Exceptional Model Mining to support Multi-objective Optimization

Alexandre Millot, Rémy Cazabet, Jean-François Boulicaut

Exceptional Model Mining to support Multi-objective Optimization

Alexandre Millot Rémy Cazabet
Jean-François Boulicaut

November 23, 2022

Abstract

Given numerical data about objects defined by descriptive and target attributes, we investigate the discovery of interesting conjunctions of descriptive attribute range of values that are associated to optimized target values. This can be useful across many application domains where one wants to perform Multi-Objective Optimization: the goal is to find the best compromise between the competing objectives that correspond to the different targets and one expects to learn about the optimal values of the descriptive attributes. For this purpose, we design Exceptional Model Mining instances: we look for subsets of objects – subgroups – whose models deviate significantly from the same models fitted on the whole dataset. A first method, called Exceptional Pareto Front Deviation Mining (EPFDM), characterizes the differences between the Pareto front computed on the original data and the Pareto front computed after removing a subgroup of objects. We discuss in detail the design of a generic quality measure for EPFDM and we provide comprehensive empirical results. 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. Beside empirical studies that consider both the qualitative and quantitative aspects of EPFDM and EPFAM, we present a use-case on plant growth recipe optimization in controlled environments, a timely challenge for a smarter agriculture. Exceptional Model Mining Multi-objective Optimization Pareto Subgroup Discovery Pattern Mining Plant Growth Optimization

1 Introduction

When studying a given process, it is common to collect data about some of its important parameters, say descriptive attribute values, but also to have target attributes that quantify, e.g., a quality or a cost measure. When all the attributes are numerical, we can support the discovery of interesting conjunctions of descriptive attribute range of values that are associated to optimized target values like, e.g., a good trade-off between quality and cost. This can be useful

across many application domains where one wants to perform Multi-Objective Optimization (MOO).

For example, the application scenario at the heart of this research is the design of better plant growth recipes in controlled environments. Nowadays, conventional farming methods have to face many tough challenges like, e.g., 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. In such farms, plants grow in controlled environments according to recipes that specify the different growth stages and instructions regarding many parameters (e.g., temperature, humidity, CO_2, light). Throughout the whole process, automated systems keep track of such parameters and, at the end, values can be assigned to different objective variables (e.g., yield, taste, energy cost) for each recipe. We can exploit recipe data to optimize the growth process and it is intrinsically a MOO problem. This growth recipe optimization case is an example of a difficult task. The underlying model of the objective functions is unknown and new experiments can not easily be generated due to time and cost constraints. In other terms, typical MOO algorithms – that require a model of the objective functions and generating a large number of points at each iteration – cannot be used. There is therefore a need for methods that can a data-driven discovery of relevant and exploitable information in such MOO problems.

To support our data mining approaches to MOO, we decided to investigate Exceptional Model Mining (EMM) instances. Hereafter, the descriptive attributes are numerical or categorical ones and we have numerical target attributes. EMM has been introduced over 10 years ago (Leman et al., 2008). It is a generalization of subgroup discovery (Klösgen, 1996; Herrera et al., 2011). In subgroup discovery, given labeled data, we look for subsets of objects – subgroups – defined by interesting descriptions or patterns according to a quality measure computed on a unique target variable. The measure has to capture discrepancies between the target variable distribution in the subgroup and its distribution in the overall dataset. The greater and more significant the difference, the more interesting the subgroup is deemed. Within a typical EMM setting for a given class of models, we look for subgroups whose models deviate significantly from the same models fitted on the entire dataset. Once a type of data and a language of patterns have been chosen, each new EMM instance and thus type of targeted model requires (1) suitable quality measures to assess the subgroup interestingness but also, (2) algorithms that can explore the search space to identify good models. 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.

MOO is a sub-field of Multi-criteria Decision Making that is focused on find-

[1] `https://aerofarms.com/`, `https://infarm.com/`, `https://boweryfarming.com/`. Accessed on 13/09/2021.

ing 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. Pareto optimization (Deb, 2014; Zhou et al., 2011) relies on the dominance between solutions of the objective space. A solution is said to be non-dominated – or Pareto optimal – if it is impossible to improve an objective without degrading another. The set of Pareto optimal solutions is known as the Pareto front. The result of a MOO algorithm then involves not one, but a set of solutions – the Pareto front.

In our previous work (Millot et al., 2021), we introduced a first approach to EMM in a MOO context called Exceptional Pareto Front Mining (EPFM). It is based on the discovery of exceptional Pareto front deviations – now called Exceptional Pareto Front Deviation Mining (EPFDM). EPFDM identifies and characterizes the differences between the Pareto front computed on the original dataset and the Pareto front computed after removing a subgroup of objects. It can be used as an exploratory analysis tool to discover interesting pieces of knowledge about MOO problems such as (i) subspaces of the current Pareto front where data might be missing, (ii) subsets of better or worse solutions of the Pareto front with an explicit and concise description in the attribute description space, (iii) anomalous parts of the Pareto front. Suitable measures to estimate the deviations and a first generic quality measure designed for EPFM have been proposed. However, considering only distance-based Pareto front deviations limits the actionability of discovered subgroups.

We build on this previous work and significantly extend it, by further investigating the cross-fertilization between EMM and MOO. We design a generic model class for EPFM. Beside distance-based measures, we also discuss the added value of a new volume-based measure for EPFDM that enables the discovery of interesting deviation models. We also detail the conceptual basis of our generic quality measure, and introduce different ways to make it more robust. While EPFDM can be used as an exploratory tool to discover interesting knowledge regarding MOO problems, it cannot be consistently exploited to generate new, 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 concepts related to pattern relevancy for EPFDM can be easily reused or adapted for EPFAM.

The added value of both EPFDM and EPFAM is investigated by means of quantitative and qualitative empirical studies. Most of the qualitative empirical evaluations with EPFDM are novel, significantly extending the analysis in (Millot et al., 2021). We also describe in detail a promising EPFM application

scenario for the optimization of plant growth recipes in controlled environments. We introduce a realistic growth recipe simulator and show how (i) EPFDM can be used as an exploratory data analysis tool to discover interesting pieces of knowledge, such as a subspace of the objective space where the yield/cost trade-off is worse than in its complement, (ii) EPFAM provides actionable information through its subgroup descriptions and enables the design of new, better growth recipes. Our data and code are made available at `https://bit.ly/3EwnWUi`.

The remaining of the paper is organized as follows. Section 2 formalizes our mining task. We review the literature in Section 3. In Section 4, we detail our contributions to EPFDM. We then introduce our contribution to EPFAM in Section 5. Section 6 introduces a generic quality measure for EPFM. In Section 7, we present an in-depth empirical evaluation of both EPFDM and EPFAM. Section 8 introduces our application scenario for EPFM in the field of smart agriculture. In Section 9, we discuss the limitations of our work. Finally, Section 10 concludes.

2 Preliminaries

2.1 Multi-Objective Optimization

The process to be optimized is known thanks to a collected dataset.

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, M contains real-valued and categorical attributes – the domain of any attribute $m \in M$ is denoted by $Dom(m)$ – while the domain of each target $t \in T$ is a finite ordered set. Indeed, the targets can represent continuous or discrete variables, but the finiteness of the data restricts their domain to a finite ordered set. In this work, when numerical attributes are present in a dataset, we consider that a discretization method has to be applied on them as a pre-processing step.

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. Given a dataset (G, M, T), the vectors of objective function values 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 optimizes all objectives simultaneously. When no order or relevance can be defined a priori on the different objectives, a Pareto optimization method is required. It is based on the dominance between solutions of the objective space.

A vector $a = (a_1, \ldots, a_n)^T$ is said to dominate a vector $b = (b_1, \ldots, b_n)^T$, denoted $a \prec b$ if and only if $\forall i \in \{1, \ldots, n\}, u_i \leq v_i\ and\ u \neq v$. For example, in Figure 1, $A \prec B$ (i.e., object A dominates object B) since $f_1(A) < f_1(B)$ and

$f_2(A) < f_2(B)$.
A non-dominated solution is called Pareto optimal.

A solution x is called Pareto optimal if and only if $\nexists y \in M$ such that $F(y) \prec F(x)$ The set of all Pareto optimal solutions is called the 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 (Zitzler et al., 2000; Deb et al., 2005; Huband et al., 2006). The true Pareto front (i.e., ground truth) of these functions is usually known – the quality of a solution set returned by any MOO algorithm can therefore easily be compared to the ground truth – and they are designed such that Pareto front approximation by algorithms is difficult. To illustrate our work and some of 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 objective 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 and we retrieve the true Pareto front of the function from jMetal[2]. `Fonseca` is the name of the corresponding dataset. Table 1 provides a toy dataset – with discretized numerical attributes using equal-width with 5 bins – that is a subset of the 5000 random objects of `Fonseca`.

