parameter a used in the q^a_{mean} quality measure. In standard subgroup discovery, it controls the number of objects in the returned subgroups. A higher value of a means larger subgroups. For us, a larger subgroup means a larger search space to sample. By extension, a higher value of a means more iterations to be able to reach smaller subspaces of the search space. For that reason, we rename the parameter as the *convergence rate*. The second hyperparameter is called the *minimal improvement* ($minImp$). It defines the minimal improvement of the *Optimization measure* – f_{mean} in our setting – needed from one iteration to another for the framework to keep running. After each iteration, we check whether the following statement is true or false.

$$\frac{f_{mean_{it}} - f_{mean_{it-1}}}{f_{mean_{it-1}}} \geqslant minImp$$

If it is true, then the optimization framework keeps running, else we consider that the recipes cannot be improved any further. This parameter has a direct effect on the number of iterations needed for the algorithm to converge. A higher value for $minImp$ means a lower number of iterations and vice versa. We can also forget $minImp$ and set the number of iterations by means of another parameter that would denote a budget.

Sampling the subspace. After each iteration, to generate new recipes to experiment with, we need to sample the subspace corresponding to the description of the best subgroup. Three sampling methods are currently available and this defines again a new hyperparameter. The first method consists in sampling recipes using the original set of values of each attribute (i.e., in the first iteration) minus the excluded values due to the new restrictions applied on the subspace. Let D^1_m be the domain of values of attribute m at Iteration 1 and $[a^i_m, b^i_m]$ be

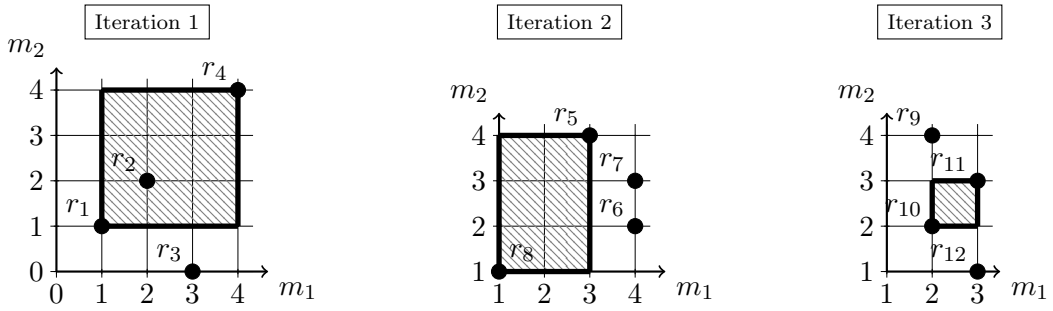

Figure 6.4: Example of execution of the optimization framework in 3 iterations. We consider a two-dimensional space (i.e., 2 attributes m_1 and m_2) where 4 recipes are generated during each iteration using our first sampling method. The best subgroup (optimizing the yield) of each iteration (hatched) serves as the next iteration sampling space.

the interval of attribute m at Iteration i according to the description of the best subgroup of Iteration $i-1$. Then, $\forall v \in D_m^1, v \in D_m^i \Leftrightarrow b_m^i \geqslant v \geqslant a_m^i$. Using this method, the number of values available for sampling for each attribute gets smaller after each iteration, meaning that each iteration is faster than the previous one. The second consists in discretizing the search space through the discretization of each attribute in k intervals of equal length. Parameter k is set before launching the framework. Recipes are then sampled using the discretized domain of values for each attribute. Finally, we can use *Latin Hypercube Sampling* (McKay et al., 1979) as a third method. In *Latin Hypercube Sampling*, each attribute is divided into S equally probable intervals, with S the number of samples (i.e., recipes). Using this method, recipes are sampled such that each recipe is the only one in each hyperspace that contains it. The number of samples generated for each iteration is also a hyperparameter of the framework.

An explainable generic framework. Our optimization framework is explainable contrary to black-box optimization algorithms. Each step of the process is easily understandable due to the descriptive nature of subgroup discovery. Although we have been referring to our algorithm `OSMIND` when introducing the optimization framework, other subgroup discovery algorithms can be used, including (Lemmerich et al., 2016a). Notice however that the better the quality of the provided subgroup, the better the results returned by our framework will be. Finally, our method can be applied to quite many application domains where we want to optimize a numerical target given collections of numerical features (e.g., hyperparameter optimization in machine learning). The pseudocode of the framework is available in Algorithm 4.

6.4.3 Experiments

The FUL partner in the FUI DUF 4.0 project (2018-2021) is designing new types of urban farms. It was however impossible to experiment with their prototype before late 2020. Therefore, we found a way to support the empirical study of our recipe optimizing framework thanks to inexpensive experiments enabled by a simulator. In an urban farm, plants grow in a controlled environment. In the absence of failure, recipe instructions are followed and

Algorithm 4 Optimization framework

Input: Subgroup quality measure q, minimal improvement $minImp$, framework quality measure fq, expert knowledge ek, sampling method sm, convergence rate cr
Output: List optimal_objects

```
1: objects_list ← generateInitialObjects(ek, sm)
2: constraints ← computeConstraints(objects_list)
3: current_improvement ← minImp
4: avg_objects ← computeAvgTarget(objects_list)
5: optimal_objects ← new list()
6: while (current_improvement ⩾ minImp) do
7:     current_optimal_sg ← computeOptimalSG(objects_list, q, cr)
8:     current_improvement ← fq(current_optimal_sg, avg_objects)
9:     if (current_improvement > 0) then
10:        optimal_objects ← getObjects(current_optimal_sg)
11:    end if
12:    if (current_improvement ⩾ minImp) then
13:        avg_objects ← computeAvgTarget(current_optimal_sg)
14:        constraints ← updateConstraints(current_optimal_sg)
15:        objects_list ← generateNewObjects(constraints, sm)
16:    end if
17: end while
18: return optimal_objects
```

we can investigate the optimization of the plant yield at the end of the growth cycle. We simulate recipe experiments thanks to PCSE by setting the characteristics (e.g., the climate) of the different growing stages. Let us here focus on 3 variables that set the amount of solar irradiation Irrad (range [0,25000]), Wind (range [0,30]) and Rain (range [0,40]). The plant growth is split into 3 stages of equal length such that we finally get 9 attributes. In real life, we can control most of the parameters of an urban farm (e.g., providing more or less light) and a recipe optimization iteration needs for new insights about the promising parameter values. This is what we can emulate using the crop simulator: given the description of the optimal subgroup, we get insights to support the design of the next simulations, say experiments, as if we were controlling the growth environment. At the end of the growth cycle, we retrieve the total mass of plants harvested using a given recipe. Note that in the following experiments, unless stated otherwise, no assumption is made on the values of parameters (i.e., no restriction is applied on the range of values defined above and expert knowledge is not taken into account). Table 6.5 features examples of plant growth recipes. The source code and datasets used in our evaluation are available at `https://bit.ly/3ilhir5`.

`OSMIND` vs `SD-MAP*`.

We study the description of the best subgroup returned by `OSMIND` and `SD-MAP*`, the state-of-the art algorithm for subgroup discovery in numerical data. Table 6.6 depicts the descriptions for a dataset comprised of 30 recipes generated randomly with the simulator. Besides the higher quality of the subgroup returned by `OSMIND`, the optimal subgroup description also

Table 6.5: Examples of growth recipes split in 3 stages (P1, P2, P3), 3 attributes, and a target label (Yield).

R	Rain^{P1}	Irrad^{P1}	Wind^{P1}	Rain^{P2}	Irrad^{P2}	Wind^{P2}	Rain^{P3}	Irrad^{P3}	Wind^{P3}	Yield
r_1	10	23250	5	10	23250	5	15	21000	10	22000
r_2	35	10000	14	5	25000	10	16	19500	30	20500
r_3	15	17500	26	22	15000	18	30	4000	3	8600
r_4	18	22800	17	38	17000	25	38	12000	19	14200

Table 6.6: Comparison between descriptions of the overall dataset (DS), the optimal subgroup returned by `OSMIND`, the optimal subgroup returned by `SD-MAP*`. "-" means no restriction on the attribute compared to DS, Q and S respectively the quality and size of the subgroup.

Subgroup	Rain^{P1}	Irrad^{P1}	Wind^{P1}	Rain^{P2}	Irrad^{P2}	Wind^{P2}	Rain^{P3}	Irrad^{P3}	Wind^{P3}	Q	S
DS	[0,39]	[1170,23471]	[2,29]	[0,37]	[111,24111]	[0,29]	[2,40]	[964,24197]	[1,30]	0	30
`OSMIND`	[16,37]	[1170,22085]	[2,24]	[7,37]	[18309,23584]	[2,24]	[15,37]	[12626,24197]	[1,25]	33874	7
`SD-MAP*`	[21,39]	-	-	-	[14455,24111]	-	-	[12760,24197]	-	30662	5

enables to extract information that is missing from the description obtained with `SD-MAP*`. In fact, where `SD-MAP*` only offers a strong restriction on 3 attributes, `OSMIND` provides actionable information on all the considered attributes, i.e., the 9 attributes. It confirms its qualitative superiority over `SD-MAP*` which has to proceed to attribute discretizations.

Empirical Evaluation of the Model Hyperparameters.

Our optimization framework involves several hyperparameters whose values need to be studied to define proper ranges or values that will lead to optimized results with a minimized number of recipe experiments. We choose to apply a random search on discretized hyperparameters. Note that in this setting, grid search is a bad solution due to the combinatorial number of hyperparameter values and the high time cost of the optimization process itself. We discretize each hyperparameter in several values (the convergence rate is split into 10 values ranging from 0.1 to 1, the minimal improvement parameter is split into 12 values between 0 and 0.05, the sampling parameter is split between the 3 available methods, and the number of recipes for each iteration is either 20 or 30). We run 100 iterations of random search, with each iteration – read set of parameter values – being tested 10 times and averaged to account for the randomness of the recipes generated. After each iteration of random search, we store the set of hyperparameter values and the corresponding best recipe found. Fig. 6.5 depicts the results of the experiments. Optimal values for convergence rate seem to be around 0.5, between 0.001 and 0.01 for minimal improvement, and the best sampling method is tied between the first and second one. Generating 30 recipes for each iteration yields better results than 20 (average yield of 23857 for 30 recipes against 22829 for 20 recipes). To compare our method against other methods, we run our framework with the following parameters: 30 recipes times 5 iterations (for a total of 150 recipes), 0.5 convergence rate, using the second sampling method with $k = 15$. To address the variance in the yield due to randomness in the recipe generation process, we run the framework 10 times, store the best recipe found at each iteration and then compute the average of the stored recipes. We report the results in Table 6.7.

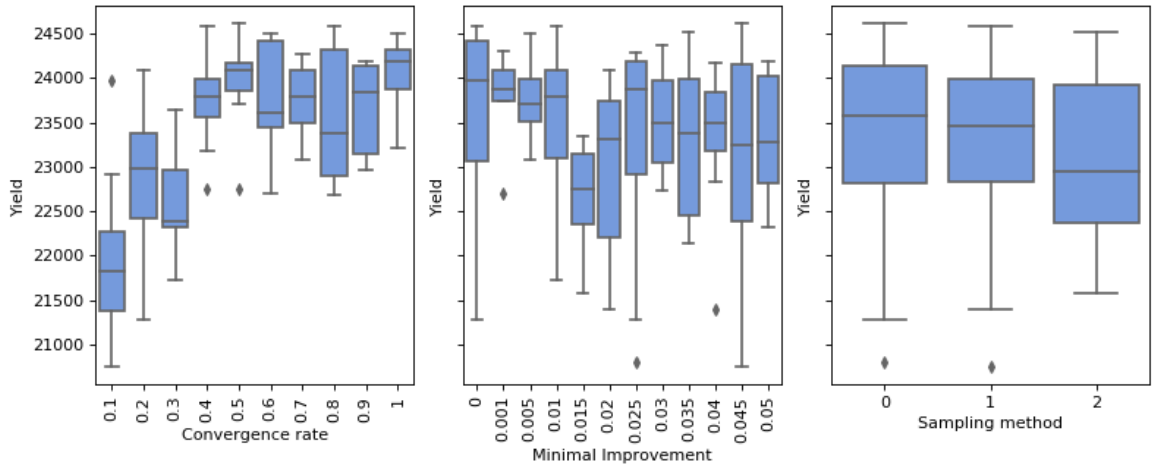

Figure 6.5: Yield of the best recipe depending on the value of different hyperparameters using 100 sample recipes for each hyperparameter.

Table 6.7: Comparison of the description and the yield of the best recipe returned by each method. EK = Expert Knowledge, RS = Random Search, SM = Surrogate Modeling, VC = Virtuous Circle (our framework).

Method	Rain^{P1}	Irrad^{P1}	Wind^{P1}	Rain^{P2}	Irrad^{P2}	Wind^{P2}	Rain^{P3}	Irrad^{P3}	Wind^{P3}	Yield
EK	10	0	5	10	25000	5	10	25000	5	23472
RS	17	23447	8	31	22222	23	39	22385	7	23561
SM	20	44	0	20	24981	0	40	31	30	10170
VC	19	16121	18	25	24052	28	14	21126	7	24336

Comparison with Alternative Methods.

Good hyperparameter values have been defined for our optimization framework and we can now compare our method with other ones. Let us consider the use of expert knowledge and random search. First, we want to create a model using expert knowledge. With the help of an agricultural engineer, we defined a priori good values for each parameter using expert knowledge and we generated a recipe that can serve as a baseline for our experiments. We then choose to compare our method against a random search model without expert knowledge. We set the number of recipes to 150 for all methods to provide a fair comparison with our own model where the number of recipes is set to 150. To account for randomness in the recipe generation, we run 10 iterations of the random search model, we store the value of the best recipe found in each iteration, and we compute their average yield. Results of the experiments and a description of the best recipe for each method are available in Table 6.7. Random search and expert knowledge find recipes with almost equal yields, while our framework finds a recipe with a higher yield. Note that in industrial settings, an improved yield of 3% to 4% has a significant impact on revenues.

Let us now compare our framework to the Surrogate Modeling method presented in (Johnson et al., 2019). To be fair, we give the same number of data points to build the Symbolic Regression surrogate model as we used in previous experiments, i.e., 150 for training the

model (we evaluated the RMSE of the model on a test set of 38 other samples). We use gplearn(Stephens, 2013), with default parameters, except for the number of generations and the number of models evaluated for each generation, which are respectively of 1000 and 2000, as in (Johnson et al., 2019). Note that the model obtained has a RMSE of 2112, and it is composed of more than 2000 terms (including mathematical operators), therefore the argument of interpretability is questionable. A grid search is finally done on this model and we select the best recipe and obtain their true yield using the PCSE simulation environment. The number of steps for each attribute for the grid search has to be defined. We set it to 5. As we have 9 parameters, it means that the model needs to be evaluated on nearly 9 million potential recipes. Also, the model is composed of hundreds of terms such that experiments are computationally expensive. The best recipe found is given in Table 6.7. The surrogate model predicts a yield value of 21137. Compared to the ground truth of 10170, the model has a strong bias. It illustrates that using a surrogate model for this kind of problem will give good recipes only if it is reliable enough. Interestingly, the RMSE seems to be quite good at first glance, but this does not guarantee that the model will behave correctly on all elements of the search space: on the best recipe found, it largely overestimates the yield, leading to a non-interesting recipe. It seems that this method performs poorly on recipes with more attributes than in (Johnson et al., 2019). Further studies are here needed.
We investigated the optimization of plant growth recipes in controlled environments when a single objective needs to be maximized. We motivated the reasons why existing methods fall short of real-life constraints, including the necessity to minimize the number of experiments needed to provide good results. We detailed a new optimization framework that leverages subgroup discovery to iteratively find better growth recipes through the use of a virtuous circle. Let us now deal with multiple target labels at the same time (e.g., optimizing the yield while keeping the energy cost as low as possible.)