Table 1: Toy dataset related to the Fonseca-Fleming function.

	x_1	x_2	x_3	f_1	f_2
g_1	[-4,-2.4]	[2.4,4]	[-0.8,0.8]	0.99	0.99
g_2	[-2.4,-0.8]	[-0.8,0.8]	[-2.4,-0.8]	0.99	0.89
g_3	[-0.8,0.8]	[-2.4,-0.8]	[0.8,2.4]	0.99	0.99
g_4	[-0.8,0.8]	[-0.8,0.8]	[-2.4,-0.8]	0.90	0.50
g_5	[-0.8,0.8]	[-0.8,0.8]	[-2.4,-0.8]	0.99	0.60

Figure 1 depicts the Pareto front (i.e., the non-dominated solutions) of the Fonseca-Fleming function. Very few points lie close to the Pareto front. 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 not optimal to solve MOO problems.

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

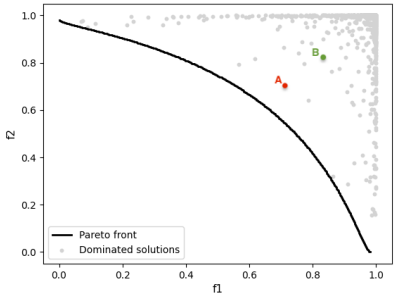

Figure 1: True Pareto front of the Fonseca-Fleming function (in black) and objects of the `Fonseca` dataset (in grey). A and B are two random objects in the objective space.

2.2 Exceptional Model Mining

EMM is a generalization of subgroup discovery that can handle more than one target attribute by using model classes. A subgroup p can be described either by its intent – the description of the subgroup in terms of attribute values – or by its extent – the coverage of the subgroup in the dataset. The intent of a subgroup p is given by $p_d = \langle \varphi_1, \ldots, \varphi_n \rangle$ where each φ_i is a restriction on the domain value of $m_i \in M$. A restriction for a nominal attribute m_i is given by $m_i = v$ with $v \in Dom(m_i)$. The intent p_d of subgroup p covers the set of objects denoted $ext(p_d) \subseteq G$. For example, given the toy dataset in Table 1, we could find a subgroup whose intent is $\langle x_1 = [-0.8, 0.8],\ x_2 = [-0.8, 0.8],\ x_3 = [-2.4, -0.8] \rangle$ and whose extent is $\{g_4, g_5\}$.

In subgroup discovery, we have only one target. The quality of a subgroup is usually defined as the difference between the distribution of the target variable in the subgroup and its distribution over the entire dataset. Since important discrepancies can easily be achieved with small subsets of objects, a factor that takes into account the size of the subgroup can be used as well (see, e.g., the popular Weighted Relative Accuracy measure (Lavrac et al., 2004)).

Exceptional Model Mining enables for two or more target variables depending on the chosen model class. In the standard EMM setting, the interestingness of a subgroup p 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 for which subset 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 scenario. Duivesteijn et al. (2016) show that there is not always a clear-cut best solution.

From an algorithmic perspective, subgroups are explored by progressively

specializing their intents, from general to more specific descriptions. At each stage, a specialization operator is applied to create more complex subgroups by addition of a restriction on an attribute.

3 Related Work

We investigate possibilities for cross-fertilization between EMM and MOO thanks to the computation of Pareto fronts on objects of the dataset. To the best of our knowledge, we were the first to consider EMM based on this concept (Millot et al., 2021). However, closely related topics involving pattern mining and multi-objective optimization have been investigated in the past few years.

3.1 Pattern Discovery and Exceptional Model Mining

Subgroup discovery (SD) has been introduced 25 years ago (Klösgen, 1996) in the Explora discovery assistant. Since then, numerous approaches involving exhaustive (Atzmueller and Puppe, 2006) and heuristic algorithms (Mampaey et al., 2012) were introduced. Early on, research was mainly focused on producing efficient enumeration methods for binary and nominal data (Herrera et al., 2011). Works on numerical concepts, be it for attributes or target variables, were few and far between. The usual way to deal with such data is to discretize the data as a preprocessing step (see, e.g., Fayyad and Irani, 1993) even though it leads to loss of information.

Over 20 years ago, Aumann and Lindell (1999) introduced the concept of *Quantitative Association Rules* 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 the overall dataset. Later on, Webb (2001) proposed an extension of such rules called *Impact Rules*. Jorge et al. (2006) introduced *Distribution Rules*, a type of association rules that involves a statistical distribution on the consequent. In this work, a rule is deemed interesting if its target distribution is significantly different from that of the overall dataset. Their approach however only considers one target and the Kolmogorov-Smirnov test as an interestingness measure.

In the past few years, interest for numerical data took off. Lemmerich et al. (2016) proposes an exhaustive algorithm for subgroup discovery with numerical target concepts. They introduce several quality measures, algorithms and bounds to mine for high quality subgroups in numerical data. However, the approach is only exhaustive with prior discretization of the numerical attributes, leading to non-optimal results. Meeng et al. (2020) introduced a new type of interestingness measure for numerical targets. They 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. Recently, we introduced a new algorithm for optimal subgroup discovery

in purely numerical data (i.e., data where both the attributes and the target are numerical) (Millot et al., 2020a). No discretization is needed and the approach provides a guarantee on the optimality of the returned subgroup, at the cost of scalability in certain applications. While much has been done for numerical data with a unique target variable, these approaches cannot deal with complex models involving multiple targets.

Exceptional Model Mining (Leman et al., 2008; Duivesteijn et al., 2016) was introduced as a generalization of subgroup discovery for multi-target problems. Several approaches, both heuristic (Krak and Feelders, 2015; Moens and Boley, 2014) and exhaustive (Lemmerich et al., 2012) have been developed. In EMM, each problem is linked to a model class and requires a tailored search algorithm. For instance, considering the mining of exceptional correlations (Downar and Duivesteijn, 2017) and the mining of exceptional Bayesian networks (Duivesteijn et al., 2010) require specific quality measures and search algorithms.

Duivesteijn et al. (2012a) take on what they call the "workhorse" of data analysis problems: linear regression. They introduce a new model class for exceptional regression model mining relying on a quality measure based on Cook's distance. Other interesting EMM instances have been investigated. Among them, Duivesteijn and Thaele (2014) and Duivesteijn et al. (2012b) both work with model classes for classification problems. The former approach aims to identify subspaces 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 context of model diagnosis and interpretable machine learning. The latter approach aims to identify and exploit exceptional interdependencies between labels in a multi-label classification setting, allowing the improvement of the classifier overall quality. More recently, we also find proposals about the discovery of exceptional (dis-) agreements between groups (Belfodil et al., 2019), exceptional mediation models (Lemmerich et al., 2020) and exceptional spatio-temporal behavior (Du et al., 2020).

3.2 Multi-Objective Optimization

EMM deals with complex interactions between multiple targets, and Multi-objective Optimization (see, e.g., (Deb et al., 2002)) is a nice example of a task that involves such complex interactions. While some MOO problems can be solved by transforming them into uni-objective ones, most problems require methods based on Pareto optimization (Roijers et al., 2013). The goal is then to design algorithms that approximate the true Pareto front of a given problem as well as possible. Most common approaches resort to generic nature-inspired heuristics (Zhou et al., 2011), including the well-known NSGA-II genetic algorithm (Deb, 2014). 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 the model of the objective functions is known and numerous experiments can be carried out. It is worth noting that since we work with limited data and unknown underlying models, our methods do not compare multi-objective algorithms per se, but rather the

end results of MOO processes (i.e., the Pareto fronts).

While algorithms are able to create sets of solutions, quality indicators that allow for the comparison of different sets of solutions are needed (Li and Yao, 2019). Generally speaking, we find quality indicators that need a reference set of solutions (Schutze et al., 2012) to compare new solution sets against, and quality indicators that require a reference point (Hansen and Jaszkiewicz, 1998) – such as the Nadir point or the anti-ideal point – to be computed. The comparison of algorithms and quality indicators for MOO approaches is important and numerous benchmark functions with various constraints, such as the Fonseca-Fleming function (Fonseca and Fleming, 1995), have been proposed. It is worth noting that our problem is different from the selection of a single good solution from a Pareto front as in Fuente et al. (2018), since we are interested in identifying exceptional deviations and approximations of Pareto fronts thanks to a description in the attribute space.

Although the literature on Pareto-based MOO is well-supplied, 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.

3.3 Cross-Fertilization between EMM and MOO

While both MOO and pattern mining, including both SD and EMM, have been seriously investigated, contributions at the intersection of the two subfields have been few and far between (Srinivasan and Ramakrishnan, 2011). Recently, we introduced a generic framework using actionable subgroup discovery to solve optimization problems when the underlying model is unknown (Millot et al., 2020b). However, this was intrinsically dedicated to uni-objective problems. Carmona et al. (2010) introduced an evolutionary fuzzy system named NMEEF-SD based on the NSGA-II algorithm to discover interpretable and high quality subgroups. While their approach is interesting, 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 (i.e., they consider the Pareto front of subgroups and not the Pareto front of objects of the dataset).

Soulet et al. (2011) have exploited the notion of *skyline queries*, introducing the notion of *skyline patterns*. 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 the set of patterns which lie on the Pareto

front. Their approach presents several advantages: (i) it finds patterns which are non-dominated by any other pattern, (ii) it is generic, as it naturally extends 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 them to compute the set of *skyline patterns* in an efficient way. Ugarte et al. (2017) build on this work, further investigating the relationships between the so-called condensed representations of patterns and skyline pattern mining. As a result, they can build an interesting skypattern mining algorithm based on a dynamic constraint satisfaction problem. The concept of skyline was also exploited in Van Leeuwen and Ukkonen (2013) to mine for skylines of subgroup sets. They present an exhaustive and a heuristic algorithm for the discovery of top-K subgroup sets that offer the best trade-off 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 we are interested in the computation of exceptional models whose interestingness is at the object level.