6.5 Multi-Objective Plant Growth Recipe Optimization

Although single-objective optimization of growth recipes is interesting and has proven to be effective in optimizing the yield of plants on synthetic data, it ignores the fact that plant growth optimization is an intrinsically multi-objective problem. We are now interested in plant recipe optimization in a multi-objective context. We want to exploit our EPFM methods to show their relevance and actionability on plant growth optimization. We randomly generated 300 recipes of sugar beet using the simulator. Recipes are described by 6 variables (RAIN, IRRAD, WIND, VAP, TMIN, TMAX) and are split into 3 stages (P1, P2, P3), for a total of 18 descriptive variables for each recipe. We then extract the yield and compute the cost of each recipe according to the detailed method in Section 6.3. The value domain of each variable is available in Table 6.8. At the start of our experiments, the domain of a given variable is the same for all 3 stages. As can be seen, the domains are different from those chosen in Section 6.4 of the chapter. This is due to complex interactions between the different weather variables in the synthetic data generator PCSE which forces us to use certain value domains to generate proper growth recipes. Moreover, we used basic agronomic knowledge to choose value domains that will lead to the generation of domain-relevant plant recipes from the start. This is an important part of simulating a real-life plant recipe optimization scenario. Indeed, in real urban farms where recipes need to be optimized, at least basic

agronomic knowledge would be used to generate a first set of recipes, instead of generating completely random and obviously irrelevant solutions.

Table 6.8: Value domain of each environment variable.

RAIN	IRRAD	WIND	VAP	TMIN	TMAX
[5,30]	[10000,30000]	[0,20]	[1.1,1.6]	[15,22]	[23,30]

Randomly generated examples of such recipes can be found in Table 6.9. We kept the number of recipes relatively low on purpose to simulate a real-life urban farm where the number of experiments is limited by numerous constraints. When it comes to discretization of the numerical attributes, we once again apply equal-width discretization using 2, 3 and 5 bins and we retain only the one that leads to the best models.

Table 6.9: Examples of growth recipes split in 3 stages (P1, P2, P3), 6 attributes, and 2 objectives (Yield and Cost).

R	RAIN^{P1}	IRRAD^{P1}	...	RAIN^{P2}	IRRAD^{P2}	...	RAIN^{P3}	IRRAD^{P3}	...	Yield	Cost
r_1	10	23250	...	10	23250	...	15	21000	...	22000	0.56
r_2	35	10000	...	5	25000	...	16	19500	...	20500	0.60
r_3	15	17500	...	22	15000	...	30	4000	...	8600	0.65
r_4	18	22800	...	38	17000	...	38	12000	...	14200	0.7

In the following experiments, we compute the best models returned by our algorithm for EPFDM with HD and HV_{dev}. Figure 6.6 depicts the best model found for each measure. The model found with HD is composed of recipes that highly optimize the yield, but show poor performance cost-wise. The subgroup is described by: $\langle IRRAD^{P3} \in [20000, 30000]$, $TPMAX^{P2} \in [23, 26.5]$, $IRRAD^{P2} \in [20000, 30000]$, $TPMAX^{P3} \in [23, 26.5]\rangle$. It supports what can be observed in the figure: with high values of solar irradiation and maximal temperature during most of the growth process, the yield will be optimized, at the price of a very high cost for the recipes. This exceptional model is interesting since it represents a local interesting part of the Pareto front that can be exploited. Indeed, it seems that this part of the Pareto front contains recipes whose trade-off between yield and cost might not be optimal compared to other parts of the fronts. This is confirmed by the severe deviation in the Pareto front shape once the subgroup is removed. This information can be exploited: when generating new recipes, we can make sure that they do not fall in the description space of the subgroup, lowering the chances of generating recipes with sub-optimal trade-offs. With HV_{dev}, we find a model that creates a large deviation that affects most of the Pareto front. This subgroup is interesting as well. Its description can be exploited to generate new recipes that will provide good trade-offs between yield and cost. The models found with HD and HV_{dev} are complementary when generating new recipes: the first one can be used to exclude parts of the search space where bad recipes exist while the second helps to focus on promising parts. EPFDM is useful as an exploratory tool that enables the discovery of interesting knowledge for MOO problems, however, it cannot be relied upon to design new, more optimized growth recipes.

Let us now consider the exploitation of EPFAM to iteratively optimize the yield-cost trade-off of recipes. It was already shown in Chapter 5 that EPFAM can be used in an iterative

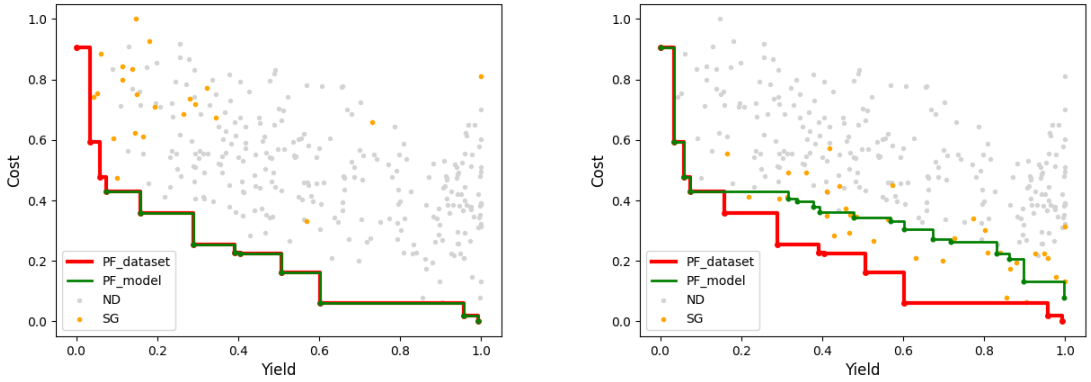

Figure 6.6: EPFDM best models using HD (left) and HV_{dev} (right).

optimization framework to improve trade-offs iteration after iteration. We will use the same method here to optimize recipes: (i) we first run EPFAM to find a good approximation of the overall Pareto front of the original set of recipes (ii) we retrieve the description of the corresponding subgroup (iii) we use the description to apply new restrictions on the domain of values of the corresponding variables (iv) we generate a new set of recipes, and back to (i) if the quality has improved sufficiently. This simple iterative process exploiting EPFAM can be applied successively until no further optimization can be made. It can be seen as a generic virtuous circle, where each new iteration uses information previously gathered to iteratively improve the targeted process, much like the framework introduced in Section 6.4. In this application scenario, we decided to apply this process until we either found two iterations in a row with no improvement in the hypervolume of the dataset or until we reached 10 iterations. Please note that for fair comparison, the hypervolume of each dataset of recipes has to be recalculated after each iteration, since the reference point (built out of the worst values found for each objective out of all the recipes encountered) can change at each generation of a new set of recipes. The best approximation found on the original set of recipes can be observed in Figure 6.7. We find a subgroup whose Pareto front covers a large part of the Pareto front of the dataset. Furthermore, the subgroup covers very few recipes. Its description is $\langle IRRAD^{P1} \in [10000, 20000]$, $TPMAX^{P1} \in [23, 26.5]$, $WIND^{P2} \in [0, 10]$, $TPMAX^{P2} \in [23, 26.5]$, $WIND^{P3} \in [0, 10]\rangle$.

It is concise and understandable, making it easy to exploit to design the set of recipes of the next iteration. We use the description of the best subgroup previously found with EPFAM to apply restrictions on the generation of new recipes. For each environment variable that occurs in the subgroup description, the corresponding restrictions are applied to the values of the newly designed recipes. Then, we generate 300 random new recipes using these restrictions and compute their corresponding yield and cost. The hypervolume of the new dataset is then computed and we check whether it has improved from the previous iteration, which is the case. Following the process, we once again apply EPFAM on the dataset of the second iteration, and so on until we reach 10 iterations or 2 consecutive iterations with no improvement. In this case, the process reached 10 iterations, and let us have an in-depth discussion about its results. Figure 6.8 depicts a comparison between the Pareto front of

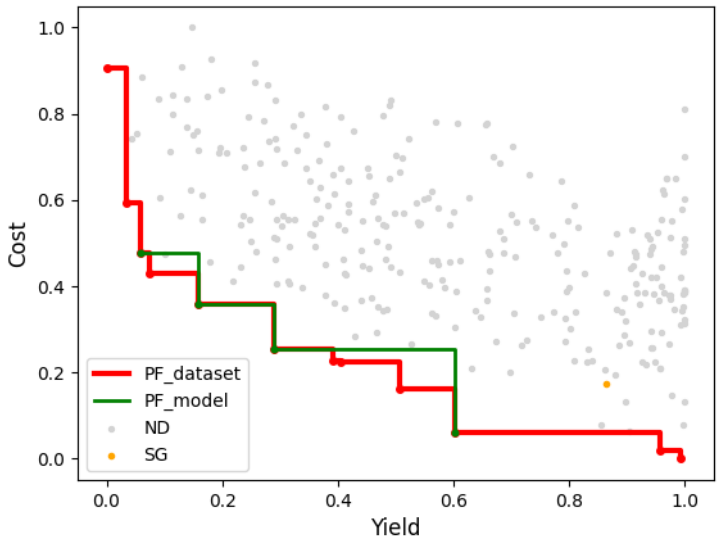

Figure 6.7: Best approximation found using EPFAM.

each of the 10 iterations. First, we can see that the Pareto front improves iteration after iteration, and seems to converge after the ninth or tenth iteration. The improvement during the first iterations is substantial, and then, as we get closer to the hidden true Pareto front, the improvement slows down but continues until the last iteration. The only iteration where no improvement was observed is the fourth iteration, which could be due to the inherent randomness of the recipe generation.

These observations are confirmed by the numbers available in Table 6.10. When studying the hypervolume of the different iterations, we can clearly see a large improvement iteration after iteration, putting aside Iteration 4 where a slight decrease was observed. In the end, the improvement from the first to the last iteration is substantial: the first iteration had a hypervolume of 0.57, while the last iteration of recipes features a hypervolume of 0.88, which represents an improvement of 54% of the quality of the Pareto front. We also observe that the final set of recipes features much better trade-offs than the original set, with the average yield going from 0.62 to 0.35, and the average cost going from 0.60 to 0.19. It is interesting to note that while the standard deviation of the cost improves throughout the process, the standard deviation of the yield remains unchanged.

Let us now discuss the improvement of the yield and cost between the start and the end of our optimization process. Table 6.11 depicts the results. We transformed the normalized values back into their original form, i.e., the yield needs to be maximized and the cost needs to be minimized. As can be seen, the average yield between the original recipes and the final recipes has improved by over 70%. Furthermore, the average cost of each recipe has been lowered by over 30%, allowing us to easily generate recipes with substantially better yield-cost trade-offs than originally. Finally, the standard deviation of both variables has decreased, allowing us to generate very good recipes at a higher rate (i.e., less randomness). It confirms the relevance and actionability of our iterative process to solve MOO problems.

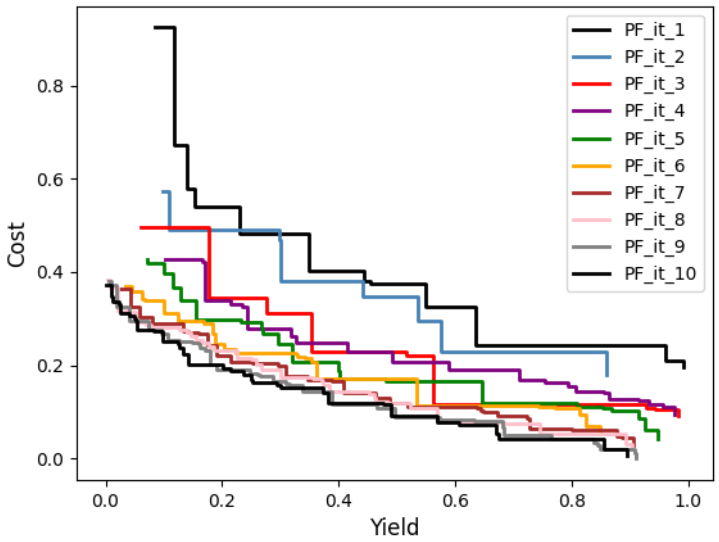

Figure 6.8: Comparison of the Pareto fronts of the 10 iterations of our EPFAM process.

Table 6.10: Comparison of the average, median and standard deviation values of both the yield and the cost, and comparison of the hypervolume between the original set of recipes and the sets of recipes generated at each iteration.

	$Yield_{avg}$	$Cost_{avg}$	$Yield_{med}$	$Cost_{med}$	$Yield_{std}$	$Cost_{std}$	$Hypervolume$
Original recipes	0.62	0.60	0.60	0.59	0.26	0.15	0.57
Iteration 2	0.55	0.54	0.49	0.51	0.24	0.13	0.61
Iteration 3	0.61	0.35	0.60	0.35	0.23	0.1	0.73
Iteration 4	0.54	0.32	0.49	0.33	0.25	0.1	0.70
Iteration 5	0.48	0.26	0.42	0.26	0.25	0.09	0.76
Iteration 6	0.44	0.24	0.37	0.24	0.27	0.09	0.80
Iteration 7	0.41	0.23	0.32	0.22	0.27	0.09	0.84
Iteration 8	0.36	0.22	0.30	0.22	0.25	0.08	0.86
Iteration 9	0.38	0.19	0.29	0.19	0.26	0.08	0.87
Iteration 10	0.35	0.19	0.26	0.19	0.26	0.08	0.88

Table 6.11: Comparison of the average, median and standard deviation non-normalized values of both the yield and the cost between the original and the last set of recipes.

	$Yield_{avg}$	$Cost_{avg}$	$Yield_{med}$	$Cost_{med}$	$Yield_{std}$	$Cost_{std}$
Original recipes	10055	0.54	10640	0.54	7074	0.07
Iteration 10	17338	0.36	19727	0.36	6849	0.04

Although we have demonstrated that our contributions can be exploited to substantially improve the growth of recipes in a multi-objective optimization context, we now want to compare it to a random search method to prove its relevance compared to a well-known search model. Indeed, random search is widely used in numerous optimization applications,

such as hyperparameter optimization, and is known for being very simple to understand and providing good results with a relatively limited amount of objects. To compare those methods, we generate 3000 random recipes, using for each variable the domain of values that were used for the first iteration of our own process. We choose the number 3000 since, in our own process, we ended up generating 3000 recipes (i.e., 300 recipes × 10 iterations). The goal is then to compare the quality of the 3000 randomly generated recipes with that of the last set of recipes of our process. Figure 6.9 depicts the comparison between those 2 sets of recipes. As can be seen, the set of recipes created through our method offers significantly better results than those generated through random search. Moreover, every single recipe generated through our method is better than the Pareto front of the random search (i.e., our 300 recipes are non-dominated by the 3000 random search recipes).

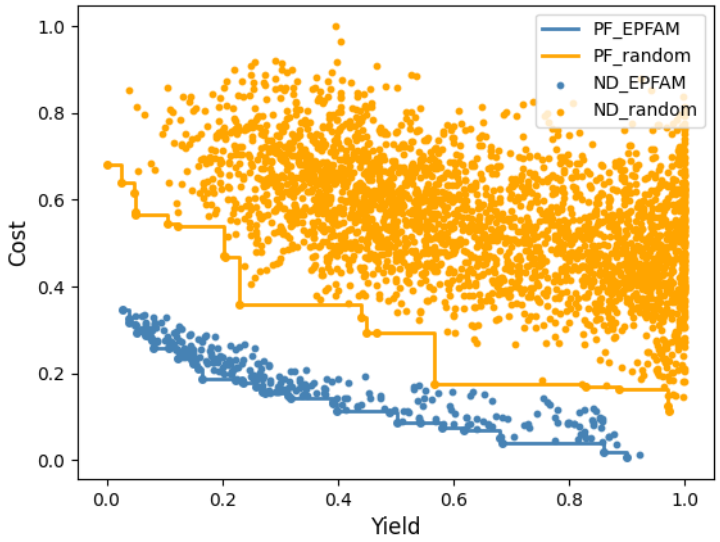

Figure 6.9: Comparison of the yield-cost trade-offs of the recipes generated through the exploitation of EPFAM with the recipes generated through random search.

These observations are confirmed by the numbers available in Table 6.12. Indeed, as can be seen, both the average yield (0.37 vs 0.65) and the average cost (0.18 vs 0.55) of our recipes are much more optimized than those of the random search. Finally, the superiority of our approach is also confirmed by the much better hypervolume of our Pareto front (0.86 vs 0.68). Through this in-depth application scenario to plant growth recipe optimization, we have shown (i) the relevance of our method to solve such problems (ii) the actionability of EPFAM (iii) the superiority of our EMM-based approach compared to a random search model.