Finally, in Millot et al. (2021), we introduced a first approach to Exceptional Pareto Front Mining (EPFM). Here, we discuss the limits of this preliminary approach – renamed EPFDM – both in terms of quality measure and applications scenarios. We aim to build better, more generic quality measures, and show that the mining of other types of exceptional Pareto front patterns beyond EPFDM can be interesting.

4 Mining Exceptional Pareto Front Deviations

We want to build a model class for EMM in a MOO setting and 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, called 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. Given two Pareto fronts PF_{target} and $PF_{reference}$, the Partial Pareto Front $PPF(PF_{target}, PF_{reference})$ is equal to:

$$PF_{target} \setminus PF_{reference}$$

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 $PF_{dataset}$ or for PF_{model}. We have $PPF_{model} = PF_{model} \setminus PF_{dataset}$ and $PPF_{dataset} = PF_{dataset} \setminus PF_{model}$. Figure 2 depicts the PPFs of PF_{model} (left) and $PF_{dataset}$ (right). 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.

4.1 Designing Quality Measures for EPFDM

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 MOO 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) show that taking into account both distances provides measures that are more resilient to outliers and uncommonly shaped Pareto fronts. Therefore, we exploit measures that consider both the distance between the partial Pareto front of the subgroup PPF_{model} and the Pareto front

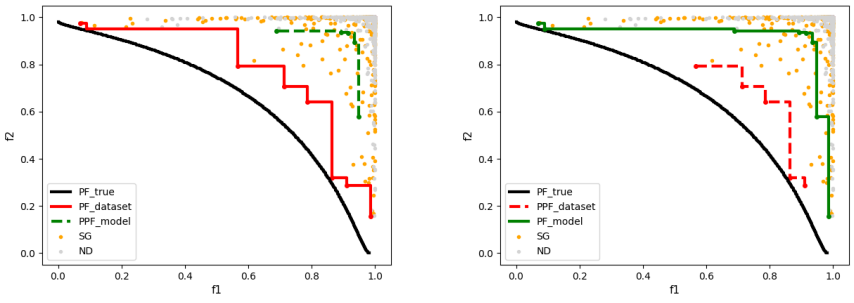

Figure 2: Partial Pareto fronts of PF_{model} (left) and $PF_{dataset}$ (right) in `Fonseca`.

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 targets 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 Target j in the dataset.

Our measures are based on the popular Hausdorff Distance that estimates how far two subsets of points in 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.

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

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

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 Hausdorff Distance, called Averaged Hausdorff Distance (AHD) and introduced in Schutze et al. (2012). The Averaged Hausdorff Distance $AHD(PF_{model}, PF_{dataset})$ between PF_{model} and $PF_{dataset}$ is:

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

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, the MOO literature presents 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 volume-based measure taken from the MOO literature, the so-called Hypervolume (HV) (Zitzler and Thiele, 1998). 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 words, the concept of Partial Pareto Front is only relevant for distance-based measures, and will not be used with HV.

The Hypervolume $HV(PF)$ between a given Pareto front PF and its reference point r is:

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

where λ is the Lebesgue measure.

Informally, the Hypervolume value of a Pareto front is the area (in two dimensions) or volume (in three or more dimensions) 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. The Nadir point is defined as the vector of the worst possible value of each objective according to the optimal true Pareto front. One issue with the Nadir point is that it cannot be precisely estimated in most scenarios. 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 where the underlying model is known, and clearly impossible to compute when the model is unknown. Figure 3 (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 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 3. For this reason, we define our own version of a reference point, that ensures that no object lies outside the enclosed area.

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

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

Informally, the reference point of a dataset is the vector composed of the worst value for each target in the overall dataset. Figure 3 (right) depicts a comparison between our reference point and the typical Nadir point. This novel reference point ensures that the HV can be properly computed for any subset of a given dataset.

Let us now detail how the HV of a given subgroup is computed in EPFDM. We look for subgroups whose removal produce 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.

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.

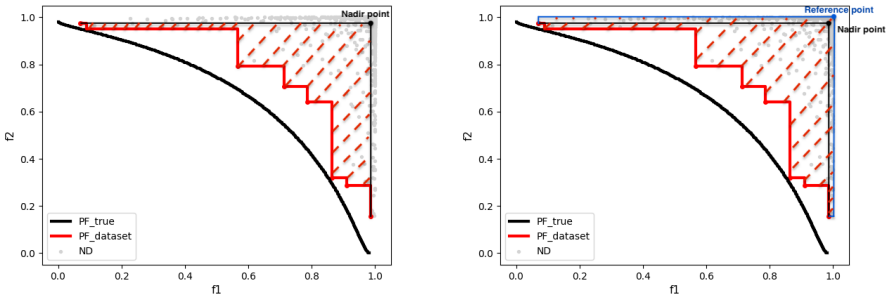

Figure 3: Hypervolume of `Fonseca` with the Nadir point (left) and our reference point (right).

4.2 Algorithm

The source code is available at `https://bit.ly/3EwnWUi`. Detailed explanations of the algorithm are available at Millot (2021). Our enumeration algorithm is based on a top-K beam search (Duivesteijn et al., 2016). 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 its specializations if it does not contain an object that belongs to the true Pareto front.

5 Mining Exceptional Pareto Front Approximations

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. 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. We once again assume that we have to work with objective minimization, and any maximization problem is transformed into a minimization one. For numerical attributes, we also assume that they have been discretized a priori such that all attributes are binary or categorical.

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.

5.1 Designing Quality Measures for EPFAM

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 discard distance-based measures and we focus on volume-based measures like HV, that 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 4. 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.

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.

5.2 Algorithm

The source code is available at `https://bit.ly/3EwnWUi`. Detailed explanations of the algorithm are available at Millot (2021). For the computation of top-K EPFAM, a slightly modified strategy from the one introduced in Section 4 for EPFDM can be used. 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 belong to the true Pareto front is only applied if a minimum support constraint is used.

6 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 a typical EMM setting, 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 the large distance, such subgroups are not interesting. Figure 4 (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 apart 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 4 (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.

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

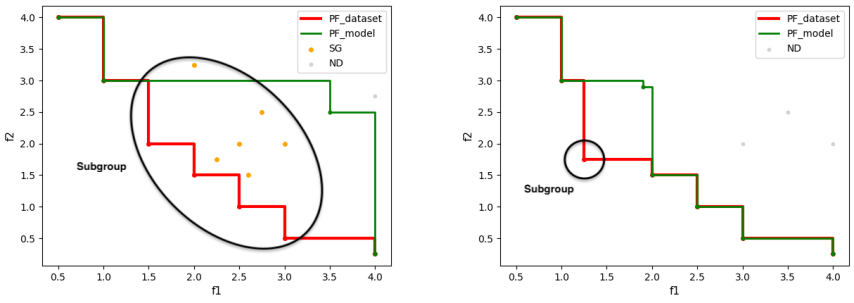

Figure 4: Examples of undesirable subgroups whose removal creates a large deviation from the true Pareto front: a large subgroup that covers most of the dataset (left), a low entropy subgroup that only affects one object of the true Pareto front (right).

The locality indicator of a subgroup p is:

$$Locality(p) = 1 - \left(\frac{m}{M}\right)$$

where M is the total number of objects of the dataset and m is the number of objects of p. This locality indicator 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 indicator might be too strict in some application cases where larger subgroups might be more interesting. Therefore, we add a factor that tunes the importance of the locality indicator.

The locality indicator of a subgroup p, with its importance factor, is:

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

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

To deal with the issue of subgroups with few objects on the true Pareto front creating unwanted large deviations in the Pareto front of the model, we propose several solutions. First, let us use the entropy of the split between the objects of the true 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.

The weighted 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 weighted 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.

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.

For a given subgroup p and a minimum support $minSupp$, we compute the following function MSV:

$$MSV(p, minSupp) = \begin{cases} 0, & \text{if } \frac{n}{N} < \text{minSupp} \\ 1, & \text{if } \frac{n}{N} \geq \text{minSupp} \end{cases}$$

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. The function returns 0 if the minimum support constraint is not satisfied by the subgroup, and 1 otherwise.

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.

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 indicator, and $Generality(p)$ denotes the chosen indicator for the generality of the model (i.e., $Entropy^b$, $Coverage^c$, or MSV).

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

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 indicator, and $Generality(p)$ denotes the chosen indicator for the generality of the model (i.e., $Entropy^b$, $Coverage^c$, or MSV).

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.

It is interesting to note that instead of using an aggregated quality measure, we can also exploit the concept of *skyline patterns* (Soulet et al., 2011) to mine for subgroups that offer the best trade-off between deviation (resp. approximation), locality and generality. This alternative method is empirically evaluated in Section 8.3.

7 Experiments

Let us now consider experiments on both synthetic and real life datasets. The source code and datasets used in our experiments are available at `https://bit.ly/3EwnWUi`. In the following experiments (i.e., for both EPFDM and EPFAM) and unless specified otherwise, the beam width was set to 10, the search depth to 5, the locality factor to 1, and the MSV function was employed with a minimum support of 0.1. 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. When it comes to discretization of the numerical attributes, we apply equal-width, equal-frequency, and binary representations (Meeng and Knobbe, 2021) using 2, 3, 5, 10, 15 and 20 bins on each dataset – except `Obesity` for which the algorithm could not return results in less than 24 hours when using binary representations with over 3 bins – and we retain only the one that leads to the best models. The numerical descriptive attribute values are then replaced with nominals of cardinality equal to the number of bins. In the figures, both red and orange objects belong to the best subgroup.

7.1 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 on interesting local parts of the Pareto Front. Let us first use the synthetic dataset `Fonseca` – based on the Fonseca-Fleming function – introduced in Section 2.1. We compute the best subgroup found by our algorithm with HV_{dev}, HD, AHD and MHD. On this dataset, equal-width discretization was found to be the best method, likely due to the uniform distributions of the variables. Figure 5 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. With HD, we find a model with a large deviation supported by a small subgroup whose description is $\langle x1 = [-0.8, 0.8]$,

$x2 = [-0.8, 0.8]$, $x3 = [-0.8, 0.8]\rangle$. 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.

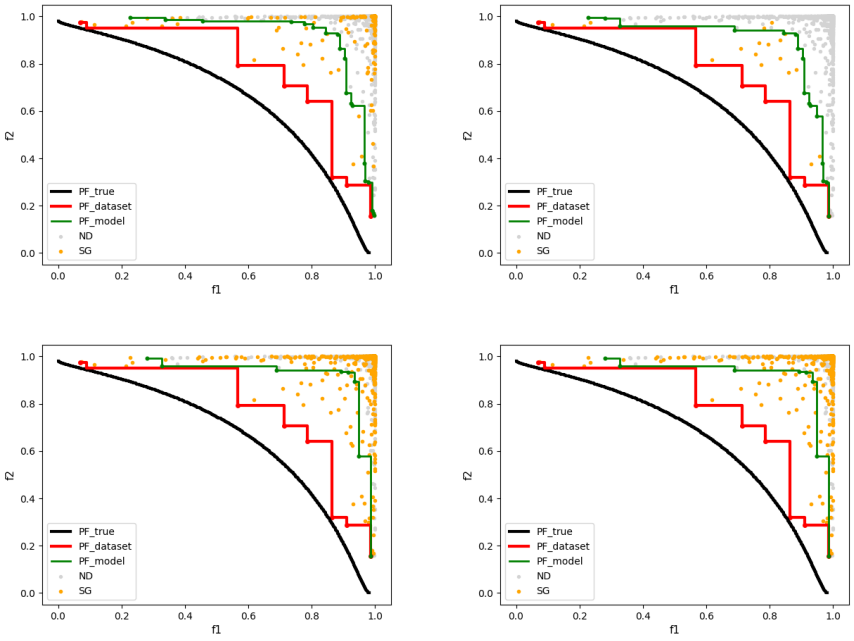

Figure 5: EPFDM best deviations on `Fonseca` with respectively HV_{dev}, HD, AHD and MHD.

Next, we want to see if EPFDM is able to discover a planted subgroup, i.e., a subgroup known a priori that has been purposefully inserted into the data. We first generate 10 objects whose attribute values belong to the following domains: $\langle x1 \in [0.15, 0.25], x3 \in [0, 0.03], x2 \in [0.45, 0.55]\rangle$. This way, we generate a small subset of objects that are all close to each other in the objective space. We then generate a new Fonseca dataset – named `FonsecaPlanted` – made of 1000 random objects, although we generate the objects so that none of them can affect our planted subgroup. The goal is to see if our algorithm is able to discover a subgroup that not only creates a large deviation when removed, but is also very local Pareto front wise, i.e., all objects are grouped around a small part of the Pareto front. We run our EPFDM method with HV_{dev} and HD, and we report the results in Figure 6.

As can be seen, using EPFDM with HV_{dev}, we were able to retrieve a small subgroup, which actually corresponds to the planted subset. However, with HD, we find a larger subgroup. Although the subgroup also contains the planted sub-

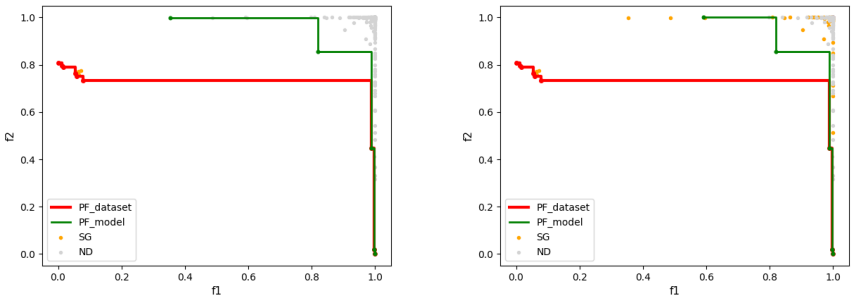

Figure 6: Best deviations found in `FonsecaPlanted` using EPFDM with HV_{dev} (left) and HD (right).

set, it also contains numerous other objects that were not expected. Therefore, the HV_{dev} measure might be better suited to discover small local subgroups.

Let us now consider use cases that are less familiar within the MOO community. Here, the data is limited to the available set (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` – records eating habits and physical conditions of people from Mexico, Peru and Colombia. It was downloaded from the UCI repository (Dua and Graff, 2017). 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 a worse height-weight trade-off. On this dataset, equal-width discretization was found to be the best method. We compute the best models found with HV_{dev} and HD, and we report the results in Figure 7. 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 concerns poor people who use public transportation and have a worse height/weight trade-off than the rest of the population.

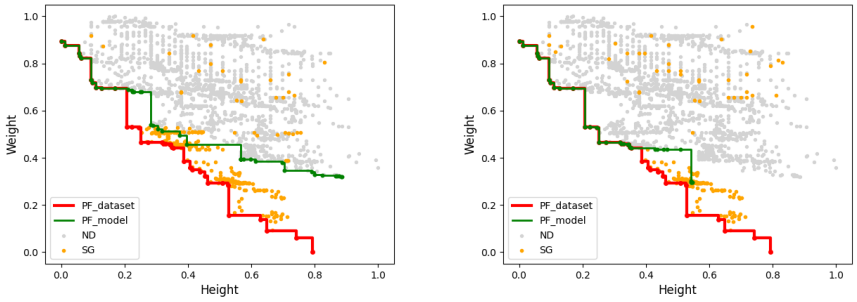

Figure 7: EPFDM best deviations on `Obesity` with HV_{dev} (left) and HD (right).

Let us now consider a third experiment about the trade-off between physical and chemical defense in plant seeds. The dataset named `Plant` is made of 163 observations. It was extracted from the Datadryad website[3]. Each observation is described by two variables: the family and the mass of the plant seed. The objective variables are the fiber – physical defense – and the tannin – chemical defense – contents that both need to be maximized. Again, we compute the best subgroup found with the same measures as for `Obesity`. Here, binary representations was found to be the best discretization method. The results, reported in Figure 8, show two substantially different deviation models, supported by subgroup descriptions. For HV_{dev}, the subgroup description is $\langle mass < 0.146$' $\rangle$, while for HD, the description is $\langle mass > 0.006,\ mass < 0.0708 \rangle$. The first subgroup represents a large subset of plants with no particular observable characteristic, while the second is smaller and involves plants with a restricted weight.

Although we have only considered mining for the best subgroup (i.e., top-1), a relevant task could involve looking for the top-K subgroups. We therefore computed the top-5 subgroups on the `Plant` dataset with both HV_{dev} and HD, and studied their descriptions. While the first two subgroups were significantly different for both measures, the others represented slightly modified versions of the first two. This underlines the diversity issue of top-K EMM, and motivates the exploitation of *skyline patterns* (Soulet et al., 2011) to discover more diverse sets of patterns.

The last dataset named `RealEstate` has been downloaded from the UCI repository (Dua and Graff, 2017). It concerns over 400 sales of houses in Taiwan between 2012 and 2013. It is made of 4 descriptive variables (latitude, longitude,

[3] `https://datadryad.org/stash/dataset/doi:10.5061/dryad.bv5ht`

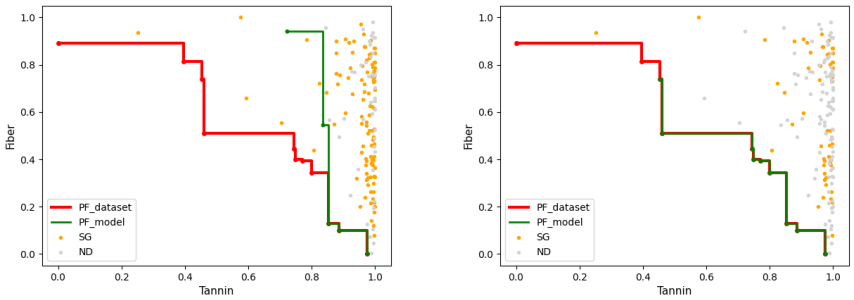

Figure 8: EPFDM best deviations on `Plants` with HV_{dev} (left) and HD (right).

house age, and number of convenience stores within walking distance) 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 9. Binary representations was found to be the best discretization method here. The measures found different deviation models. The subgroup description for HV_{dev} is $\langle latitude < 24.963, longitude < 121.5386, house\_age > 11.89\rangle$ while the subgroup description for HD is $\langle latitude < 24.963, house\_age > 6.6, number\_of\_convenience\_stores > 5\rangle$. The exploitation of these subgroups – i.e., using the constraints on attribute values provided by the description – 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.