Let us go further on recipe optimization by considering now more than two objectives. We generate 300 new recipes using the same process as described before, but this time we also exploit a third objective provided by the PCSE model for each recipe: the total weight of unusable plants (TWP). Indeed, for each recipe, the model computes the amount of usable (that we call the yield, but it actually corresponds to the total weight of storage organs)

Table 6.12: Comparison of the average, median and standard deviation values of both the yield and the cost between our optimized set of recipes and the set of recipes generated through random search.

	$Yield_{avg}$	$Cost_{avg}$	$Yield_{med}$	$Cost_{med}$	$Yield_{std}$	$Cost_{std}$	$Hypervolume$
Iteration 10	0.37	0.18	0.28	0.18	0.25	0.08	0.86
Random recipes	0.65	0.55	0.64	0.55	0.25	0.14	0.68

and unusable produced plants (that we call TWP, and it actually corresponds to the sum of the weights of leaves and stems). Once our new recipes are generated, we run our EPFDM algorithm with HV_{dev} and we report the best computed model. When dealing with Pareto fronts that are more than two-dimensional, one way to study their characteristics is to use scatter plots and visualize the pair-wise relationship of objectives (see Figure 6.10). As it can be seen on each of the 3 scatter plots, the removal of the subgroup leads to large deviations in all 3 pair-wise relationships that compose the overall Pareto front. It is particularly clear in the yield/cost scatter plot where the removal of the subgroup leads to a worse trade-off between yield and cost. The subgroup can be exploited to generate new recipes that not only offer good trade-offs between yield, cost, and TWP, but also offer a better trade-off between yield and cost.

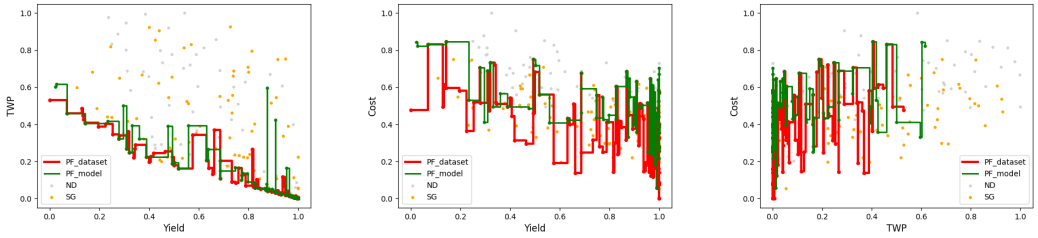

Figure 6.10: Scatter plots of the EPFDM best model with HV_{dev} showing the pair-wise relationship between objectives.

Let us use the EPFAM method also. It is used with HV_{approx} and we report the best computed model in Figure 6.11. We are able to find a small subgroup that approximates very well the overall Pareto front of the problem. It can be used to support the design of new recipes whose trade-off between yield, cost, and TWP will be close to or even on the optimal Pareto front.

Although we have been using an aggregated quality measure (q_{EPFDM} or q_{EPFAM}) up to this point, it can be argued that when a quality measure consists in the multiplication of several objectives (deviation or approximation, locality, generality), loss of information and sub-optimal subgroups may be discovered. Furthermore, when using top-K EMM, the value of K can be difficult to choose, and the top subgroups usually lack diversity. To remedy this problem, we can exploit the concept of *skyline patterns* from (Soulet et al., 2011) to mine for subgroups that offer the best trade-off between the different objectives of our quality measure. Using this method, the optimal number of subgroups returned does not have to be pre-defined, but will instead be a learned parameter of the model. We want to find the skyline of subgroups for EPFDM using HV_{dev}. We choose HV_{dev} since it has shown the best

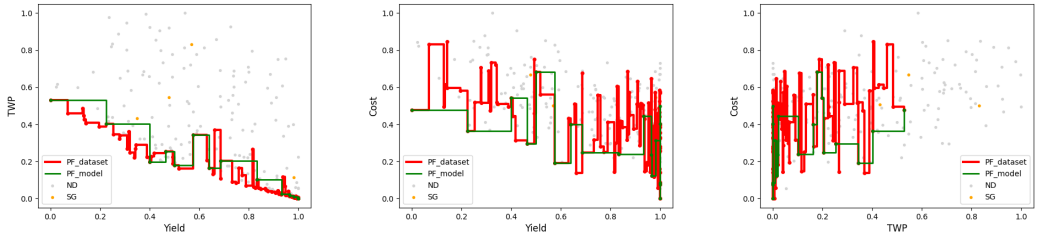

Figure 6.11: Scatter plots of the EPFAM best model with HV_{approx} showing the pair-wise relationship between objectives.

ability to find interesting and actionable subgroups. Furthermore, we choose EPFDM since looking for multiple subgroups in EPFAM makes little sense. Indeed, EPFAM exploits the aggregated measure to support the discovery of very good approximations of the Pareto front: mining a skyline of approximations seems of low interest in that case.
The skyline of subgroups cannot be computed using the algorithms discussed in Chapter 5. Indeed, since no order can be defined on the quality of subgroups that belong to a skyline, a typical beam search strategy where the best q patterns of each level need to be retrieved would not work. Instead, like in (Van Leeuwen and Ukkonen, 2013), we explore the specializations of the best q patterns at each search level, we compute the skyline of patterns of each level, and only the specializations of the *skyline patterns* of the current level are to be explored in the next level. Throughout the exploration, we add only the overall non-dominated patterns to the global skyline that should not be confused with the local skyline of each level. The pseudocode of the algorithm is available in Algorithm 5.

We use this modified version of beam search with a *"dynamic beam-width"* to mine for the skyline of exceptional models. The locality factor is set to 1 and the minimum support to 0.1, such that we have 2 objectives to maximize: the quality and the locality of the subgroup. Since we expect functions that need to be minimized, we transform each maximization into a minimization one. Figure 6.12 (left) depicts the skyline of patterns found using this configuration. Most of the discovered *skyline patterns* have a high locality and a relatively low quality, while some patterns possess a higher quality at the cost of a poorer locality. Next, we want to compare the quality and the locality of the *skyline patterns* with the quality and the locality of the top-K subgroups that are found according to our aggregated measure q_{EPFDM}. To do this, we compute the top-K subgroups using the aggregated measure, and we record the subgroup quality and locality values before multiplying them. To make the comparison as fair as possible, we choose K to be the same as the number of previously found *skyline patterns*, 18 in this case. As it can be seen in Figure 6.12 (right), the found subgroups with the aggregated measure lack diversity between quality and locality and are mostly grouped in the same subspace of the objective space. Furthermore, only one subgroup found by using the aggregated measure dominates the skyline of patterns: it confirms the relevance of skyline pattern mining to find diverse subgroups of high quality.

Finally, we want to estimate the cost of the diversity of patterns mined using Skyline EMM compared to typical top-k EMM in terms of running time. To do this, we record the running time of several configurations of top-K EMM using different values of beam-width. The results are available in Table 6.13. We see that the running times of Skyline EMM and

Algorithm 5 Beam search for Skyline EPFDM

Input: Dataset D, quality measure q, search depth dp, global pareto front pf
Output: List global_skyline

```
1: current_depth ← 0
2: candidate_list ← new list()
3: candidate_list ← candidate_list.insert({})
4: global_skyline ← new list()
5: while (current_depth < dp) do
6:     beam_skyline ← new list()
7:     while (candidate_list ≠ ∅) do
8:         lst_candidates_lvl ← specialize(candidate_list)
9:         for (pattern ∈ lst_candidates_lvl) do
10:            extent ← computeExtent(pattern)
11:            if (extent.isNotDuplicate() and extent.hasObj(pf)) then
12:                complement_extent ← computeComplement(extent)
13:                pareto_front_extent ← computeParetoFront(complement_extent)
14:                quality_extent ← q(pareto_front_extent)
15:                beam_skyline.insert(extent, quality_extent, pareto_front_extent)
16:            end if
17:        end for
18:    end while
19:    skyline_lvl ← computeSkyline(beam_skyline)
20:    candidate_list ← skyline_lvl)
21:    for (skyline_pattern ∈ skyline_lvl) do
22:        if (skyline_pattern.isNotDominatedBy(global_skyline) then
23:            global_skyline.insert(extent, quality_extent, pareto_front_extent)
24:        end if
25:    end for
26:    current_depth = current_depth + 1
27: end while
28: return global_skyline
```

top-K EMM cross each other when the beam-width is set to 10, which is the most common configuration used in our experiments throughout this paper. It shows that the diversity of Skyline EMM is obtained without a negative impact on running time.

Table 6.13: Running time comparison (in seconds) of EPFDM between q_{EPFDM} denoted q and Skyline. bw is the chosen beam width when using q_{EPFDM}.

Skyline	$q, bw = 1$	$q, bw = 3$	$q, bw = 5$	$q, bw = 10$	$q, bw = 20$	$q, bw = 50$
878	121	308	485	917	1713	3631

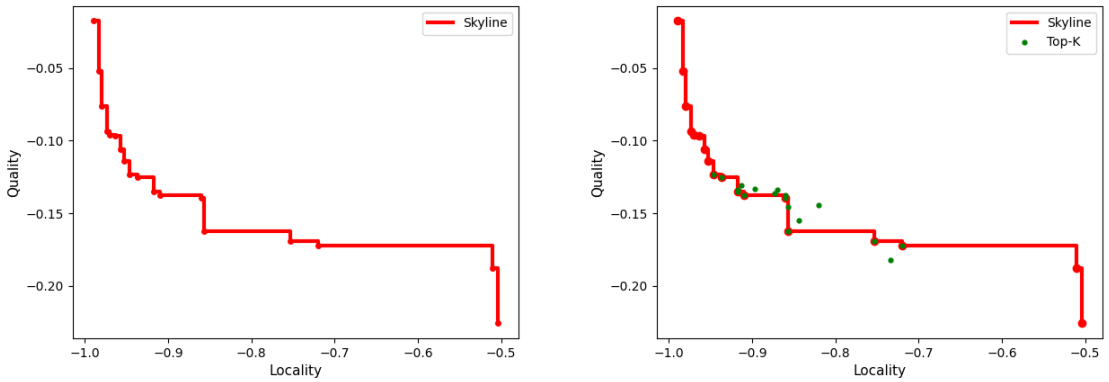

Figure 6.12: Skyline of EPFDM (left) and comparison between Skyline and q_{EPFDM} (right).

6.6 A Real-Life Application Scenario to Plant Growth Recipe Optimization

6.6.1 Experimental Design

Following our previous contributions on synthetic data, we set out to design an experimental plan that would allow us to create and collect data on real-life recipes of basil. These experiments were done in collaboration with our partner FUL, who agreed to lend us a large part of their urban farm for the purpose of testing our different methods of growth recipe optimization late 2020.

The first thing that we needed to define was which environment variables could be controlled to create the growth recipes. Due to constraints specific to the farm, we were able to work with 4 environment variables. The variables were the LED light intensity (in %) which could take values in 20, 36, 52, 68, 84, 100, the photoperiod with discrete values taken in [7,24], the number of seeds by pot with discrete values taken in [1,50], and the number of pots by $1/8^{th}$ square meter with discrete values taken in [3,15]. We had access to 41 plant trays, that could each contain up to 60 pots. Each tray was divided into 4 equal parts, such that we could grow 4 different recipes by tray (one recipe for each sub-part of the tray). This allowed us to run $41 \times 4 = 164$ growth recipes at the same time. Figure 6.13 depicts an example of plant tray used in our experiments. The division of each tray into 4 equal parts was decided for several reasons: (i) it was necessary to be able to grow a sufficient number of recipes at the same time, and collect enough data (ii) we could not divide the trays into more than 4 zones, or else it would have introduced a strong bias between the different recipes of each tray. This is because in real-life conditions the amount of LED light received is different for each part of the tray. For example, outer zones receive less light than central zones. The recipes were also kept simple, we worked with a unique growing stage (i.e., the environment stayed the same throughout the plant development). Next, the recipes were generated through random sampling for each variable. Finally, the total yield, leaf yield – both in grams of leaves per pot – and height (in centimeters) of each of the 164 recipes would be measured after harvest. Initially, 3 iterations of plant growth were planned such that we could apply our optimization

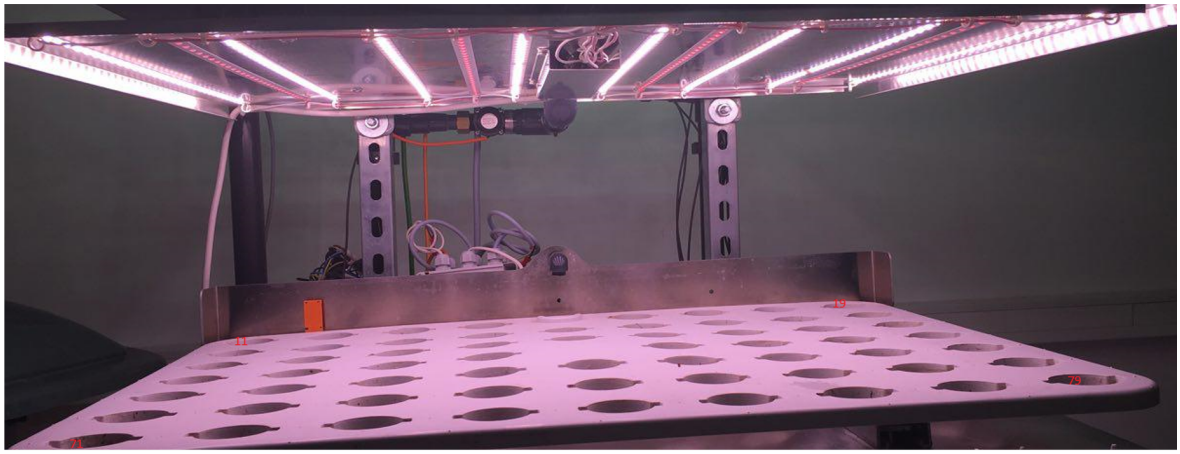

Figure 6.13: Example of a FUL plant tray used in our experiments.

frameworks and observe their relevance over multiple cycles, but due to circumstances outside of our control, the experiments had to be stopped after the first iteration, leaving us with only one set of growth recipes on which we could test our methods.

In order to compare our methods with state-of-the-art growth recipes, several expert recipes were designed by FUL and grown at the same time than our statistically generated recipes. Six plant trays divided into two zones were used to design 12 expert recipes. Each tray had its own level of light intensity, and all the trays had the same photoperiod (18 hours per day) and number of pots by zone (14). Finally, on each tray, 2 recipes with a different number of seeds by pot were grown (3 seeds and 44 seeds every time). These recipes would be used as a baseline that our optimization frameworks should try to beat.
Once the first cycle of growth was done, we were able to retrieve and analyze the resulting data provided by FUL. Due to several issues during the harvesting and measuring of the recipes, we only had access to complete data on 135 recipes out of the original 164. Furthermore, only 4 expert recipes could be retrieved (those with light intensities of 68% and 84%), which limits the relevance of a comparison with our methods (since the best expert recipes might be missing from the data).

6.6.2 Single-Objective Optimization of Recipes

We first want to investigate the exploitation of our single-objective optimization methods on the resulting basil recipes. Since we have access to several objective attributes (lead yield, total yield, height), we create separated datasets containing each one of the objectives, so that we can run `OSMIND` on them. this way, we can compare the different optimal subgroups of recipes found depending on which objective we work with. This is also interesting because if we find the same subgroup for 2 different objectives, it would mean that optimizing one objective also optimizes the other.
We start by studying the dataset using the height objective. Table 6.14 depicts the descriptions and average height of the dataset comprised of the 135 recipes generated and the optimal subgroup of recipes found using `OSMIND`. As can be seen, we find a subgroup that optimizes

well the height of the plants. Furthermore, the subgroup possesses a clear description, from which we can infer that recipes that optimize the height need relatively high light intensity and photoperiod. The description of the subgroup could also be exploited to generate new recipes with a more optimized height.

Table 6.14: Comparison between descriptions of the overall dataset (DS) and the optimal subgroup of recipes returned by OSMIND when optimizing the height of the plants.