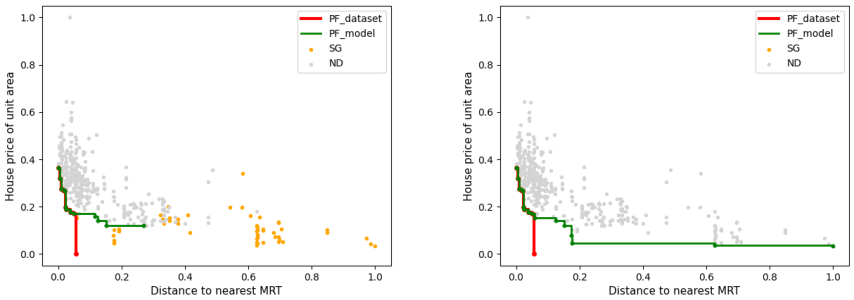

Figure 9: EPFDM best deviations on `RealEstate` with HV_{dev} (left) and HD (right).

7.2 Quantitative Evaluation of EPFDM

As the quality measure introduced offers multiple degrees of freedom, it makes sense to look at the running time issues 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 2. 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 HD has the highest execution time. We studied the size of the subgroups, as well as the number of subgroups evaluated by each measure, and found that neither could explain the time discrepancies. 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 2: Running time comparison (in seconds) of EPFDM on 4 deviation measures.

Measure	Fonseca	Obesity	Plants	RealEstate
HV_{dev}	175.4	15426	2.37	40.31
HD	176.1	17866	2.39	41.79
AHD	175.7	16690	2.39	41.69
MHD	175.9	16215	2.38	41.70

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 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 MSV function for the generality of the model, possible minimum support values are taken in $\{0.1,0.3,0.5\}$. When *Coverage* or *Entropy* is selected instead of MSV, 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 3 where *loc* denotes *Locality*.

First, we can see that using *Entropy* or *Coverage*, regardless of their factor value, seems to yield the worst 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 configurations that use either *Entropy* or *Coverage*. Configurations using MSV with 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 3: 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.	MSV			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

7.3 Relevance of EPFAM

The goal here is to investigate the relevance of EPFAM on the same datasets as EPFDM, 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 11.

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 $\langle x1 = [-0.8, 0.8],\ x2 = [-0.8, 0.8],\ x3 = [-0.8, 0.8]\rangle$ supports the easy generation of very high quality solutions close to the true Pareto front. Indeed, if one wanted to generated new high quality solutions with a high probability, he could sample new data points whose attributes values lie in the intervals of values provided by the subgroup description.

Next, we want to see if we can discover a planted subgroup using EPFAM. We first generate 10 objects whose attribute values belong to the following domains: $\langle x1 \in [-0.8, 0.8],\ x2 \in [-0.8, 0.8],\ x3 \in [-0.8, 0.8]\rangle$. This way, we generate a small subset of objects with high quality that we want to retrieve. We then generate a new Fonseca dataset – named `FonsecaPlanted2` – made of 1000 random objects, although we generate the objects so that none of them can affect our planted subgroup. The goal is to see if our algorithm is able to

discover a subgroup that approximates well the true Pareto front. We run our EPFAM method with HV_{approx} and we report the results in Figure 10.

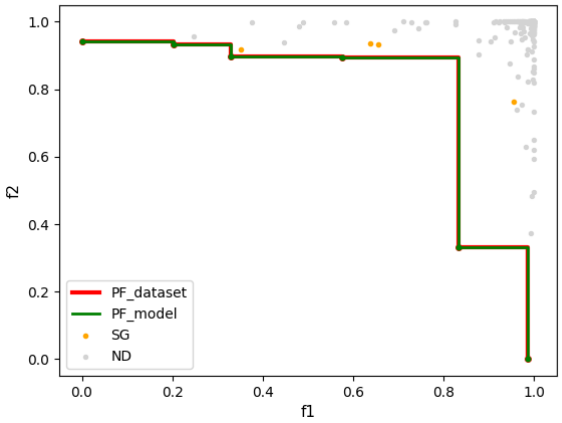

Figure 10: Best approximation found in `FonsecaPlanted2` using EPFAM with HV_{approx}.

As can be seen, with EPFAM we easily discover the planted subgroup, which is the subgroup whose Pareto front best approximates 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 `Plant`, we find a good approximation of the Pareto front, supported by the description $\langle family = \text{'}Combretaceae\text{'}, mass > 0.00251\rangle$. Therefore, the family of plants known as '*Combretaceae*' with a minimum weight of 0.00251 appears as representative for a high quality trade-off between physical and chemical defense in plant seeds.

Finally, we study the best model found on the `RealEstate` dataset. We again find a very good approximation of the true Pareto front, supported by a very small subgroup whose description is:
$\langle latitude > 24.955, latitude < 24.963, house\_age > 3.5,$
$number\_of\_convenience\_stores > 5, number\_of\_convenience\_stores < 7\rangle$.
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 which best approximates it. Please note that this could be due to us using 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.

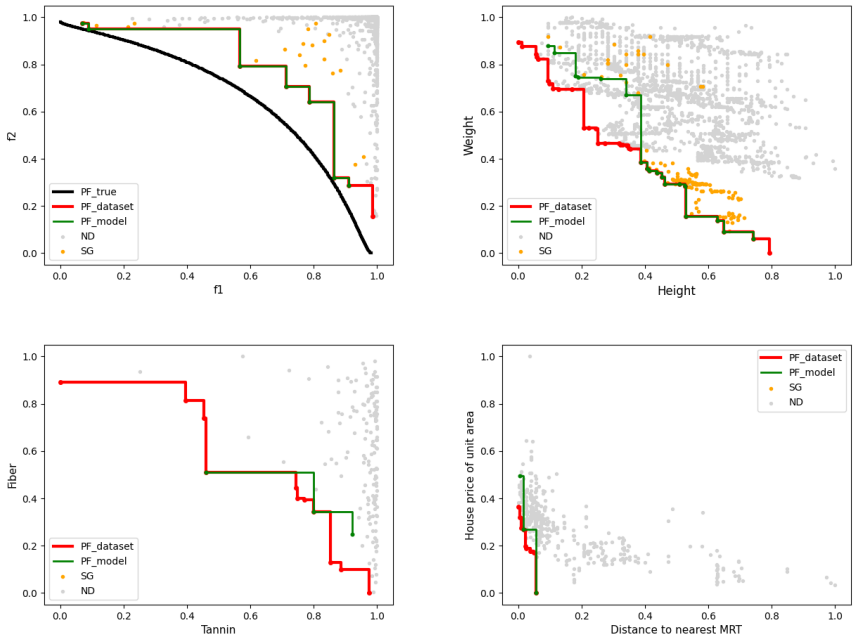

Figure 11: EPFAM best approximations with HV_{approx} on `Fonseca`, `Obesity`, `Plants` and `RealEstate` respectively.

7.4 Quantitative Comparison of EPFDM and EPFAM

Since different configurations of the algorithm have already been considered in Section 4 for several datasets, the same study is not needed for EPFAM. However, studying the running time of EPFAM using HV_{approx} on different datasets and comparing it to EPFDM using HV_{dev} is relevant. The results of these evaluations are available in Table 4. 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 much 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 much larger than the subgroup, hence the computation is faster for EPFDM than for EPFAM.

Table 4: Running time comparison (in seconds) of EPFDM using HV_{dev} and EPFAM using HV_{approx} on 4 datasets.

Dataset	Fonseca	Obesity	Plants	RealEstate
EPFDM	175.4	15426	2.37	40.31
EPFAM	23.5	2320	0.88	7.69

8 Optimizing Plant Growth Recipes in Controlled Environments

8.1 Vertical Urban Farms and Plant Growth Recipes

Urban farming enables the growth of plants in fully controlled environments close to the end-consumers (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), their chemical proportions (Wojciechowska et al., 2015) or their yield (Millot et al., 2020b)). The number of parameters influencing plant growth can be fairly large (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 crops are planted up to the day of harvest. Not only can experts specify a priori the expected values for these descriptive attributes but the actual values can also be recorded through sensors 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 words, the value of a given crop can be quantified with respect to a few different numerical objectives. Given the different growth stages for a given plant, the concept of growth recipe consists in the aggregation of the growth conditions set at each stage, and it can be evaluated by one or several numerical objectives. Our goal is to discover the characteristics of an optimized growth. Experts can then exploit these characteristics to improve recipes. It is worth noting that in the context of plant growth optimization, the underlying model is unknown, preventing the use of traditional genetic or evolutionary methods. Moreover, experiments have to be run in batch (i.e., sets of recipes are tested in parallel)

and in very limited occurrences (i.e., the number of recipes for each test set is low) due to time and cost constraints.

8.2 Plant Growth Recipe Optimization

Earlier, we tackled the problem of recipe optimization with a single objective, the yield, using subgroup discovery (Millot et al., 2020b). Notice however that looking for better recipes that optimize only the yield in a fully controlled environment may be fairly easy (i.e., using more light, enforcing higher temperatures, etc). It makes little sense to optimize the yield while ignoring other aspects like, for instance, the cost: the recipe optimization problem is intrinsically multi-objective. Here we investigate the optimization of plant growth recipes for any number of objectives by leveraging our contributions on EMM. We expect to extract interesting pieces of information on the Pareto front of recipes, such as hollow parts of the objective space (i.e., zones where more data on recipes is needed), interesting local zones containing very good recipes, anomalous recipes, and subsets of recipes representing a good approximation of the overall model.

8.2.1 A Synthetic Data Generator

We consider urban farm recipe optimization as a main application of our methods, as this research is partially funded by a project on urban farms optimization. However, the project is still in early development and we therefore do not yet have access to real farming data. Yet, we were able to perform an empirical study of our proposed approach on this use-case thanks to a crop simulation model, the Python Crop Simulation Environment PCSE[4] simulator. The PCSE was originally 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 12.

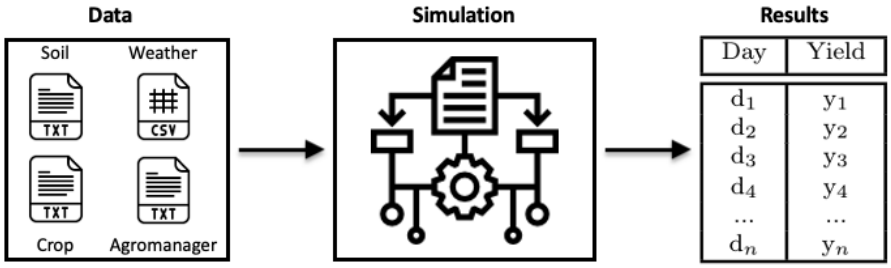

Figure 12: Illustration of the PCSE simulation process. The process takes a set of files as input data, then simulates the growth of the crops, and outputs a file with the day by day evolution of the yield.

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

To simulate the growth of a given crop, the model needs several files as input. First, the *soil file* 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 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 which can be used to control the environment can be found in Table 5.

Table 5: Weather file variable description.

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

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, 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]. Table 6 depicts an example of recipe considering each day between the planting of the crop and its harvest as a distinct growth stage.

Table 6: 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 growth stages does not make sense since (i) the growth process for most plants takes weeks to months, (ii) we know from the literature that the growth process of many plants can be split into just three stages. For this reason, we have been splitting the Weather file to consider 3 stages using the following method: (i) we define the number of days of the growth process until harvesting, (ii) we divide this number by 3 (i.e., 3 stages) which defines the length of a stage, (iii) 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 7.

Table 7: 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 12. 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 designing urban farms, we were given access to the detailed energy consumption of their pilot farm for each environment variable. From this data, we were able to apportion the energy consumption among the environment variables of the farm (which are different from, but close enough to the PCSE variables). However, this information is confidential and cannot be reported 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.

8.2.2 Exploiting EPFM to Optimize Recipes

To apply our EMM approaches to 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. For each recipe, we extract its yield and compute its cost according to the detailed method. Randomly generated examples of such recipes can be found in Table 8.

We kept the number of recipes low on purpose to simulate a real life urban farm where the number of experiments is limited by numerous constraints. To speed up the computation, we discretize the numerical descriptive attribute values and replace them with nominals of cardinality equal to the number of bins. We use a discretization by means of equal-width with 2 cut points (i.e., 3 bins), as it was the discretization that yielded the best results in terms of quality measure when dealing with datasets involving variables with uniform distributions.

Table 8: Examples of growth recipes split in 3 stages (P1, P2, P3), 6 attributes, and 2 objectives (Yield and Cost).

R	RAIN^{P1}	IRRAD^{P1}	$...^{P1}$	RAIN^{P2}	IRRAD^{P2}	$...^{P2}$	RAIN^{P3}	IRRAD^{P3}	$...^{P3}$	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 13 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 description of the subgroup is the following : $\langle IRRAD^{P3} = [20000, 30000], TPMAX^{P2} = [23, 26.5], IRRAD^{P2} = [20000, 30000], TPMAX^{P3} = [23, 26.5]\rangle$. 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 locally 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 front. 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 more useful as an exploratory tool that enables the discovery of interesting knowledge for MOO problems and it cannot be relied upon to design new, more optimized growth recipes. EPFAM was designed to directly answer this kind of problem.

Let us now consider the exploitation of EPFAM to iteratively optimize the yield-cost trade-off of recipes. We will use the following method 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

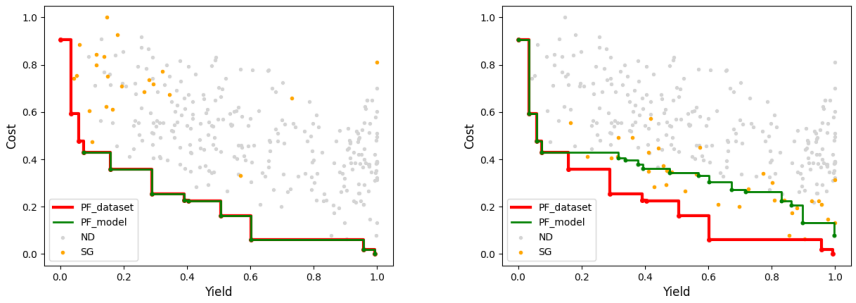

Figure 13: EPFDM best models using HD (left) and HV_{dev} (right).

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. 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 can be observed in Figure 14 (left). 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 $\langle IRRAD^{P1} = [10000, 20000]$, $TPMAX^{P1} = [23, 26.5]$, $WIND^{P2} = [0, 10]$, $TPMAX^{P2} = [23, 26.5]$, $WIND^{P3} = [0, 10]\rangle$.

It is concise and understandable, making it easy to exploit when designing new and hopefully better recipes. 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 (in a real-life scenario, it would be equivalent to planting the crops, letting them grow fully according to their new recipes, and then retrieving their 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

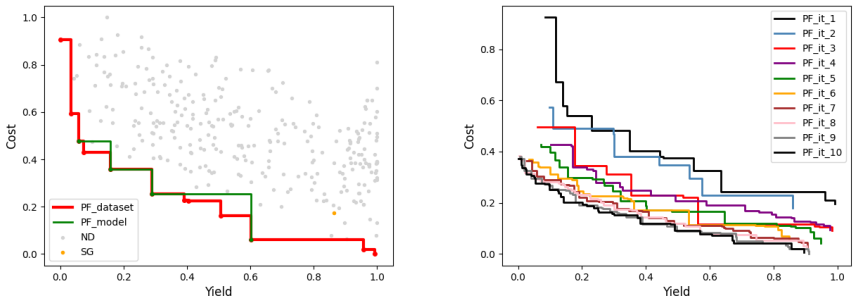

Figure 14: Best approximation found using EPFAM in the original dataset (left) and comparison of the Pareto fronts of the 10 iterations of our EPFAM process (right).

discussion about its results. Figure 14 (right) depicts a comparison between the Pareto front of 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 9. 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 10 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

Table 9: 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

to solve MOO problems.

Table 10: 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 $\times$ 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 15 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).

These observations are confirmed by the numbers available in Table 11. 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

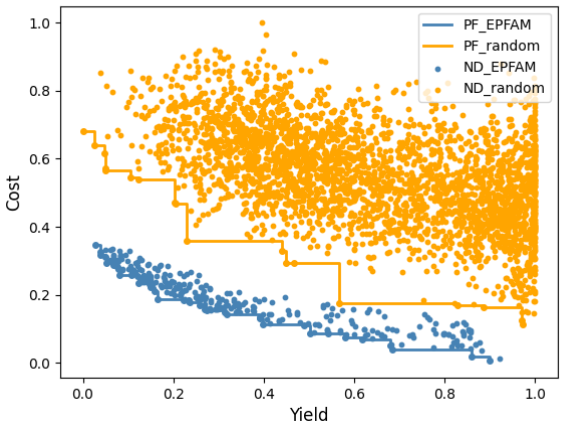

Figure 15: Comparison of the yield-cost trade-offs of the recipes generated through the exploitation of EPFAM with the recipes generated through random search.

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.

Table 11: 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

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) 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 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 16). As 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.

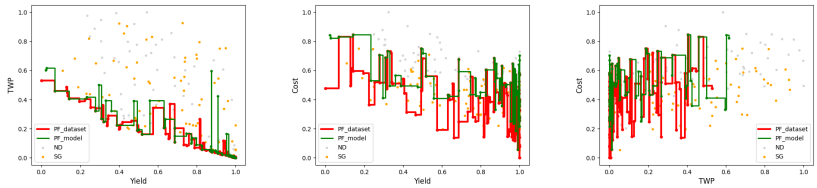

Figure 16: Scatter plots of the EPFDM best model with HV_{dev} showing the pair-wise relationship between objectives.