Subgroup	Light Intensity	Photoperiod	Nb seeds by pot	Nb pots by zone	Height
DS	[20,100]	[7,24]	[1,50]	[3,15]	14.2
OSMIND	[68,100]	[13,24]	[1,50]	[3,15]	20.6

We now study recipes and their total yield. Table 6.15 depicts the descriptions and average total yield of the dataset and the optimal subgroup of recipes found using the OSMIND. We find a subgroup that optimizes very well the total yield of the plants compared to the average yield of the dataset (33.6 vs 16). From its description, we can infer that recipes that optimize the total yield need relatively high light intensity and photoperiod, but also a larger quantity of seeds by pot and a lower number of pots by zone. Although the first 3 restrictions seem easy to explain (i.e., more light and seeds lead to higher yield), the restriction on the upper bound for the number of pots by zone might seem odd at first. Expert knowledge actually explains this phenomenon: when too many plants grow close to each other, they have to fight for light, such that only a few recipes end up receiving most of the light, while the rest of the recipes do not develop well. For this reason, recipes that achieve the best average yield are the ones with a lower density of pots by zone. We could also once again exploit the description to easily generate new recipes with an optimized total yield.

Table 6.15: Comparison between descriptions of the overall dataset (DS) and the optimal subgroup of recipes returned by OSMIND when optimizing the total yield of the plants.

Subgroup	Light Intensity	Photoperiod	Nb seeds by pot	Nb pots by zone	Total Yield
DS	[20,100]	[7,24]	[1,50]	[3,15]	16
OSMIND	[68,100]	[13,24]	[10,50]	[3,11]	33.6

We now set out to find a subgroup of recipes that optimizes the leaf yield. Indeed, where the total yield contains the whole plant, the leaf yield contains only the part of the plant which is exploitable (i.e., which can be sold). Table 6.16 depicts the descriptions and average leaf yields of the dataset and the optimal subgroup of recipes found. We find a subgroup of recipes that highly optimizes the average leaf yield of basil. Indeed, the average yield by pot in the overall dataset is 11.7 grams, while the average yield of the optimal subgroup of recipes is 22.5 grams. Interestingly, the subgroup is supported by a subgroup with almost exactly the same description as the subgroup optimizing the total yield. This is interesting since it means that optimizing the total yield leads to also optimizing the leaf yield. While at first this might seem like an obvious conclusion, the results could have been completely different if, for example, recipes with a high total yield were mostly made of unexploitable matter. It is also interesting to note that the subgroup is different from the one found when optimizing the height. This means that growing recipes with an optimized height does not lead to optimizing the yield, contrary to what could have been expected.

Table 6.16: Comparison between descriptions of the overall dataset (DS) and the optimal subgroup of recipes returned by OSMIND when optimizing the leaf yield of the plants.

Subgroup	Light Intensity	Photoperiod	Nb seeds by pot	Nb pots by zone	Leaf Yield
DS	[20,100]	[7,24]	[1,50]	[3,15]	11.7
OSMIND	[68,100]	[14,24]	[10,50]	[3,11]	22.5

Finally, while we have been studying the average leaf yield by pot, it ignores the fact that when optimizing the yield, an urban farm operator would likely be interested in the yield by square meter, and not the yield by pot. We therefore compute the leaf yield by square meter from the leaf yield for each of the 135 recipes. To do this, for each recipe, the leaf yield is multiplied by the number of pots, and this resulting number is multiplied by 8 (i.e., because a recipe occupies $1/8^{th}$ of a square meter). For example, for a recipe with a leaf yield of 10 grams and 5 pots by zone, we compute the leaf yield by square meter this way: $10 \times 5 \times 8 = 400$ grams by square meter. Next, we look for the optimal subgroup of recipes optimizing the leaf yield by square meter. Table 6.17 depicts the descriptions and average leaf yields by square meter of the dataset and the optimal subgroup of recipes found. Once again, we find a subgroup of recipes that highly optimizes our objective. What is interesting is that the description of the subgroup is completely different from the descriptions encountered up until now. To optimize the yield per square meter, we need a lower photoperiod, a lower number of seeds, and a higher density of pots by zone. This means that despite the plants having to fight for light, having a higher density of pots still leads to a higher overall yield per square meter, contrary to the conclusion that had been made when optimizing the leaf yield by pot.

Table 6.17: Comparison between descriptions of the overall dataset (DS) and the optimal subgroup of recipes returned by OSMIND when optimizing the leaf yield per square meter of the plants.

Subgroup	Light Intensity	Photoperiod	Nb seeds by pot	Nb pots by zone	Leaf Yield (M2)
DS	[20,100]	[7,24]	[1,50]	[3,15]	779
OSMIND	[68,100]	[10,24]	[6,48]	[8,15]	1495

To conclude, we want to compare the results of our algorithm with the 4 expert recipes that were retrieved from the experiments. To do this, we compute for each expert recipe its leaf yield per square meter, and we retain only the recipe with the best yield, which we compare to the yield of our optimal subgroup. The best expert recipe has a leaf yield of 12 grams per pot, and a density of 14 pots by zone. We compute its leaf yield by square meter, and we find $12 \times 14 \times 8 = 1344$ grams per square meter. Interestingly, the average yield of our optimal subgroup of recipes is 1495, which is already better than the best expert recipe that was provided to us. There is however an important caveat to these conclusions: we were not provided with most of the experts recipes – 8 were missing out of 12 – meaning that other expert recipes could have potentially contained better recipes than the 4 we have at hands. Finally, although the scope of our experiments on real-life recipes is definitely limited due to circumstances, we have shown on preliminary data the relevance and actionability of our contributions to help solve single-objective optimization problems akin to that of plant growth

recipe optimization in controlled environments.

6.6.3 Multi-Objective Optimization of Recipes

We now want to investigate the exploitation of our EPFM methods on the basil growth recipes. Although we can compute the leaf yield (in grams per square meter) of each recipe, we were not provided with the cost of the recipes. We therefore need to find a way to compute a cost for each recipe, which will allow us to apply our EPFDM and EPFAM methods to try to extract relevant information. Since we have little information to work with, we decide to define the cost as the multiplication of the light intensity (divided by 100) by the photoperiod (in hours/day). For example, for a recipe with 20% light intensity and a photoperiod of 18 hours, we compute the cost the following way: $\frac{20}{100} \times 18 = 3.6$. We then compute the cost of each recipe using this technique. When it comes to discretization of the numerical attributes, we apply equal-width using 3 bins for EPFDM and equal-width using 5 bins for EPFAM, since it was shown to provide the best theoretical results on synthetic data in Chapter 5.

In the following experiments, we compute the best models returned by our algorithm for EPFDM with HD and HV_{dev}. Figure 6.14 depicts the best model found for each measure. The model found with HD is composed of recipes with low energy cost and low leaf yield. The description of the subgroup is the following: $light\_intensity \in [19.92, 46.67]$. It supports our previous observations; when the light intensity is set to a low value, we produce recipes of low yield and energy cost. This subgroup can be exploited to easily generate new recipes of low cost and low yield if it is of interest to the expert. Next, we study the best model found with HV_{dev}. As can be seen in the figure, we find a large subgroup that creates an important deviation of the true Pareto front. The subgroup represents recipes with good trade-offs between cost and yield. Its description is $photoperiod \in [6.98, 12.67]$. We can infer that recipes with a relatively low photoperiod create plants with a good trade-off between yield and cost. The description of the subgroup can easily be exploited to generate new recipes of good quality.

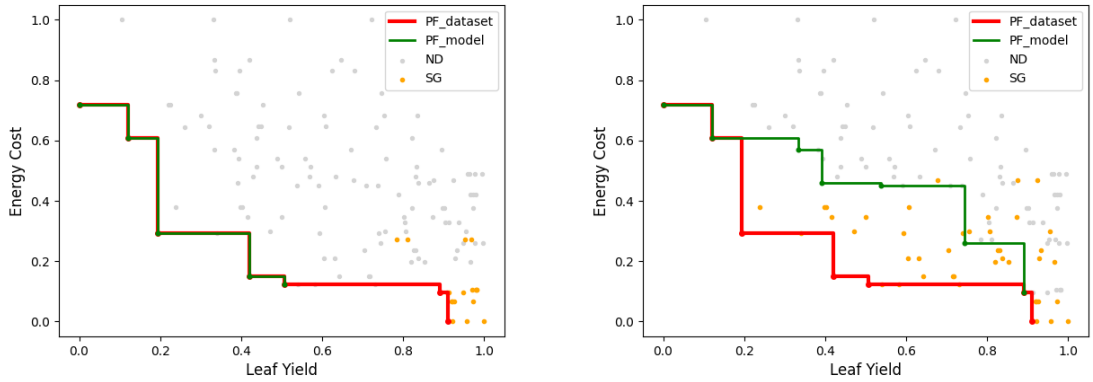

Figure 6.14: EPFDM best models using HD (left) and HV_{dev} (right).

Next, we want to exploit EPFAM to find a good approximation of the true Pareto front of recipes. Figure 6.15 (left) depicts the best model found. As can be observed, we find a small subgroup (<10% of the dataset) which is a good approximation of the global Pareto

front. This is confirmed by its high quality of 0.82. Through its description – *photoperiod* $\in [6.98, 10.4]$ *and nb_pots_by_zone* $\in [12.6, 15]$ – we can infer that recipes with a high density of pots and a low photoperiod lead to plants with good trade-offs between leaf yield and cost. Information could be exploited to generate new, better recipes in an iterative optimization framework, as was done in Section 6.5. Unfortunately, we were not able to test such an EPFAM-based framework further due to circumstances outside of our control.

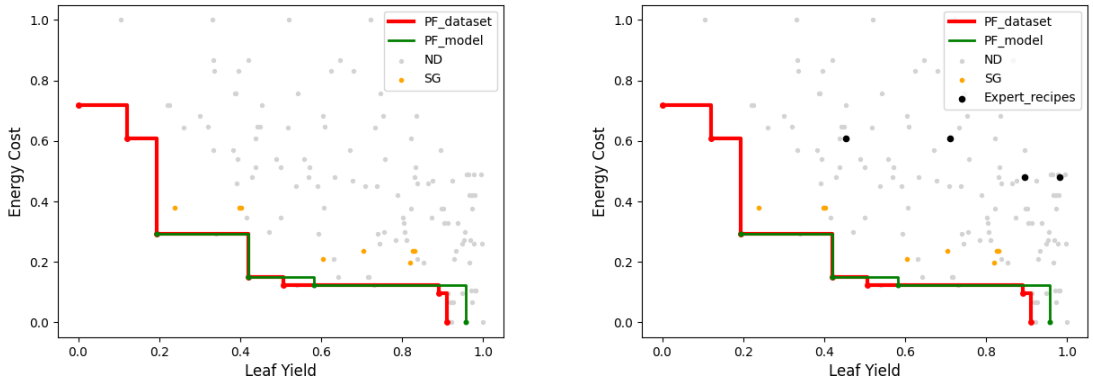

Figure 6.15: EPFAM best model (left) and comparison of the generated recipes with the expert recipes (right).

Finally, we want to compare the expert recipes with the generated recipes and the best model found with EPFAM. While in the previous section we found that the best expert had a leaf yield close to that of our optimal subgroup, we now compute the cost of the expert recipes to have a better idea of their quality in a multi-objective optimization setting. Figure 6.15 (right) depicts the expert recipes compared to the other recipes in the objective space. We can observe that expert recipes provide bad trade-offs between energy cost and leaf yield compared to the recipes of the subgroup found with EPFAM. It confirms that by exploiting the best approximation, we could generate new recipes which already provide better trade-offs than those created by the experts.

While these experiments are a good start to confirm the relevance of our work, it suffers from several limitations: (i) the number of environment variables that could be modified were limited (ii) we were only able to generate one iteration of recipes, which severely limits our capacity to assess the quality of our iterative frameworks (iii) we did not have a clean and precise way to compute the cost of the recipes (iv) the recipes only had one stage, compared to 3 stages in our synthetic scenarios (v) some expert recipes were missing, which made the comparison with our method less relevant.

6.7 Conclusion

In this chapter, our focus was to demonstrate how the contributions introduced in the previous chapters can be exploited in a real-life application scenario, namely plant growth recipe optimization in controlled environments. We first started by introducing all the necessary concepts, including the concept of plant growth recipe. We then introduced a synthetic data

generator which allows for the simulation and generation of growth recipes in a controlled environment.
We investigated the optimization of plant growth recipes in controlled environments when only one objective is considered. We motivated the reasons why existing methods fall short of real-life constraints and we detailed a new optimization framework that exploits subgroup discovery to iteratively generate better growth recipes.
Next, we introduced a promising use case where EPFM can be used to support plant growth recipe optimization in a multi-objective setting. Both EPFDM and EPFAM have been applied on an "in silico" approach to recipe optimization. Through this in-depth application scenario, we have shown (i) the relevance of our methods to help solve such problems (ii) the high actionability of EPFAM (iii) the superiority of our EPFM-based approach compared to random search. When considering this use case, we have also considered Skyline Exceptional Model Mining. It offers a better diversity of subgroups over top-K Exceptional Model Mining, and it does not require for the number of returned subgroups to be known beforehand.
Thanks to temporary access to a real-life FUL urban farm, we were able to generate actual growth recipes in a controlled environment. Preliminary results confirm the actionability of our methods to optimize recipes, both in single-objective and multi-objective optimization settings.

Chapter 7

Conclusion

7.1 Summary

In this thesis, we investigated the design of pattern discovery methods that allow for the discovery of relevant parameter values driving the optimization of a process. Mining purely numerical data is becoming ever more popular. It concerns data made of objects described by numerical attributes, and one of these attributes can be considered as a target label. We considered Subgroup Discovery, a task that aims at discovering subsets of objects in a dataset – subgroups – whose target label distribution statistically deviates from that of the overall dataset. While a large panel of SD algorithms has been proposed so far, most of these approaches consider a set of nominal attributes with a binary label. SD with numerical data has historically been of relatively low interest for researchers, with few contributions existing in the literature.

We investigated the optimal SD with respect to a quality measure in purely numerical data and motivated the reasons why existing methods achieve suboptimal results and lead to critical loss of information by employing discretization techniques. To achieve the search for optimality, we decided to search efficiently the space of interval patterns as defined in (Kaytoue et al., 2011). Our first contribution (Chapter 4) consists in the `OSMIND` algorithm that enables optimal subgroup discovery. Our approach (i) exploits the concept of closure on the positives adapted to a numerical setting to operate in a subspace (ii) uses a new faster tight optimistic estimate that can be applied for several quality measures (iii) uses advanced pruning techniques (forward checking, branch reordering). The empirical evaluation has illustrated the added-value and the exploitability of the `OSMIND` algorithm when compared to the state of the art algorithm `SD-Map*`.

While `OSMIND` is certainly a good first approach for the discovery of good parameter values optimizing a single-objective process, the reality is that most processes involve several objectives that need to be optimized concurrently. For this reason, we next put our focus on Exceptional Model Mining, a generalization of SD that can deal with complex problems where several objectives are involved. In EMM, we look for subgroups whose models deviate significantly from the same models fitted on the entire dataset. Examples of complex interactions between variables can be found such that they can support Multi-objective Optimization. While MOO possesses a large literature, typical approaches can not be exploited when the underlying model is unknown and/or experiments are limited due to time and cost

constraints. There was therefore a need for methods that would support the discovery of relevant and exploitable information in such problems.
We investigated the cross-fertilization between EMM and Pareto-based MOO by designing a generic model class for Exceptional Pareto Front Mining (EPFM) (Chapter 5). While other approaches that link pattern discovery to MOO work at the pattern level, EPFM is able to find relevant patterns at the object level. Our first approach, EPFDM, looks for deviations in the shape of the Pareto front created by the absence of a subgroup of objects, compared to the same Pareto front computed on the whole dataset. Our second approach, EPFAM, looks for subgroups whose Pareto front approximates exceptionally well the true Pareto front. EPFDM can be used as an exploratory analysis tool to discover interesting nuggets of knowledge for MOO problems, and EPFAM can be exploited to find exceptionally good approximations of the true Pareto front. In other words, EPFAM enables the generation of Pareto optimal solutions with a higher probability. The relevance and actionability of EPFM was validated on an application scenario to hyperparameter optimization in Machine Learning.
Next, we investigated the actionability of our contributions to SD and EMM for plant growth recipe optimization in controlled environments like vertical urban farms (Chapter 6), the real-life setting that has motivated our research. Plant growth optimization is an MOO problem by essence. Indeed, in such controlled environments, when trying to optimize the quality or quantity of plants, other variables like the energy cost have to be taken into account. Therefore, optimizing recipes means finding the ideal set of environment parameter values that lead to the best trade-offs between several concurrent objectives. This is a complex task: in recipe optimization, the underlying model that governs the growth is unknown, and experiments are severely limited due to time and cost constraints, rendering common MOO approaches unusable.
We first detailed how an existing crop simulator can be exploited to generate synthetic recipes that replicate a controlled environment and assist the empirical validation of our work. We then investigated the optimization of plant growth recipes in controlled environments when only one objective is considered. We motivated the reasons why existing methods fall short of real-life constraints and we detailed a new optimization framework – based on a virtuous circle principle – that exploits subgroup discovery to iteratively generate better growth recipes. Next, we introduced a promising use case where EPFM can be used to support plant growth recipe optimization in a multi-objective setting. Both EPFDM and EPFAM have been applied on an "in silico" approach to recipe optimization. Through this in-depth application scenario, we have shown (i) the relevance of our methods to help solving such problems (ii) the high actionability of EPFAM (iii) the superiority of our EPFM-based approach compared to random search. Finally, we were able to generate actual growth recipes in a controlled environment thanks to short time access to a real urban farm, which allowed us to assess the relevance of our contributions in a real-life urban farm setting. Our preliminary results have confirmed the potential of our methods to optimize recipes, both in single-objective and multi-objective optimization settings.

7.2 Perspectives

7.2.1 Pattern Discovery Methods

Enhancing and diversifying `OSMIND`.

From an algorithmic perspective, the main research topic would concern either the enhancement of `OSMIND` scalability for high-dimensional datasets, or the development of a new efficient algorithm for Optimal SD in high-dimensional numerical data. Indeed, it has been shown that the current iteration of the algorithm struggles with even medium-sized datasets, which severely limits its application potential. Furthermore, there is still no proper algorithm in the literature for exhaustive SD in high-dimensional numerical data without pre-discretization. Second, our algorithm is mostly built around the q^a_{mean} family of quality measures. Indeed, the closure system, as well as the pruning system that we developed directly use properties that are specific to these measures. Therefore it would be interesting to explore other common quality measures (e.g., t-score, AUC, median), and see if equivalent bounds and compressing techniques can be found.
Next, the closure scheme that we currently exploit computes the most restrictive closed-on-the-positive patterns. This means that the descriptions of the subgroups contain as many restrictions on attributes as possible. While this was relevant to our setting where having more information to build better recipes is interesting, we also know that many scenarios involve discovering the shortest subgroup descriptions. We could therefore rework our closure scheme so that the algorithm returns the shortest descriptions available.
Finally, the techniques used for optimal SD in purely numerical data could be extended to EMM, where no equivalent algorithm currently exists to perform this task. Indeed, techniques for search space compression – such as closure systems and equivalence classes – and search space pruning – i.e., optimistic estimates and advanced techniques like forward checking and branch reordering – could be included in an efficient algorithm to render an exhaustive search tractable.

Going further with Exceptional Pareto Front Mining.

While this first proposition for EPFM is a fairly good first step, it would be interesting to investigate the design of new approaches to the model class. For example, we could compare the Pareto front of a subgroup with its complement, which would involve discovering larger subgroups whose models are exceptionally different from that of their complements. In its current version, EPFM has been used with a greedy algorithm, namely beam search. It could therefore be interesting to investigate exhaustive approaches. This would be especially interesting for EPFAM, where finding the optimal subgroup that best approximates the true Pareto front is crucial.
Next, we notice the lack of diversity in the subgroups that are returned by the current iteration of our algorithm. Therefore, digging deeper into the Skyline EPFM concept to discover a set of non-redundant models would be interesting, particularly for EPFDM, where finding a set of different local deviations in the Pareto front is much more interesting. Finally, it would also be of interest to create new quality measures for Pareto front comparison based on

the MOO literature. Indeed, our current preferred method employs the hypervolume, whose scalability limitations are well-known.

7.2.2 Optimization Frameworks

Revamping the virtuous-circle-based optimization framework.

The current version of our iterative optimization framework involves pure exploitation, which compromises its ability to discover optimal solutions. Indeed, since our framework only exploits information about the best subgroup at each step, it most likely finds a local optimum. However, expanding our method by adding an exploration step would improve the diversity of the solutions explored during the search, maximizing our chances of discovering the global optimum.
Next, the integration of expert knowledge is currently limited to inputting the bounds of each variable for the first iteration of the process. Improving the interactiveness of our framework by allowing the expert to apply a more diverse set of constraints throughout the process could help in finding better solutions. To go even further, it would be interesting to allow the inclusion of more advanced expert information into the framework, such as models describing subsystems of complex processes. For example, in plant growth optimization, the relationships between pairs of variables – such as hygrometry and temperature – and their effects on plant development are well-known by experts. Being able to introduce such information about complex interaction would therefore be extremely relevant. Finally, our framework is agnostic with regard to the considered SD algorithm, which means that any other method that can deal with numerical targets can be employed. However, we have not investigated more complex optimization problems, that could involve nominal targets instead of numerical ones, or even a combination of numerical and nominal targets to be optimized at the same time. For example, in plant growth optimization, one objective could involve a label regarding a qualitative aspect of the plants (e.g., the color of the leaves, the taste). In this setting, being able to optimize qualitative and quantitative aspects simultaneously would be crucial.

Formalizing and improving the multi-objective optimization framework.

The integration of our method in a properly formalized EMM-based iterative optimization framework seems like a logical next step to fully exploit its potential in MOO. Indeed, the basic iterative framework which was introduced in this thesis has not been properly formalized. Furthermore, it only considers leveraging the best approximation (i.e., the best subgroup) found at each step. This is a method based on pure exploitation which certainly lacks diversity and is only able to find locally optimal solutions. Introducing exploration techniques into this process would increase the diversity of the explored solutions, and therefore maximize the probability of discovering Pareto optimal solutions.

To conclude, while this thesis was a good first step toward exploiting and maximizing the actionability of Pattern Discovery (i.e., SD and EMM here but other methods could be used) to help solve complex optimization problems, further research that may involve more complex methods, data, and optimization problems will be interesting to investigate in the future.

7.2.3 Plant Growth Recipe Optimization

In our work, we have considered large urban farms that can be exploited by economical entities to mass-produce plants and vegetables. In this setting, optimizing the yield and minimizing the cost simultaneously is extremely relevant. However, plant growth in controlled environments does not have to be limited to this setting. Tomorrow, we can imagine smaller farms like phytotrons that can be used for medical research on high-value plants, or even smaller-sized environments that could be used as private gardens to feed families. In these settings, optimizing the yield could be less relevant, but designing plant growth recipes would still be of interest. Indeed, medical researchers could be interested in optimizing the chemical composition of the plants, while individuals could be interested in producing vegetables with a particular taste, scent, or color that corresponds to their preference. It is also known that some plants develop different properties depending on the harshness of their growing environment. For example, some plants that are put in environments with fewer nutrients or less light (which is closer to real-life uncontrolled conditions), can develop much different tastes than those that are grown in abundant environments (Oliveira et al., 2013). Optimized recipes could therefore be designed such that plants develop in more or less harsh environments to develop specific interesting properties.

Bibliography

Tarek Abudawood and Peter Flach. Evaluation measures for multi-class subgroup discovery. In *Proceedings ECML/PKDD*, pages 35–50. Springer, 2009. 15

Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. Mining association rules between sets of items in large databases. In *Proceedings ACM SIGMOD*, pages 207–216, 1993. 3, 12

Rakesh Agrawal, Heikki Mannila, Ramakrishnan Srikant, Hannu Toivonen, A Inkeri Verkamo, et al. Fast discovery of association rules. *Advances in Knowledge Discovery and Data Mining*, 12(1):307–328, 1996. 12

Ines Alaya, Christine Solnon, and Khaled Ghedira. Ant colony optimization for multi-objective optimization problems. In *Proceedings ICTAI*, volume 1, pages 450–457. IEEE, 2007. 43

Jesús Alcalá-Fdez, Luciano Sánchez, Salvador Garcia, Maria Jose del Jesus, Sebastian Ventura, Josep Maria Garrell, José Otero, Cristóbal Romero, Jaume Bacardit, Victor M Rivas, et al. Keel: a software tool to assess evolutionary algorithms for data mining problems. *Soft Computing*, 13(3):307–318, 2009. 20

Ehsan Asadi, Manuel Gameiro Da Silva, Carlos Henggeler Antunes, and Luís Dias. Multi-objective optimization for building retrofit strategies: A model and an application. *Energy and Buildings*, 44:81–87, 2012. 48

Timothy Ward Athan and Panos Y Papalambros. A note on weighted criteria methods for compromise solutions in multi-objective optimization. *Engineering Optimization*, 27(2): 155–176, 1996. 39

Martin Atzmueller. Subgroup discovery. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 5(1):35–49, 2015. 12

Martin Atzmueller and Florian Lemmerich. Fast subgroup discovery for continuous target concepts. In *Proceedings ISMIS*, pages 35–44. Springer, 2009. 6, 27, 53

Martin Atzmueller and Florian Lemmerich. Vikamine–open-source subgroup discovery, pattern mining, and analytics. In *Proceedings ECML/PKDD*, pages 842–845. Springer, 2012. 20

Martin Atzmueller and Frank Puppe. Sd-map – a fast algorithm for exhaustive subgroup discovery. In *Proceedings ECML/PKDD*, pages 6–17. Springer, 2006a. 14, 24, 28

Martin Atzmueller and Frank Puppe. A methodological view on knowledge-intensive subgroup discovery. In *Proceedings EKAW*, pages 318–325. Springer, 2006b. 17

Martin Atzmueller and Frank Puppe. A case-based approach for characterization and analysis of subgroup patterns. *Applied Intelligence*, 28(3):210–221, 2008. 20

Martin Atzmueller, Frank Puppe, and Hans-Peter Buscher. Towards knowledge-intensive subgroup discovery. In *Proceedings LWA*, pages 111–117, 2004. 17

Martin Atzmueller, Frank Puppe, and Hans-Peter Buscher. Exploiting background knowledge for knowledge-intensive subgroup discovery. In *Proceedings IJCAI*, pages 647–652, 2005. 17

Martin Atzmueller, Stephan Doerfel, and Folke Mitzlaff. Description-oriented community detection using exhaustive subgroup discovery. *Information Sciences*, 329:965–984, 2016. 21

Yonatan Aumann and Yehuda Lindell. A statistical theory for quantitative association rules. In *Proceedings ACM SIGKDD*, page 261–270, 1999. 26

Sanghamitra Bandyopadhyay, Sriparna Saha, Ujjwal Maulik, and Kalyanmoy Deb. A simulated annealing-based multiobjective optimization algorithm: Amosa. *IEEE Transactions on Evolutionary Computation*, 12(3):269–283, 2008. 44

Yves Bastide, Rafik Taouil, Nicolas Pasquier, Gerd Stumme, and Lotfi Lakhal. Mining frequent patterns with counting inference. *ACM SIGKDD Explorations Newsletter*, 2(2): 66–75, December 2000. 19

Stephen D Bay and Michael J Pazzani. Detecting group differences: Mining contrast sets. *Data Mining and Knowledge Discovery*, 5(3):213–246, 2001. 12

Adnene Belfodil, Sylvie Cazalens, Philippe Lamarre, and Marc Plantevit. Flash points: Discovering exceptional pairwise behaviors in vote or rating data. In *Proceedings ECML/PKDD*, pages 442–458. Springer, 2017. 34

Adnene Belfodil, Aimene Belfodil, Anes Bendimerad, Philippe Lamarre, Céline Robardet, Mehdi Kaytoue, and Marc Plantevit. Fssd-a fast and efficient algorithm for subgroup set discovery. In *Proceedings IEEE DSAA*, pages 91–99, 2019a. 14, 18

Adnene Belfodil, Wouter Duivesteijn, Marc Plantevit, Sylvie Cazalens, and Philippe Lamarre. Deviant: Discovering significant exceptional (dis-) agreement within groups. In *Proceedings ECML/PKDD*, pages 3–20. Springer, 2019b. 34

Aimene Belfodil, Adnene Belfodil, and Mehdi Kaytoue. Anytime subgroup discovery in numerical domains with guarantees. In *Proceedings ECML/PKDD*, pages 500–516. Springer, 2018. 15, 25, 29, 55

Michael R Berthold, Nicolas Cebron, Fabian Dill, Thomas R Gabriel, Tobias Kötter, Thorsten Meinl, Peter Ohl, Kilian Thiel, and Bernd Wiswedel. Knime-the konstanz information miner: version 2.0 and beyond. *ACM SIGKDD Explorations Newsletter*, 11(1):26–31, 2009. 20

To Thanh Binh. A multiobjective evolutionary algorithm: The study cases. In *Proceedings ACM GECCO Workshop Program*, pages 127–128, 1999. 47

To Thanh Binh and Ulrich Korn. Mobes: A multiobjective evolution strategy for constrained optimization problems. In *Proceedings ACM ICGA*, volume 25, page 27, 1997. 47

Julian Blank and Kalyanmoy Deb. pymoo: Multi-objective optimization in python. *IEEE Access*, 8:89497–89509, 2020. 49

Mario Boley and Henrik Grosskreutz. Non-redundant subgroup discovery using a closure system. In *Proceedings ECML/PKDD (1)*, pages 179–194. Springer, 2009. 14, 19

Mario Boley, Claudio Lucchese, Daniel Paurat, and Thomas Gärtner. Direct local pattern sampling by efficient two-step random procedures. In *Proceedings ACM SIGKDD*, pages 582–590, 2011. 14

Mario Boley, Sandy Moens, and Thomas Gärtner. Linear space direct pattern sampling using coupling from the past. In *Proceedings ACM SIGKDD*, pages 69–77, 2012. 14, 32

Mario Boley, Bryan R. Goldsmith, Luca M. Ghiringhelli, and Jilles Vreeken. Identifying consistent statements about numerical data with dispersion-corrected subgroup discovery. *Data Mining and Knowledge Discovery*, 31(5):1391–1418, Jun 2017. 15, 27, 29

Guillaume Bosc, Jean-François Boulicaut, Chedy Raïssi, and Mehdi Kaytoue. Anytime discovery of a diverse set of patterns with monte carlo tree search. *Data Mining and Knowledge Discovery*, 32:604–650, 2018. 15, 19, 25, 29, 62

Jean-François Boulicaut, Luc De Raedt, and Heikki Mannila, editors. *Constraint-Based Mining and Inductive Databases*. Springer, 2005. 15

Onur Boyabatli and Ihsan Sabuncuoglu. Parameter selection in genetic algorithms. *Journal of Systemics, Cybernetics and Informatics*, 4(2):78, 2004. 43

Jürgen Branke, Jurgen Branke, Kalyanmoy Deb, Kaisa Miettinen, and Roman Slowiński. *Multiobjective optimization: Interactive and evolutionary approaches*, volume 5252. Springer, 2008. 41

Jürgen Branke, Salvatore Greco, Roman Słowiński, and Piotr Zielniewicz. Interactive evolutionary multiobjective optimization using robust ordinal regression. In *Proceedings EMO*, pages 554–568. Springer, 2009. 43

Percy Williams Bridgman. *Dimensional analysis*. Yale university press, 1922. 39

Sergey Brin, Rajeev Rastogi, and Kyuseok Shim. Mining optimized gain rules for numeric attributes. *IEEE Transactions on Knowledge and Data Engineering*, 15(2):324–338, 2003. 23

Björn Bringmann and Albrecht Zimmermann. One in a million: picking the right patterns. *Knowledge and Information Systems*, 18(1):61–81, 2009. 12

Juan Carlos Fernández Caballero, Francisco José Martínez, César Hervás, and Pedro Antonio Gutiérrez. Sensitivity versus accuracy in multiclass problems using memetic pareto evolutionary neural networks. *IEEE Transactions on Neural Networks*, 21(5):750–770, 2010. 48, 84

Toon Calders, Christophe Rigotti, and Jean-François Boulicaut. A survey on condensed representations for frequent sets. *Constraint-based mining and inductive databases*, pages 64–80, 2006. 50

José-Ramón Cano, Salvador García, and Francisco Herrera. Subgroup discover in large size data sets preprocessed using stratified instance selection for increasing the presence of minority classes. *Pattern Recognition Letters*, 29(16):2156–2164, 2008. 20