Let us use the EPFAM method also. It is used with HV_{approx} and we report the best computed model in Figure 17. 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.

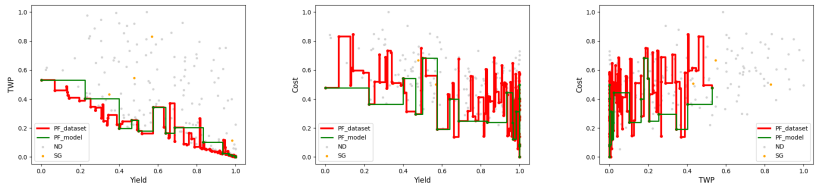

Figure 17: Scatter plots of the EPFAM best model with HV_{approx} showing the pair-wise relationship between objectives.

8.3 Exploiting Skyline Patterns for EPFDM

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* (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 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 algorithm discussed in Section 4. 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, which should not be confused with the local skyline of each level.

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 MSV function is used with a minimum support of 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 18 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 can be seen in Figure 18, 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 12. We see that the running times of Skyline EMM and top-K EMM cross each other when the beam-width is set to 10, that is the most common configuration used in our experiments throughout this paper. It shows that the diversity of Skyline EMM

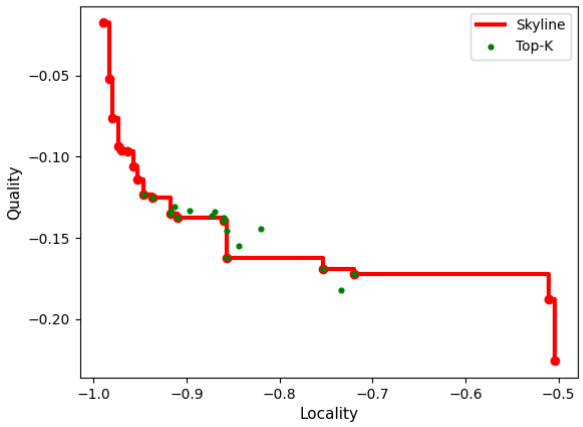

Figure 18: Skyline of EPFDM and comparison between Skyline and q_{EPFDM}.

is obtained without a negative impact on running time.

Table 12: 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

9 Limitations

9.1 Pareto Compliance of Quality Measures

In this work we have considered indicators that enable the comparison of Pareto fronts. We worked with distance-based and volume-based quality measures, that have been inspired by the MOO literature on quality indicators for Pareto fronts comparison (Li and Yao, 2019). A well-known limitation of many quality indicators in MOO is their non compliance with the dominance relation. A quality indicator is said to be *Pareto compliant* (Knowles et al., 2006), if and only if, when comparing two Pareto fronts, the Pareto front that dominates the other also has a better value in terms of the quality indicator. A quality indicator that is Pareto compliant guarantees that it does not contradict the order of Pareto fronts induced by the dominance relation. Given two Pareto fronts A, B and a Pareto compliant indicator, if A is better than B, then B can never dominate A.

Most distance measures are Pareto non-compliant, including the Generational Distance, the Inverted Generational Distance and the Hausdorff Distance,

which were used in our work for EPFDM. In EPFDM, we considered distance maximization, i.e., the farther the Pareto front is from the true Pareto front, the better the quality measure value. According to the Pareto non-compliance of our measures, a given Pareto front could have a larger distance to the true Pareto front than an other front, even though it dominates that same front.

Fortunately, the hypervolume, which has been considered for both EPFDM and EPFAM is one of the few existing quality indicators that is Pareto compliant. We therefore advise to use the volume-based measures.

9.2 Scaling Dependence of Quality Measures

When the considered objectives possess ranges of values that are incommensurable, most quality indicators require a transformation of the objectives so that they can be comparable, i.e., roughly in the same range. This property is called *scaling dependence*, and distance-based measures are subject to it. Indeed, without scaling of the objectives, some objectives could contribute more than others to the quality indicator values.

This is the case for our distance-based measures, which are scaling dependent, meaning that applying a monotone transformation on at least one of the targets could affect the order of solutions (i.e., subgroups) according to our quality measures. This is a known limitation of distance-based indicators in MOO, which is why scaling of the objectives is advised when using EPFDM with these measures. Please note that, to the best of our knowledge, using scaled targets with distance-based quality measures to compare Pareto fronts is relevant, in line with state-of-the-art MOO, and not comparable to treating the problem as single-objective where a unique solution is optimal.

The hypervolume is *scaling independent*, meaning that no scaling is needed a priori, and that applying monotone transformations on one or more targets would have no effect on the order of the resulting solutions. Once again, the use of our volume-based measures seems more relevant than the distance-based measures for EPFM.

To verify those claims, we ran a scaling dependence analysis of the HD and HV measures on our synthetic plant recipes dataset. For each measure, we retrieved the top 3 subgroups on the two following datasets: (i) the plant recipes dataset used in our other experiments, which includes a scaling (i.e., a monotonic transformation) of the targets according to our proposed method in Section 4, and (ii) the original plant recipes dataset, with the targets left untouched (i.e., no normalization).

With EPFDM, using HD, all 3 subgroups are different from each other. For example, the best subgroup found on the normalized dataset is:
$\langle IRRAD^{P2} = [20000, 30000], IRRAD^{P3} = [20000, 30000], TPMAX^{P2} = [23, 26.5], TPMAX^{P3} = [23, 26.5]\rangle$
While the best subgroup found on the non-normalized data is:
$\langle IRRAD^{P1} = [10000, 20000], IRRAD^{P3} = [10000, 20000], TPMAX^{P1} = [26.5, 30], VAP^{P1} = [1.1, 1.35], RAIN^{P3} = [5, 17.5]\rangle$. This confirms the claim that nor-

malization of the targets is crucial for distance-based measures, and that applying monotonic transformations can lead to different results.

With EPFDM and EPFAM, using respectively HV_{dev} and HV_{approx}, we find the exact same top 3 subgroups for both datasets. This confirms the claim that the hypervolume is scaling independent, which makes it more versatile than distance-based measures.

10 Conclusion

We investigate cross-fertilization between Exceptional Model Mining and Multi-objective Optimization. Doing so, we extend our preliminary results published in Millot et al. (2021). We build a new model class called Exceptional Pareto Front Mining. While other approaches that link pattern mining to MOO work at the pattern level (Carmona et al., 2010; Soulet et al., 2011), 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.

Through experiments on both synthetic and real life data, we show the relevance of our methods on several different use cases. EPFDM can be used as an exploratory analysis tool to discover interesting pieces of knowledge about MOO problems. Notably, it 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. EPFAM can be exploited to find exceptionally good approximations of the true Pareto front. In other words, EPFAM enables the generation of high quality solutions with a higher probability.

We present a promising application of EPFM to plant growth recipe optimization in controlled environments like urban farms. This is a relevant context where (i) data can be collected about many descriptive attributes that trace the recipe application (ii) farmer expectation are intrinsically multi-objective. Indeed, optimizing a single objective, say the yield, is rather easy because we can control the climate and thus, for instance, use more light or increase the temperature. In practice, we also need to minimize the cost and/or the needed energy. Both EPFDM and EPFAM have been applied on a simulated recipe optimization scenario, and we have shown how EPFAM can be effectively exploited in a virtuous circle framework to iteratively and efficiently optimize a given process. 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 the number of returned subgroups to be fixed a priori.

As future work, it seems interesting to integrate our method in a fully developed EMM-based iterative optimization process, as it was done in Millot et al.

(2020b) for the case of a unique objective. It seems like a logical next step to fully exploit EPFM potential for multi-objective optimization.

This research is partially funded by the FUI DUF 4.0 French programme (2017-2021) and a CAF America grant (2020-2021).