Yongtao Cao, Byran J Smucker, and Timothy J Robinson. On using the hypervolume indicator to compare pareto fronts: Applications to multi-criteria optimal experimental design. *Journal of Statistical Planning and Inference*, 160:60–74, 2015. 47

Cristóbal J Carmona, Pedro González, María José del Jesus, Mercedes Navío-Acosta, and Luis Jiménez-Trevino. Evolutionary fuzzy rule extraction for subgroup discovery in a psychiatric emergency department. *Soft Computing*, 15(12):2435–2448, 2011. 50

Cristóbal J. Carmona, Pedro González, María José Del Jesus, and Francisco Herrera. NMEEF-SD: Non-dominated multiobjective evolutionary algorithm for extracting fuzzy rules in subgroup discovery. *IEEE Transactions on Fuzzy Systems*, 18:958–970, 11 2010. 51

Carolina Centeio Jorge, Martin Atzmueller, Behzad M Heravi, Jenny L Gibson, Rosaldo JF Rossetti, and Cláudio Rebelo de Sá. "want to come play with me?" outlier subgroup discovery on spatio-temporal interactions. *Expert Systems*, 2021. 21

Vira Chankong and Yacov Y Haimes. *Multiobjective decision making: theory and methodology*. Courier Dover Publications, 2008. 38

Abraham Charnes, William W Cooper, and Robert O Ferguson. Optimal estimation of executive compensation by linear programming. *Management Science*, 1(2):138–151, 1955. 39

Hwang Ching-Lai and Syed Md Masud Abu. *Multiple objective decision making, methods and applications: a state-of-the-art survey*. Springer, 1979. 39

Peter Clark and Tim Niblett. The CN2 induction algorithm. *Machine learning*, 3(4):261–283, 1989. 18

Carlos A Coello Coello and Maximino Salazar Lechuga. Mopso: A proposal for multiple objective particle swarm optimization. In *Proceedings CEC*, volume 2, pages 1051–1056. IEEE, 2002. 43

Carlos A Coello Coello and Margarita Reyes Sierra. A study of the parallelization of a coevolutionary multi-objective evolutionary algorithm. In *Proceedings MICAI*, pages 688–697. Springer, 2004. 45

Carlos A Coello Coello, Gregorio Toscano Pulido, and M Salazar Lechuga. Handling multiple objectives with particle swarm optimization. *IEEE Transactions on Evolutionary Computation*, 8(3):256–279, 2004. 43

David W Corne, Joshua D Knowles, and Martin J Oates. The pareto envelope-based selection algorithm for multiobjective optimization. In *Proceedings PPSN*, pages 839–848. Springer, 2000. 43

Bruno Crémilleux and Jean-François Boulicaut. Simplest rules characterizing classes generated by δ-free sets. In *Research and development in intelligent systems XIX*, pages 33–46. Springer, 2003. 19

Yunfei Cui, Zhiqiang Geng, Qunxiong Zhu, and Yongming Han. Multi-objective optimization methods and application in energy saving. *Energy*, 125:681–704, 2017. 48

Indraneel Das and John E Dennis. Normal-boundary intersection: A new method for generating the pareto surface in nonlinear multicriteria optimization problems. *SIAM Journal on Optimization*, 8(3):631–657, 1998. 44

Dilip Datta and Jose Rui Figueira. Some convergence-based m-ary cardinal metrics for comparing performances of multi-objective optimizers. *Computers & Operations Research*, 39(7):1754–1762, 2012. 44

Cláudio Rebelo de Sá, Wouter Duivesteijn, Carlos Soares, and Arno Knobbe. Exceptional preferences mining. In *Proceedings DS*, pages 3–18. Springer, 2016. 34

Kalyanmoy Deb. Multi-objective optimization. In *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques*, pages 403–449. Springer, 2014. 4, 38, 39

Kalyanmoy Deb and Sachin Jain. Multi-speed gearbox design using multi-objective evolutionary algorithms. *Journal of Mechanical Design*, 125(3):609–619, 2003. 43

Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002a. 42

Kalyanmoy Deb, Lothar Thiele, Marco Laumanns, and Eckart Zitzler. Scalable multi-objective optimization test problems. In *Proceedings CEC*, volume 1, pages 825–830. IEEE, 2002b. 47

Kalyanmoy Deb, Lothar Thiele, Marco Laumanns, and Eckart Zitzler. Scalable test problems for evolutionary multiobjective optimization. In *Evolutionary multiobjective optimization*, pages 105–145. Springer, 2005. 47

Kalyanmoy Deb, Shamik Chaudhuri, and Kaisa Miettinen. Towards estimating nadir objective vector using evolutionary approaches. In *Proceedings GECCO*, pages 643–650, 2006. 46

Kalyanmoy Deb, Ankur Sinha, Pekka J Korhonen, and Jyrki Wallenius. An interactive evolutionary multiobjective optimization method based on progressively approximated value functions. *IEEE Transactions on Evolutionary Computation*, 14(5):723–739, 2010. 43

Satchidananda Dehuri and Rajib Mall. Predictive and comprehensible rule discovery using a multi-objective genetic algorithm. *Knowledge-Based Systems*, 19(6):413–421, 2006. 49

María José Del Jesus, Pedro González, and Francisco Herrera. Multiobjective genetic algorithm for extracting subgroup discovery fuzzy rules. In *Proceedings IEEE MCDM*, pages 50–57, 2007a. 50

María José Del Jesus, Pedro González, Francisco Herrera, and Mikel Mesonero. Evolutionary fuzzy rule induction process for subgroup discovery: a case study in marketing. *IEEE Transactions on Fuzzy Systems*, 15(4):578–592, 2007b. 50

Junning Deng, Bo Kang, Jefrey Lijffijt, and Tijl De Bie. Explainable subgraphs with surprising densities: A subgroup discovery approach. In *Proceedings SIAM DM*, pages 586–594, 2020. 21

Francisco Domingo-Perez, Jose Luis Lazaro-Galilea, Andreas Wieser, Ernesto Martin-Gorostiza, David Salido-Monzu, and Alvaro de la Llana. Sensor placement determination for range-difference positioning using evolutionary multi-objective optimization. *Expert Systems with Applications*, 47:95–105, 2016. 48

Guozhu Dong and Jinyan Li. Efficient mining of emerging patterns: Discovering trends and differences. In *Proceedings ACM SIGKDD*, pages 43–52, 1999. 12

Lennart Downar and Wouter Duivesteijn. Exceptionally monotone models—the rank correlation model class for exceptional model mining. *Knowledge and Information Systems*, 51 (2):369–394, 2017. 34

Mădălina M Drugan and Dirk Thierens. Stochastic pareto local search: Pareto neighbourhood exploration and perturbation strategies. *Journal of Heuristics*, 18(5):727–766, 2012. 43

Xin Du, Yulong Pei, Wouter Duivesteijn, and Mykola Pechenizkiy. Exceptional spatio-temporal behavior mining through bayesian non-parametric modeling. *Data Mining and Knowledge Discovery*, 34(5):1267–1290, 2020. 34

Wouter Duivesteijn and Arno Knobbe. Exploiting false discoveries–statistical validation of patterns and quality measures in subgroup discovery. In *Proceedings IEEE ICDM*, pages 151–160, 2011. 17

Wouter Duivesteijn and Julia Thaele. Understanding where your classifier does (not) work–the scape model class for emm. In *Proceedings IEEE ICDM*, pages 809–814, 2014. 33

Wouter Duivesteijn, Arno Knobbe, Ad Feelders, and Matthijs van Leeuwen. Subgroup discovery meets bayesian networks–an exceptional model mining approach. In *Proceedings IEEE ICDM*, pages 158–167, 2010. 33

Wouter Duivesteijn, Ad Feelders, and Arno Knobbe. Different slopes for different folks: Mining for exceptional regression models with cook's distance. In *Proceedings ACM SIGKDD*, page 868–876, 2012a. 33

Wouter Duivesteijn, Eneldo Loza Mencía, Johannes Fürnkranz, and Arno Knobbe. Multi-label lego – enhancing multi-label classifiers with local patterns. In *Proceedings IDA*, page 114–125. Springer, 2012b. 33

Wouter Duivesteijn, Ad J. Feelders, and Arno Knobbe. Exceptional model mining. *Data Mining and Knowledge Discovery*, 30(1):47–98, 2016. 4, 30, 31, 33, 70

Juan J Durillo and Antonio J Nebro. jmetal: A java framework for multi-objective optimization. *Advances in Engineering Software*, 42(10):760–771, 2011. 41, 48

Saso Dzeroski, Bart Goethals, and Pance Panov, editors. *Inductive Databases and Constraint-Based Data Mining*. Springer, 2010. 15

A.E. Eiben, R. Hinterding, and Z. Michalewicz. Parameter control in evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 3(2):124–141, 1999. 43

Agoston E Eiben and Jim Smith. From evolutionary computation to the evolution of things. *Nature*, 521(7553):476–482, 2015. 4, 38

Usama M. Fayyad and Keki B. Irani. Multi-interval discretization of continuous-valued attributes for classification learning. In *Proceedings IJCAI*, pages 1022–1029, 1993. 13, 23

José C Ferreira, Carlos M Fonseca, and António Gaspar-Cunha. Methodology to select solutions from the pareto-optimal set: a comparative study. In *Proceedings GECCO*, pages 789–796, 2007. 46

Carlos M Fonseca and Peter J Fleming. Multiobjective genetic algorithms made easy: selection sharing and mating restriction. In *Proceedings GALESIA*, pages 45–52, 1995. 41, 47

Carlos M Fonseca, Peter J Fleming, et al. Genetic algorithms for multiobjective optimization: Formulation discussion and generalization. In *Proceedings ICGA*, pages 416–423. Elsevier, 1993. 42

Alex A Freitas. On rule interestingness measures. In *Research and Development in Expert Systems XV*, pages 147–158. Springer, 1999. 15

Dimas Fuente, Miguel A Vega-Rodríguez, and Carlos J Pérez. Automatic selection of a single solution from the pareto front to identify key players in social networks. *Knowledge-Based Systems*, 160:228–236, 2018. 46

Takeshi Fukuda, Yasuhido Morimoto, Shinichi Morishita, and Takeshi Tokuyama. Mining optimized association rules for numeric attributes. In *Proceedings ACM PODS*, pages 182–191, 1996a. 23

Takeshi Fukuda, Yasukiko Morimoto, Shinichi Morishita, and Takeshi Tokuyama. Data mining using two-dimensional optimized association rules: Scheme, algorithms, and visualization. *ACM SIGMOD Record*, 25(2):13–23, 1996b. 23

Maria Jose Gacto, Rafael Alcalá, and Francisco Herrera. Adaptation and application of multi-objective evolutionary algorithms for rule reduction and parameter tuning of fuzzy rule-based systems. *Soft Computing*, 13(5):419–436, 2009. 49

Arianna Gallo, Pauli Miettinen, and Heikki Mannila. Finding subgroups having several descriptions: Algorithms for redescription mining. In *Proceedings SIAM DM*, pages 334–345, 2008. 21

Dragan Gamberger and Nada Lavrac. Expert-guided subgroup discovery: Methodology and application. *Journal of Artificial Intelligence Research*, 17(1):501–527, December 2002a. 17

Dragan Gamberger and Nada Lavrac. Generating actionable knowledge by expert-guided subgroup discovery. In *Proceedings PKDD*, pages 163–175. Springer, 2002b. 17

Dragan Gamberger, Nada Lavrač, and Goran Krstačić. Active subgroup mining: a case study in coronary heart disease risk group detection. *Artificial Intelligence in Medicine*, 28(1): 27–57, 2003. 21

Dragan Gamberger, Nada Lavrač, Antonija Krstačić, and Goran Krstačić. Clinical data analysis based on iterative subgroup discovery: experiments in brain ischaemia data analysis. *Applied Intelligence*, 27(3):205–217, 2007. 21

Bernhard Ganter and Rudolf Wille. *Formal Concept Analysis: Mathematical Foundations*. Springer, 1998. 24

Salvador Garcia, Julian Luengo, Jose A. Saez, Victoria Lopez, and Francisco Herrera. A survey of discretization techniques: Taxonomy and empirical analysis in supervised learning. *IEEE Transactions on Knowledge and Data Engineering*, 25(4):734–750, 2013. 13, 23

Gemma C. Garriga, Petra Kralj, and Nada Lavrač. Closed sets for labeled data. *Journal of Machine Learning Research*, 9:559–580, 2008. 19, 25, 28

Liqiang Geng and Howard J. Hamilton. Interestingness measures for data mining: A survey. *ACM Computing Surveys*, 38(3), September 2006. 15

Bryan R Goldsmith, Mario Boley, Jilles Vreeken, Matthias Scheffler, and Luca M Ghiringhelli. Uncovering structure-property relationships of materials by subgroup discovery. *New Journal of Physics*, 19(1):013031, 2017. 21

Marc Gravel, Wilson L Price, and Caroline Gagné. Scheduling continuous casting of aluminum using a multiple objective ant colony optimization metaheuristic. *European Journal of Operational Research*, 143(1):218–229, 2002. 43

Henrik Grosskreutz. Cascaded subgroups discovery with an application to regression. In *Proceedings ECML/PKDD*, page 33. Springer, 2008. 27

Henrik Grosskreutz and Fraunhofer IAIS. Class relevant pattern mining in output-polynomial time. In *Proceedings SIAM DM*, pages 284–294, 2012. 19

Henrik Grosskreutz and Daniel Paurat. Fast and memory-efficient discovery of the top-k relevant subgroups in a reduced candidate space. In *Proceedings ECML/PKDD*, pages 533–548. Springer, 2011. 14, 19, 29

Henrik Grosskreutz and Stefan Rüping. On subgroup discovery in numerical domains. *Data Mining and Knowledge Discovery*, 19(2):210–226, 2009. 24, 29

Henrik Grosskreutz, Stefan Rüping, and Stefan Wrobel. Tight optimistic estimates for fast subgroup discovery. In *Proceedings ECML/PKDD*, pages 440–456. Springer, 2008. 14, 20

Henrik Grosskreutz, Mario Boley, and Maike Krause-Traudes. Subgroup discovery for election analysis: a case study in descriptive data mining. In *Proceedings DS*, pages 57–71. Springer, 2010. 21

Thomas Guyet, René Quiniou, and Véronique Masson. Mining relevant interval rules. *CoRR*, abs/1709.03267, 2017. 25, 29, 55

Yacov Haimes. On a bicriterion formulation of the problems of integrated system identification and system optimization. *IEEE Transactions on Systems, Man, and Cybernetics*, 1(3):296–297, 1971. 39

Wilhelmiina Hämäläinen. Statapriori: an efficient algorithm for searching statistically significant association rules. *Knowledge and Information Systems*, 23(3):373–399, 2010. 16

Wilhelmiina Hämäläinen and Geoffrey I Webb. A tutorial on statistically sound pattern discovery. *Data Mining and Knowledge Discovery*, 33(2):325–377, 2019. 17

Mohamed Hamdy, Anh-Tuan Nguyen, and Jan LM Hensen. A performance comparison of multi-objective optimization algorithms for solving nearly-zero-energy-building design problems. *Energy and Buildings*, 121:57–71, 2016. 48

Mohamed Ali Hammal, Hélène Mathian, Luc Merchez, Marc Plantevit, and Céline Robardet. Rank correlated subgroup discovery. *Journal of Intelligent Information Systems*, 53(2): 305–328, 2019. 34

Jiawei Han, Jian Pei, and Yiwen Yin. Mining frequent patterns without candidate generation. In *Proceedings ACM SIGMOD*, pages 1–12, 2000. 12, 14, 31

Caleb Harper and Mario Siller. Openag: A globally distributed network of food computing. *IEEE Pervasive Computing*, 14:24–27, 2015. 94

Zhenan He and Gary G Yen. Visualization and performance metric in many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 20(3):386–402, 2015. 48

Sumyea Helal, Jiuyong Li, Lin Liu, Esmaeil Ebrahimie, Shane Dawson, and Duncan J Murray. Identifying key factors of student academic performance by subgroup discovery. *International Journal of Data Science and Analytics*, 7(3):227–245, 2019. 21

Andrew T Hendrickson, Jason Wang, and Martin Atzmueller. Identifying exceptional descriptions of people using topic modeling and subgroup discovery. In *Proceedings ISMIS*, pages 454–462. Springer, 2018. 34

Francisco Herrera, Cristóbal J. Carmona, Pedro González, and María José Del Jesus. An overview on subgroup discovery: Foundations and applications. *Knowledge and Information Systems*, 29:495–525, 2011. 12, 15

Patrick Hoffman, Georges Grinstein, Kenneth Marx, Ivo Grosse, and Eugene Stanley. Dna visual and analytic data mining. In *Proceedings IEEE Visualization*, pages 437–441, 1997. 48

Jeffrey Horn, Nicholas Nafpliotis, and David E Goldberg. A niched pareto genetic algorithm for multiobjective optimization. In *Proceedings CEC*, pages 82–87. IEEE, 1994. 42

Simon Huband, Philip Hingston, Luigi Barone, and Lyndon While. A review of multiobjective test problems and a scalable test problem toolkit. *IEEE Transactions on Evolutionary Computation*, 10(5):477–506, 2006. 47

Amin Ibrahim, Shahryar Rahnamayan, Miguel Vargas Martin, and Kalyanmoy Deb. 3d-radvis antenna: Visualization and performance measure for many-objective optimization. *Swarm and Evolutionary Computation*, 39:157–176, 2018. 46

Dino Ienco, Albert Bifet, Indrė Žliobaitė, and Bernhard Pfahringer. Clustering based active learning for evolving data streams. In *Proceedings DS*, pages 79–93. Springer, 2013. 38

Alfred Inselberg and Bernard Dimsdale. Parallel coordinates: a tool for visualizing multi-dimensional geometry. In *Proceedings IEEE Visualization*, pages 361–378, 1990. 48

Hisao Ishibuchi and Tadahiko Murata. Multi-objective genetic local search algorithm. In *Proceedings CEC*, pages 119–124. IEEE, 1996. 42

Hisao Ishibuchi, Hiroyuki Masuda, and Yusuke Nojima. A study on performance evaluation ability of a modified inverted generational distance indicator. In *Proceedings GECCO*, pages 695–702, 2015. 44, 45, 46

Nanlin Jin, Peter Flach, Tom Wilcox, Royston Sellman, Joshua Thumim, and Arno Knobbe. Subgroup discovery in smart electricity meter data. *IEEE Transactions on Industrial Informatics*, 10(2):1327–1336, 2014. 21

Arielle Johnson, Elliot Meyerson, John Parra, Timothy Savas, Risto Miikkulainen, and Caleb Harper. Flavor-cyber-agriculture: Optimization of plant metabolites in an open-source control environment through surrogate modeling. *Plos ONE*, 14:e0213918, 2019. 94, 99, 104, 105

Donald R. Jones, Matthias Schonlau, and William J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4):455–492, 1998. 99

Branko Kavsek and Nada Lavrac. Analysis of example weighting in subgroup discovery by comparison of three algorithms on a real-life data set. In *Proceedings ECML/PKDD*, pages 64–76. Springer, 2004a. 18

Branko Kavsek and Nada Lavrac. Using subgroup discovery to analyze the UK traffic data. *Metodoloski zvezki*, 1(1):249, 2004b. 21

Branko Kavšek, Nada Lavrač, and Viktor Jovanoski. Apriori-sd: Adapting association rule learning to subgroup discovery. In *Proceedings IDA*, pages 230–241. Springer, 2003. 14, 18

Branko Kavšek, Nada Lavrac, and Ljupco Todorovski. ROC analysis of example weighting in subgroup discovery. *Algorithms*, 2:3, 2004. 18

Mehmet Kaya and Reda Alhajj. Multi-objective genetic algorithm based method for mining optimized fuzzy association rules. In *Proceedings IDEAL*, pages 758–764. Springer, 2004. 49

Mehdi Kaytoue, Sergei O. Kuznetsov, and Amedeo Napoli. Revisiting numerical pattern mining with formal concept analysis. In *Proceedings IJCAI*, pages 1342–1347, 2011. 6, 24, 25, 54, 55, 121

Willi Klösgen. Explora: A multipattern and multistrategy discovery assistant. In *Advances in Knowledge Discovery and Data Mining*, page 249–271. AAAI, 1996. 12, 27

Joshua D Knowles and David W Corne. Approximating the nondominated front using the pareto archived evolution strategy. *Evolutionary computation*, 8(2):149–172, 2000. 43

John R. Koza. Genetic programming: On the programming of computers by means of natural selection. pages 162–169. MIT Press, Cambridge, MA, USA, 1992. 42, 99

Thomas E Krak and Ad Feelders. Exceptional model mining with tree-constrained gradient ascent. In *Proceedings SIAM DM*, pages 487–495, 2015. 31

Petra Kralj, Nada Lavrac, Blaz Zupan, and Dragan Gamberger. Experimental comparison of three subgroup discovery algorithms: Analysing brain ischemia data. *Information Society*, pages 220–223, 2005. 23

Frank Kursawe. A variant of evolution strategies for vector optimization. In *Proceedings PPSN*, pages 193–197. Springer, 1990. 47

Marco Laumanns, Lothar Thiele, Kalyanmoy Deb, and Eckart Zitzler. Combining convergence and diversity in evolutionary multiobjective optimization. *Evolutionary Computation*, 10(3):263–282, 2002. 43

Nada Lavrač and Dragan Gamberger. Relevancy in constraint-based subgroup discovery. In *Constraint-based mining and inductive databases*, pages 243–266. Springer, 2006. 15, 23

Nada Lavrač, Peter Flach, and Blaz Zupan. Rule evaluation measures: A unifying view. In *Proceedings ILP*, pages 174–185. Springer, 1999. 15

Nada Lavrač, Bojan Cestnik, Dragan Gamberger, and Peter Flach. Decision support through subgroup discovery: three case studies and the lessons learned. *Machine Learning*, 57(1): 115–143, 2004. 20

Nada Lavrac, Branko Kavsek, Peter Flach, and Ljupco Todorovski. Subgroup discovery with CN2-SD. *Journal of Machine Learning Research*, 5(2):153–188, 2004. 14, 18

Dennis Leman, Ad Feelders, and Arno Knobbe. Exceptional model mining. In *Proceedings ECML/PKDD*, pages 1–16. Springer, 2008. 30, 31, 32, 33

Florian Lemmerich. Novel techniques for efficient and effective subgroup discovery, PhD thesis. 2014. 15, 20, 26, 27

Florian Lemmerich and Martin Becker. pysubgroup: Easy-to-use subgroup discovery in python. In *Proceedings ECML/PKDD*, pages 658–662. Springer, 2018. 20

Florian Lemmerich and Frank Puppe. Local models for expectation-driven subgroup discovery. In *Proceedings IEEE ICDM*, pages 360–369, 2011. 16

Florian Lemmerich, Mathias Rohlfs, and Martin Atzmueller. Fast discovery of relevant subgroup patterns. In *Proceedings FLAIRS*, 2010. 14, 19, 27, 29

Florian Lemmerich, Martin Becker, and Martin Atzmueller. Generic pattern trees for exhaustive exceptional model mining. In *Proceedings ECML/PKDD*, pages 277–292. Springer, 2012. 31

Florian Lemmerich, Martin Becker, and Frank Puppe. Difference-based estimates for generalization-aware subgroup discovery. In *Proceedings ECML/PKDD*, pages 288–303. Springer, 2013. 16

Florian Lemmerich, Martin Atzmueller, and Frank Puppe. Fast exhaustive subgroup discovery with numerical target concepts. *Data Mining and Knowledge Discovery*, 30(3):711–762, 2016a. 14, 16, 20, 27, 29, 56, 57, 60, 101

Florian Lemmerich, Martin Becker, Philipp Singer, Denis Helic, Andreas Hotho, and Markus Strohmaier. Mining subgroups with exceptional transition behavior. In *Proceedings ACM SIGKDD*, pages 965–974, 2016b. 34

Florian Lemmerich, Christoph Kiefer, Benedikt Langenberg, Jeffry Cacho Aboukhalil, and Axel Mayer. Mining exceptional mediation models. In *Proceedings ISMIS*, pages 318–328. Springer, 2020. 34

Jiuyong Li, Jixue Liu, Hannu Toivonen, Kenji Satou, Youqiang Sun, and Bingyu Sun. Discovering statistically non-redundant subgroups. *Knowledge-Based Systems*, 67:315–327, 2014. 15, 17, 19, 28

Miqing Li and Xin Yao. Quality evaluation of solution sets in multiobjective optimisation: A survey. *ACM Computing Surveys*, 52(2), 2019. 44, 68, 69

Jefrey Lijffijt, Bo Kang, Wouter Duivesteijn, Kai Puolamaki, Emilia Oikarinen, and Tijl De Bie. Subjectively interesting subgroup discovery on real-valued targets. In *Proceedings IEEE ICDE*, pages 1352–1355, 2018. 27, 34

Chunming Liu, Xin Xu, and Dewen Hu. Multiobjective reinforcement learning: A comprehensive overview. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 45(3): 385–398, 2014. 38

José M Luna, José Raúl Romero, Cristóbal Romero, and Sebastián Ventura. Discovering subgroups by means of genetic programming. In *Proceedings EuroGP*, pages 121–132. Springer, 2013. 14

Jose Maria Luna, Mykola Pechenizkiy, and Sebastian Ventura. Mining exceptional relationships with grammar-guided genetic programming. *Knowledge and Information Systems*, 47(3):571–594, 2016. 34

José María Luna, Mykola Pechenizkiy, Wouter Duivesteijn, and Sebastián Ventura. Exceptional in so many ways—discovering descriptors that display exceptional behavior on contrasting scenarios. *IEEE Access*, 8:200982–200994, 2020. 34

Michael Mampaey, Siegfried Nijssen, Ad Feelders, and Arno Knobbe. Efficient algorithms for finding richer subgroup descriptions in numeric and nominal data. In *Proceedings IEEE ICDM*, pages 499–508, 2012. 24

Michael Mampaey, Siegfried Nijssen, Ad Feelders, Rob Konijn, and Arno Knobbe. Efficient algorithms for finding optimal binary features in numeric and nominal labeled data. *Knowledge and Information Systems*, 42(2):465–492, 2015. 14, 24

Diana Martín, Alejandro Rosete, Jesús Alcalá-Fdez, and Francisco Herrera. A multi-objective evolutionary algorithm for mining quantitative association rules. In *Proceedings ISDA*, pages 1397–1402. IEEE, 2011. 50

Jacinto Mata, José-Luis Alvarez, and José-Cristobal Riquelme. An evolutionary algorithm to discover numeric association rules. In *Proceedings ACM SAC*, pages 590–594, 2002. 23

Romain Mathonat, Diana Nurbakova, Jean-François Boulicaut, and Mehdi Kaytoue. Seqscout: Using a bandit model to discover interesting subgroups in labeled sequences. In *Proceedings IEEE DSAA*, pages 81–90, 2019. 15, 21

Romain Mathonat, Diana Nurbakova, Jean-François Boulicaut, and Mehdi Kaytoue. Anytime mining of sequential discriminative patterns in labeled sequences. *Knowledge and Information Systems*, 63(2):439–476, 2021. 15, 21

M. D. McKay, R. J. Beckman, and W. J. Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21(2):239–245, 1979. 101

Patrick R McMullen. An ant colony optimization approach to addressing a jit sequencing problem with multiple objectives. *Artificial Intelligence in Engineering*, 15(3):309–317, 2001. 43

Marvin Meeng and Arno Knobbe. Flexible enrichment with cortana–software demo. In *Proceedings BeneLearn*, pages 117–119, 2011. 20

Marvin Meeng and Arno Knobbe. For real: a thorough look at numeric attributes in subgroup discovery. *Data Mining and Knowledge Discovery*, 35(1):158–212, 2021. 25

Marvin Meeng, Wouter Duivesteijn, and Arno Knobbe. Rocsearch—an roc-guided search strategy for subgroup discovery. In *Proceedings SIAM DM*, pages 704–712, 2014. 25, 29

Marvin Meeng, Harm de Vries, Peter Flach, Siegfried Nijssen, and Arno Knobbe. Uni-and multivariate probability density models for numeric subgroup discovery. *Intelligent Data Analysis*, 24(6):1403–1439, 2020. 28

Achille Messac. Physical programming-effective optimization for computational design. *AIAA Journal*, 34(1):149–158, 1996. 39

Achille Messac, Amir Ismail-Yahaya, and Christopher A Mattson. The normalized normal constraint method for generating the pareto frontier. *Structural and Multidisciplinary Optimization*, 25(2):86–98, 2003. 44

Renée J Miller and Yuping Yang. Association rules over interval data. *ACM SIGMOD Record*, 26(2):452–461, 1997. 23

Alexandre Millot, Rémy Cazabet, and Jean-François Boulicaut. Optimal subgroup discovery in purely numerical data. In *Proceedings PAKDD*, pages 112–124. Springer, 2020a. 7, 55, 59

Alexandre Millot, Rémy Cazabet, and Jean-François Boulicaut. Découverte d'un sous-groupe optimal dans des données purement numériques. In *Extraction et Gestion des Connaissances (EGC)*, pages 25–36, 2020b. 7, 55

Alexandre Millot, Romain Mathonat, Rémy Cazabet, and Jean-François Boulicaut. Actionable subgroup discovery and urban farm optimization. In *Proceedings IDA*, pages 339–351. Springer, 2020c. 8, 99

Alexandre Millot, Rémy Cazabet, and Jean-François Boulicaut. Exceptional model mining meets multi-objective optimization. In *Proceedings SIAM DM*, pages 378–386, 2021. 8, 66

Seyedali Mirjalili. Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. *Neural Computing and Applications*, 27(4):1053–1073, 2016. 43

Sandy Moens and Mario Boley. Instant exceptional model mining using weighted controlled pattern sampling. In *Proceedings IDA*, pages 203–214. Springer, 2014. 32

Katherine Moreland and Klaus Truemper. Discretization of target attributes for subgroup discovery. In *Proceedings MLDM*, pages 44–52. Springer, 2009. 13, 26

Katharina Morik, Jean-François Boulicaut, and Arno Siebes, editors. *Local Pattern Detection, International Seminar, Dagstuhl Castle, Germany, April 12-16, 2004, Revised Selected Papers*. Springer, 2005. 12

Shinichi Morishita and Jun Sese. Transversing itemset lattices with statistical metric pruning. In *Proceedings ACM PODS*, pages 226–236, 2000. 19, 26

Sanaz Mostaghim and Jürgen Teich. Strategies for finding good local guides in multi-objective particle swarm optimization (mopso). In *Proceedings SIS*, pages 26–33. IEEE, 2003. 43

Marianne Mueller, Rómer Rosales, Harald Steck, Sriram Krishnan, Bharat Rao, and Stefan Kramer. Subgroup discovery for test selection: a novel approach and its application to breast cancer diagnosis. In *Proceedings IDA*, pages 119–130. Springer, 2009. 21

Hoang Vu Nguyen and Jilles Vreeken. Flexibly mining better subgroups. In *Proceedings SIAM DM*, pages 585–593, 2016. 24

Aurelice B Oliveira, Carlos FH Moura, Enéas Gomes-Filho, Claudia A Marco, Laurent Urban, and Maria Raquel A Miranda. The impact of organic farming on quality of tomatoes is associated to increased oxidative stress during fruit development. *PLoS One*, 8(2):e56354, 2013. 125

Francisco Padillo, José María Luna, and Sebastián Ventura. Subgroup discovery on big data: Pruning the search space on exhaustive search algorithms. In *Proceedings IEEE Big Data*, pages 1814–1823, 2016. 20

Hari Mohan Pandey, Ankit Chaudhary, and Deepti Mehrotra. A comparative review of approaches to prevent premature convergence in ga. *Applied Soft Computing*, 24:1047–1077, 2014. 43

Vilfredo Pareto. Manuale di economica politica, societa editrice libraria. *Manual of political economy*, 1971, 1906. 40

Gregory Piatetsky-Shapiro, U Fayyad, and Padraic Smith. From data mining to knowledge discovery: An overview. *Advances in Knowledge Discovery and Data Mining*, 1:35, 1996. 12

Barbara FI Pieters, Arno Knobbe, and Sašo Dzeroski. Subgroup discovery in ranked data, with an application to gene set enrichment. In *Proceedings PL workshop at ECML/PKDD*, volume 10, pages 1–18, 2010. 13, 21, 26