References

Atzmueller M, Puppe F (2006) Sd-map – a fast algorithm for exhaustive subgroup discovery. In: Proceedings ECML/PKDD, Springer, pp 6–17, DOI 10.1007/11871637_6

Aumann Y, Lindell Y (1999) A statistical theory for quantitative association rules. In: Proceedings ACM SIGKDD, p 261–270, DOI 10.1145/312129.312243

Belfodil A, Duivesteijn W, Plantevit M, Cazalens S, Lamarre P (2019) Deviant: Discovering significant exceptional (dis-) agreement within groups. In: Proceedings ECML/PKDD, Springer, pp 3–20, DOI 10.1007/978-3-030-46150-8_1

Carmona CJ, González P, Del Jesus MJ, Herrera F (2010) NMEEF-SD: Non-dominated multiobjective evolutionary algorithm for extracting fuzzy rules in subgroup discovery. IEEE Transactions on Fuzzy Systems 18:958–970, DOI 10.1109/TFUZZ.2010.2060200

Deb K (2014) Multi-objective optimization. In: Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques, Springer, pp 403–449, DOI 10.1007/978-1-4614-6940-7_15

Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 6(2):182–197, DOI 10.1109/4235.996017

Deb K, Thiele L, Laumanns M, Zitzler E (2005) Scalable test problems for evolutionary multiobjective optimization. In: Evolutionary multiobjective optimization, Springer, pp 105–145

Downar L, Duivesteijn W (2017) Exceptionally monotone models—the rank correlation model class for exceptional model mining. Knowledge and Information Systems 51(2):369–394, DOI 10.1109/ICDM.2015.81

Du X, Pei Y, Duivesteijn W, Pechenizkiy M (2020) Exceptional spatio-temporal behavior mining through bayesian non-parametric modeling. Data Mining and Knowledge Discovery 34(5):1267–1290, DOI 10.1007/s10618-020-00674-z

Dua D, Graff C (2017) UCI Machine Learning Repository. URL `http://archive.ics.uci.edu/ml`

Duivesteijn W, Thaele J (2014) Understanding where your classifier does (not) work–the scape model class for emm. In: Proceedings IEEE ICDM, pp 809–814, DOI 10.1109/ICDM.2014.10

Duivesteijn W, Knobbe A, Feelders A, van Leeuwen M (2010) Subgroup discovery meets bayesian networks–an exceptional model mining approach. In: Proceedings IEEE ICDM, pp 158–167, DOI 10.1109/ICDM.2010.53

Duivesteijn W, Feelders A, Knobbe A (2012a) Different slopes for different folks: Mining for exceptional regression models with cook's distance. In: Proceedings ACM SIGKDD, p 868–876, DOI 10.1145/2339530.2339668

Duivesteijn W, Loza Mencía E, Fürnkranz J, Knobbe A (2012b) Multi-label lego – enhancing multi-label classifiers with local patterns. In: Proceedings IDA, Springer, p 114–125, DOI 10.1007/978-3-642-34156-4_12

Duivesteijn W, Feelders AJ, Knobbe A (2016) Exceptional model mining. Data Mining and Knowledge Discovery 30(1):47–98, DOI 10.1007/s10618-015-0403-4

Fayyad UM, Irani KB (1993) Multi-interval discretization of continuous-valued attributes for classification learning. In: Proceedings IJCAI, pp 1022–1029

Fonseca CM, Fleming PJ (1995) Multiobjective genetic algorithms made easy: selection sharing and mating restriction. In: Proceedings Genetic Algorithms in Engineering Systems: Innovations and Applications, pp 45–52, DOI 10.1049/cp:19951023

Fuente D, Vega-Rodríguez MA, Pérez CJ (2018) Automatic selection of a single solution from the pareto front to identify key players in social networks. Knowledge-Based Systems 160:228–236, DOI 10.1016/j.knosys.2018.07.018

Hansen MP, Jaszkiewicz A (1998) Evaluating the quality of approximations to the non-dominated set. In: Technical University of Denmark, IMM-REP-1998-7

Harper C, Siller M (2015) Openag: A globally distributed network of food computing. IEEE Pervasive Computing 14:24–27

Herrera F, Carmona CJ, González P, Del Jesus MJ (2011) An overview on subgroup discovery: Foundations and applications. Knowledge and Information Systems 29:495–525, DOI 10.1007/s10115-010-0356-2

Huband S, Hingston P, Barone L, While L (2006) A review of multiobjective test problems and a scalable test problem toolkit. IEEE Transactions on Evolutionary Computation 10(5):477–506

Johnson A, Meyerson E, Parra J, Savas T, Miikkulainen R, Harper C (2019) Flavor-cyber-agriculture: Optimization of plant metabolites in an open-source control environment through surrogate modeling. Plos ONE 14:e0213918

Jorge AM, Azevedo PJ, Pereira F (2006) Distribution rules with numeric attributes of interest. In: European Conference on Principles of Data Mining and Knowledge Discovery, Springer, pp 247–258

Klösgen W (1996) Explora: A multipattern and multistrategy discovery assistant. In: Advances in Knowledge Discovery and Data Mining, AAAI, p 249–271

Knowles JD, Thiele L, Zitzler E (2006) A tutorial on the performance assessment of stochastic multiobjective optimizers. TIK-report 214

Krak TE, Feelders A (2015) Exceptional model mining with tree-constrained gradient ascent. In: Proceedings SIAM DM, pp 487–495, DOI 10.1137/1.9781611974010.55

Lavrac N, Kavsek B, Flach P, Todorovski L (2004) Subgroup discovery with CN2-SD. Journal of Machine Learning Research 5(2):153–188

Leman D, Feelders A, Knobbe A (2008) Exceptional model mining. In: Proceedings ECML/PKDD, Springer, pp 1–16

Lemmerich F, Becker M, Atzmueller M (2012) Generic pattern trees for exhaustive exceptional model mining. In: Proceedings ECML/PKDD, Springer, pp 277–292

Lemmerich F, Atzmueller M, Puppe F (2016) Fast exhaustive subgroup discovery with numerical target concepts. Data Mining and Knowledge Discovery 30(3):711–762, DOI 10.1007/s10618-015-0436-8

Lemmerich F, Kiefer C, Langenberg B, Cacho Aboukhalil J, Mayer A (2020) Mining exceptional mediation models. In: Proceedings ISMIS, Springer, pp 318–328, DOI 10.1007/978-3-030-59491-6_30

Li M, Yao X (2019) Quality evaluation of solution sets in multiobjective optimisation: A survey. ACM Computing Surveys 52(2), DOI 10.1145/3300148

Mampaey M, Nijssen S, Feelders A, Knobbe A (2012) Efficient algorithms for finding richer subgroup descriptions in numeric and nominal data. In: Proceedings IEEE ICDM, pp 499–508, DOI 10.1109/ICDM.2012.117

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

Meeng M, de Vries H, Flach P, Nijssen S, Knobbe A (2020) Uni-and multivariate probability density models for numeric subgroup discovery. Intelligent Data Analysis 24(6):1403–1439

Millot A (2021) Exceptional model mining meets multi-objective optimization: application to plant growth recipes in controlled environments. PhD thesis, INSA Lyon, In Press

Millot A, Cazabet R, Boulicaut JF (2020a) Optimal subgroup discovery in purely numerical data. In: Proceedings PaKDD, Springer, pp 112–124, DOI 10.1007/978-3-030-47436-2_9

Millot A, Mathonat R, Cazabet R, Boulicaut JF (2020b) Actionable subgroup discovery and urban farm optimization. In: Proceedings IDA, Springer, pp 339–351, DOI 10.1007/978-3-030-44584-3_27

Millot A, Cazabet R, Boulicaut JF (2021) Exceptional model mining meets multi-objective optimization. In: Proceedings SIAM DM, pp 378–386

Moens S, Boley M (2014) Instant exceptional model mining using weighted controlled pattern sampling. In: Proceedings IDA, Springer, pp 203–214, DOI 10.1007/978-3-319-12571-8_18

Roijers DM, Vamplew P, Whiteson S, Dazeley R (2013) A survey of multi-objective sequential decision-making. Journal of Artificial Intelligence Research 48(1):67–113

Schutze O, Esquivel X, Lara A, Coello CAC (2012) Using the averaged hausdorff distance as a performance measure in evolutionary multiobjective optimization. IEEE Transactions on Evolutionary Computation 16(4):504–522, DOI 10.1109/TEVC.2011.2161872

Soulet A, Raïssi C, Plantevit M, Crémilleux B (2011) Mining dominant patterns in the sky. In: Proceedings IEEE ICDM, pp 655–664, DOI 10.1109/ICDM.2011.100

Srinivasan S, Ramakrishnan S (2011) Evolutionary multi objective optimization for rule mining: A review. Artificial Intelligence Review 36:205–248, DOI 10.1007/s10462-011-9212-3

Ugarte W, Boizumault P, Crémilleux B, Lepailleur A, Loudni S, Plantevit M, Raïssi C, Soulet A (2017) Skypattern mining: From pattern condensed representations to dynamic constraint satisfaction problems. Artificial Intelligence 244:48–69, DOI https://doi.org/10.1016/j.artint.2015.04.003

Van Leeuwen M, Ukkonen A (2013) Discovering skylines of subgroup sets. In: Proceedings ECML/PKDD, Springer, pp 272–287, DOI 10.1007/978-3-642-40994-3_18

Webb GI (2001) Discovering associations with numeric variables. In: Proceedings ACM SIGKDD, p 383–388, DOI 10.1145/502512.502569

Wojciechowska R, Długosz-Grochowska O, Kołton A, Żupnik M (2015) 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

Zhou A, Qu BY, Li H, Zhao SZ, Suganthan PN, Zhang Q (2011) Multiobjective evolutionary algorithms: A survey of the state of the art. Swarm and Evolutionary Computation 1(1):32 – 49, DOI https://doi.org/10.1016/j.swevo.2011.03.001

Zitzler E, Thiele L (1998) Multiobjective optimization using evolutionary algorithms—a comparative case study. In: International conference on parallel problem solving from nature, Springer, pp 292–301

Zitzler E, Thiele L (1999) Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. IEEE Transactions on Evolutionary Computation 3(4):257–271, DOI 10.1109/4235.797969

Zitzler E, Deb K, Thiele L (2000) Comparison of multiobjective evolutionary algorithms: Empirical results. Evolutionary Computation 8(2):173–195