Hugo M Proença, Peter Grünwald, Thomas Bäck, and Matthijs van Leeuwen. Discovering outstanding subgroup lists for numeric targets using mdl. *Proceedings ECML/PKDD*, pages 19–35, 2021. 14, 27

Andy Pryke, Sanaz Mostaghim, and Alireza Nazemi. Heatmap visualization of population based multi objective algorithms. In *Proceedings EMO*, pages 361–375. Springer, 2007. 48

Friedrich Pukelsheim. *Optimal design of experiments*. SIAM, 2006. 38

Gade Pandu Rangaiah. *Multi-objective optimization: techniques and applications in chemical engineering*, volume 5. world scientific, 2016. 48

Alan Reynolds and Beatriz de la Iglesia. Rule induction using multi-objective metaheuristics: Encouraging rule diversity. In *Proceedings IEEE IJCNN*, pages 3343–3350, 2006. 49

Luis Miguel Rios and Nikolaos V. Sahinidis. Derivative-free optimization: a review of algorithms and comparison of software implementations. *Journal of Global Optimization*, 56 (3):1247–1293, 2013. 99

Cristóbal Romero, Pedro González, Sebastián Ventura, María José Del Jesús, and Francisco Herrera. Evolutionary algorithms for subgroup discovery in e-learning: A practical application using moodle data. *Expert Systems with Applications*, 36(2):1632–1644, 2009. 51

Ulrich Ruckert, Lothar Richter, and Stefan Kramer. Quantitative association rules based on half-spaces: An optimization approach. In *Proceedings IEEE ICDM*, pages 507–510, 2004. 23

Ansaf Salleb-Aouissi, Christel Vrain, and Cyril Nortet. Quantminer: A genetic algorithm for mining quantitative association rules. In *Proceedings IJCAI*, pages 1035–1040, 2007. 23

J David Schaffer. Multiple objective optimization with vector evaluated genetic algorithms. In *Proceedings ICGAA*. Lawrence Erlbaum Associates. Inc., Publishers, 1985. 42

Martin Scholz. Sampling-based sequential subgroup mining. In *Proceedings ACM SIGKDD*, pages 265–274, 2005. 18

Jason Ramon Schott. *Fault tolerant design using single and multicriteria genetic algorithm optimization*. PhD thesis, MIT, 1995. 45

Oliver Schutze, Xavier Esquivel, Adriana Lara, and Carlos A. Coello Coello. Using the averaged hausdorff distance as a performance measure in evolutionary multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 16(4):504–522, 2012. 44, 45, 46, 68, 69

Burr Settles. Active learning. *Synthesis lectures on artificial intelligence and machine learning*, 6(1):1–114, 2012. 38

Chuan Shi, Xiangnan Kong, Philip S Yu, and Bai Wang. Multi-objective multi-label classification. In *Proceedings SIAM DM*, pages 355–366, 2012. 48, 84

Ali Shirazi, Behzad Najafi, Mehdi Aminyavari, Fabio Rinaldi, and Robert A Taylor. Thermal–economic–environmental analysis and multi-objective optimization of an ice thermal energy storage system for gas turbine cycle inlet air cooling. *Energy*, 69:212–226, 2014. 48

Arno Siebes. Data surveying: Foundations of an inductive query language. In *Proceedings ACM KDD*, pages 269–274, 1995. 12

Samuel Silvey. *Optimal design: an introduction to the theory for parameter estimation*, volume 1. Springer, 2013. 38

Arnaud Soulet, Bruno Crémilleux, and François Rioult. Condensed representation of emerging patterns. In *Proceedings PAKDD*, pages 127–132. Springer, 2004. 19

Arnaud Soulet, Chedy Raïssi, Marc Plantevit, and Bruno Crémilleux. Mining dominant patterns in the sky. In *Proceedings IEEE ICDM*, pages 655–664, 2011. 50, 111

Ramakrishnan Srikant and Rakesh Agrawal. Mining quantitative association rules in large relational tables. In *Proceedings ACM SIGMOD*, pages 1–12, 1996. 23, 26

M. Srinivas and L.M. Patnaik. Adaptive probabilities of crossover and mutation in genetic algorithms. *IEEE Transactions on Systems, Man, and Cybernetics*, 24(4):656–667, 1994. 43

Sujatha Srinivasan and Sivakumar Ramakrishnan. Evolutionary multi objective optimization for rule mining: A review. *Artificial Intelligence Review*, 36:205–248, 10 2011. 49

Wolfram Stadler. Fundamentals of multicriteria optimization. In *Multicriteria Optimization in Engineering and in the Sciences*, pages 1–25. Springer, 1988. 39

Trevor Stephens. gplearn. `https://github.com/trevorstephens/gplearn`, 2013. 105

Eric Sternberg and Martin Atzmueller. Knowledge-based mining of exceptional patterns in logistics data: approaches and experiences in an industry 4.0 context. In *Proceedings ISMIS*, pages 67–77. Springer, 2018. 21

Balram Suman and Prabhat Kumar. A survey of simulated annealing as a tool for single and multiobjective optimization. *Journal of the Operational Research Society*, 57(10):1143–1160, 2006. 44

Benguluri Surekha, Lalith K Kaushik, Abhishek K Panduy, Pandu R Vundavilli, and Mahesh B Parappagoudar. Multi-objective optimization of green sand mould system using evolutionary algorithms. *International Journal of Advanced Manufacturing Technology*, 58 (1):9–17, 2012. 48

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018. 38

Csaba Szepesvári. Algorithms for reinforcement learning. *Synthesis lectures on artificial intelligence and machine learning*, 4(1):1–103, 2010. 38

Ryoji Tanabe and Hisao Ishibuchi. An easy-to-use real-world multi-objective optimization problem suite. *Applied Soft Computing*, 89:106078, 2020. 47

Ye Tian, Xingyi Zhang, Ran Cheng, and Yaochu Jin. A multi-objective evolutionary algorithm based on an enhanced inverted generational distance metric. In *Proceedings CEC*, pages 5222–5229. IEEE, 2016. 46

Ye Tian, Ran Cheng, Xingyi Zhang, and Yaochu Jin. Platemo: A matlab platform for evolutionary multi-objective optimization. *IEEE Computational Intelligence Magazine*, 12 (4):73–87, 2017. 49

Aimo Törn and Antanas Zilinskas. Global optimization. 1989. 38

Tea Tušar and Bogdan Filipič. Visualization of pareto front approximations in evolutionary multiobjective optimization: A critical review and the prosection method. *IEEE Transactions on Evolutionary Computation*, 19(2):225–245, 2014. 48

Gwo-Hshiung Tzeng and Jih-Jeng Huang. *Multiple attribute decision making: methods and applications*. CRC press, 2011. 38

Willy Ugarte, Patrice Boizumault, Bruno Crémilleux, Alban Lepailleur, Samir Loudni, Marc Plantevit, Chedy Raïssi, and Arnaud Soulet. Skypattern mining: From pattern condensed representations to dynamic constraint satisfaction problems. *Artificial Intelligence*, 244: 48–69, 2017. 50

Matthijs Van Leeuwen and Arno Knobbe. Diverse subgroup set discovery. *Data Mining and Knowledge Discovery*, 25(2):208–242, 2012. 18

Matthijs Van Leeuwen and Antti Ukkonen. Discovering skylines of subgroup sets. In *Proceedings ECML/PKDD*, pages 272–287. Springer, 2013. 14, 19, 51, 112

David A Van Veldhuizen. Multiobjective evolutionary algorithms: classifications, analyses, and new innovations. Technical report, AFIT - Wright-Patterson AFB, 1999. 45

David A Van Veldhuizen and Gary B Lamont. Evolutionary computation and convergence to a pareto front. In *Proceedings IEEE GP*, pages 221–228, 1998. 42, 45

Vandana Venkat, Sheldon H Jacobson, and James A Stori. A post-optimality analysis algorithm for multi-objective optimization. *Computational Optimization and Applications*, 28 (3):357–372, 2004. 46

Thomas L Vincent and Walter Jervis Grantham. Optimality in parametric systems(book). *New York, Wiley-Interscience, 1981. 257 p*, 1981. 46

David J Walker, RichardM Everson, and Jonathan E Fieldsend. Visualizing mutually nondominating solution sets in many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 17(2):165–184, 2012. 48

John Wang. *Encyclopedia of data warehousing and mining*. IGI Global, 2005. 19

Weijia Wang and Michele Sebag. Multi-objective monte-carlo tree search. In *Proceedings PMLR ACML*, pages 507–522, 2012. 38

Geoffrey I. Webb. Discovering associations with numeric variables. In *Proceedings ACM SIGKDD*, page 383–388, 2001. 26

Ian H Witten and Eibe Frank. Data mining: practical machine learning tools and techniques with java implementations. *ACM SIGMOD Record*, 31(1):76–77, 2002. 20

Renata Wojciechowska, Olga Długosz-Grochowska, Anna Kołton, and Marek Żupnik. Effects of led supplemental lighting on yield and some quality parameters of lamb's lettuce grown in two winter cycles. *Scientia Horticulturae*, 187:80–86, 2015. 94

Stefan Wrobel. An algorithm for multi-relational discovery of subgroups. In *Proceedings PKDD*, pages 78–87. Springer, 1997. 12

Jin Wu and Shapour Azarm. Metrics for quality assessment of a multiobjective design optimization solution set. *Journal of Mechanical Design*, 123(1):18–25, 2001. 45

Feng Xue, Arthur C Sanderson, and Robert J Graves. Pareto-based multi-objective differential evolution. In *Proceeding CEC*, volume 2, pages 862–869. IEEE, 2003. 43

Po-Lung Yu. Cone convexity, cone extreme points, and nondominated solutions in decision problems with multiobjectives. *Journal of Optimization Theory and Applications*, 14(3): 319–377, 1974. 39

Lofti Zadeh. Optimality and non-scalar-valued performance criteria. *IEEE Transactions on Automatic Control*, 8(1):59–60, 1963. 39

Milan Zeleny. Compromise programming. *Multiple criteria decision making*, 1973. 46

Milan Zeleny. *Multiple criteria decision making Kyoto 1975*, volume 123. Springer, 2012. 38

Hong Zhang, Balaji Padmanabhan, and Alexander Tuzhilin. On the discovery of significant statistical quantitative rules. In *Proceedings ACM SIGKDD*, pages 374–383, 2004. 16

Zhaohui Zhang, Yuchang Lu, and Bo Zhang. An effective partitioning-combining algorithm for discovering quantitative association rules. In *Proceedings PAKDD*. Springer, 1997. 23

Aimin Zhou, Bo-Yang Qu, Hui Li, Shi-Zheng Zhao, Ponnuthurai Nagaratnam Suganthan, and Qingfu Zhang. Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation*, 1(1):32 – 49, 2011. 41

Albrecht Zimmermann and Luc De Raedt. Cluster-grouping: from subgroup discovery to clustering. *Machine Learning*, 77(1):125–159, 2009. 14, 28

E. Zitzler and L. Thiele. Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *IEEE Transactions on Evolutionary Computation*, 3(4): 257–271, 1999. 76

Eckart Zitzler and Lothar Thiele. Multiobjective optimization using evolutionary algorithms—a comparative case study. In *Proceedings PPSN*, pages 292–301. Springer, 1998. 44, 46

Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation*, 8(2):173–195, 2000. 41, 45, 47

Eckart Zitzler, Marco Laumanns, and Lothar Thiele. SPEA2: Improving the strength pareto evolutionary algorithm. *TIK-report*, 103, 2001. 43

Eckart Zitzler, Joshua Knowles, and Lothar Thiele. Quality assessment of pareto set approximations. *Multiobjective Optimization*, pages 373–404, 2008. 47

FOLIO ADMINISTRATIF

THESE DE L'UNIVERSITE DE LYON OPEREE AU SEIN DE L'INSA LYON

NOM : MILLOT
(avec précision du nom de jeune fille, le cas échéant)

DATE de SOUTENANCE : 04/10/2021

Prénoms : Alexandre, Rémi, Georges

TITRE : Exceptional Model Mining meets Multi-Objective Optimization: Application to Plant Growth Recipes in Controlled Environments

NATURE : Doctorat

Numéro d'ordre : 2021LYSEI056

Ecole doctorale : InfoMaths (ED 512)

Spécialité : Informatique

RESUME : Dans la société actuelle, l'information devient de plus en plus pervasive. Avec l'avènement de l'ère du numérique, collecter et stocker ces quantités presque infinies d'informations devient de plus en plus accessible. Dans ce contexte, la conception de méthodes de découverte de motifs permettant la découverte semi-automatique d'informations pertinentes ou de connaissances est cruciale. Nous considérons des données mettant en jeu un ensemble d'attributs descriptifs, avec un ou plusieurs de ces attributs qui peut (peuvent) être considéré(s) comme variable(s) cible(s). Quand on a un seul attribut cible, la découverte de sous-groupes vise à découvrir des sous-ensembles d'objets -- des sous-groupes -- dont la distribution de l'étiquette cible dévie significativement de celle de l'ensemble des données. La fouille de modèles exceptionnels est une généralisation de la découverte de sous-groupes. C'est un cadre récent permettant la découverte de déviations locales significatives dans des interactions complexes entre plusieurs variables cibles. Dans un monde où tout doit être optimisé, les méthodes d'optimisation multi-objectifs, qui trouvent les compromis optimaux entre plusieurs variables concurrentes, sont essentielles. Bien que ces différents domaines de recherche possèdent une littérature riche, leur fertilisation croisée n'a été que peu étudiée.
Avec la disponibilité de données collectées sur un processus d'intérêt, nous nous intéressons à la conception de méthodes permettant la découverte de valeurs de paramètres pertinentes pour son optimisation. Notre première contribution est OSMIND, un algorithme de découverte de sous-groupes qui retourne un motif optimal dans des données purement numériques. OSMIND exploite des techniques avancées de réduction de l'espace de recherche garantissant l'optimalité de la découverte. Notre seconde contribution consiste en un framework itératif générique qui met à profit l'exploitabilité de la découverte de sous-groupes pour résoudre des problèmes d'optimisation. Notre troisième et principale contribution est la fouille de frontières de Pareto exceptionnelles, une nouvelle classe de modèles pour la fouille de modèles exceptionnels, qui implique une fertilisation croisée entre la découverte de motifs et l'optimisation multi-objectifs. La pertinence de chacune de nos contributions a été confirmée à travers des études empiriques approfondies. Nos méthodes sont génériques et peuvent être utilisées dans de nombreux domaines d'application.
Pour évaluer l'exploitabilité de nos contributions en situation réelle, nous considérons le problème d'optimisation de recettes de pousse de plantes en environnements contrôlés tels que les fermes urbaines, le scénario d'application qui a motivé nos travaux. Améliorer la pousse des plantes est un problème intrinsèquement multi-objectifs. Nous souhaitons appliquer nos méthodes de découverte de motifs pour découvrir les valeurs de paramètres menant à une pousse optimisée. En effet, découvrir ces réglages optimaux pourrait avoir des répercussions importantes sur la rentabilité des fermes urbaines. À partir de données synthétiques et réelles, nous démontrons que nos méthodes permettent la découverte de valeurs de paramètres optimisant le compromis rendement/coûts de recettes de pousses.
MOTS-CLÉS : Découverte de Sous-Groupes, Fouille de Modèles Exceptionnels, Optimisation Multi-objectifs, Fermes Urbaines, Recettes de Pousse de Plantes

Laboratoire (s) de recherche : Laboratoire d'InfoRmatique en Image et Systèmes d'information (LIRIS)

Directeur de thèse:
Jean-François Boulicaut (Professeur des Universités, INSA de Lyon)
Rémy Cazabet (Maître de Conférences, Université Claude Bernard Lyon 1)

Président de jury :

Composition du jury :
Bruno Crémilleux (Professeur des Universités, Université de Caen) - Rapporteur
Dino Ienco (Chargé de recherche HDR, INRAE Montpellier) - Rapporteur
Bart Goethals (Professeur des Universités, Université d'Anvers) - Examinateur
Thomas Guyet (Maître de Conférences HDR, AgroCampus Ouest) - Examinateur
Céline Robardet (Professeure des Universités, INSA-Lyon) - Examinatrice
Céline Rouveirol (Professeure des Universités, Université Paris 13) - Examinatrice