Soft Computing (2019) 23:1423–1437
https://doi.org/10.1007/s00500-018-3646-3

FOCUS

A multi-objective evolutionary approach to Pareto-optimal model trees

Marcin Czajkowski[1] · Marek Kretowski[1]

Published online: 26 November 2018

Abstract

This paper discusses the multi-objective evolutionary approach to induction of model trees. The model tree is a particular case of a decision tree designed to solve regression problems. Although the decision tree induction is inherently a multi-objective task, most of the conventional learning algorithms can only deal with a single objective that may possibly aggregate multiple objectives. The goal of this paper is to demonstrate how a set of non-dominated model trees can be obtained using the Global Model Tree (GMT) system. The GMT framework can be used for the evolutionary induction of different types of decision trees, including univariate, oblique or mixed; regression and model trees. Proposed Pareto approach for GMT allows the decision maker to select desired output model according to his preferences on the conflicting objectives. Performed study covers the regression trees and the model trees with two or three objectives that relate to the tree error and the tree comprehensibility. Experimental evaluation discusses the importance of multi-objective components like crowding function and archive elitist selection, using real-life datasets. Finally, the presented multi-objective GMT solution is confronted with competitive regression and model tree inducers.

Keywords Data mining · Evolutionary algorithms · Model trees · Multi-objective optimization · Pareto optimality · Regression problem

1 Introduction

The most important role of data mining (Fayyad et al. 1996) is to reveal important and insightful information hidden in the data. Among various tools and algorithms that are able to effectively identify patterns within the data, the decision trees (DT)s (Kotsiantis 2013) represent one of the most frequently applied prediction techniques. Tree-based solutions are easy to understand, visualize, and interpret. Their similarity to the human reasoning process through the hierarchical tree structure, in which appropriate tests from consecutive nodes are sequentially applied, makes them a powerful tool (Rokach and Maimon 2008) for data analysts.

Despite 50 years of research on DTs, there is still a space for the improvement (Loh 2014), such as: the search for better structure, splits and models in the leaves; multi-objective optimization or efficient analysis of the cost-sensitive data. To help to resolve some of these issues, evolutionary algorithms (EA)s (Michalewicz 1996) are applied to DTs induction (Barros et al. 2012). The strength of this approach lies in the global search for tree structure, splits in internal nodes and predictions in leaves. It results in simpler but still accurate trees in comparison with ones induced with greedy strategies (Czajkowski and Kretowski 2014).

The goal of this paper is to study a multi-objective evolutionary algorithm for the model tree induction. We discuss the proposed Pareto approach to globally induced regression and model trees which can be seen as an extension of the regression trees. With a generated set of non-dominated predictors, the decision maker will be able to select desired output model according to her/his preferences on the tree comprehensibility and the accuracy. To the best of our knowledge, such study on multi-objective optimization for regression or model trees, surprisingly, has not yet been addressed in the litera-

Communicated by M.A.V. Rodríguez, C. Martín-Vide.

✉ Marcin Czajkowski
m.czajkowski@pb.edu.pl

Marek Kretowski
m.kretowski@pb.edu.pl

[1] Faculty of Computer Science, Bialystok University of Technology, Wiejska 45a, 15-351 Bialystok, Poland

ture. Despite the popularity of DTs, the topic has not yet been adequately explored even for classification trees.

In this work, we focus on the Global Model Tree (GMT) framework (Czajkowski and Kretowski 2014) that can be used for the evolutionary induction of different kinds of regression and model trees (Czajkowski and Kretowski 2016a) and be applied in real-life applications (Czajkowski et al. 2015a). We have extended the original fitness function of the GMT system that applied weight formula or lexicographic analysis according to Pareto-based multi-objective optimization methodology. In each step of evolutionary algorithm, we tried to incorporate the knowledge about the tree induction in to the multi-objective evolutionary search.

The proposed approach significantly extends upon previously performed initial research on Pareto-optimal model trees (Czajkowski et al. 2016). In particular, we:

- extend our solution to work also with the regression trees. Additional experimental comparison between globally induced regression and model trees is also performed;
- propose new, alternative crowding functions, that involve Bayesian Information Criterion (BIC) (Schwarz 1978) weight fitness value and the objective weights;
- perform extensive experimental evaluation that includes: analysis of new real-life datasets; regression and model trees; two-objective and three-objective optimization and visualization;
- show the significance of the crowding distance on the final form of the Pareto front.

This paper is organized as follows. Section 2 provides a brief background, and Sect. 3 describes in details proposed Pareto-optimal search for the GMT framework. Section 4 presents experimental validation of our approach on real-life datasets. In the last section, the paper is concluded and possible future works are outlined.

2 Background

In this section, we want to present some background information on DTs, multi-objective optimization and refer to the related works.

2.1 Decision trees

Different variants of DTs (Loh 2014) may be grouped according to the type of problem they are applied to, the way they are induced, or the type of their structure. In this paper, we focus on regression trees that may be considered as variants of decision trees designed to approximate real-valued functions instead of being used for classification tasks. In case of the simplest regression tree, each leaf contains a constant value, usually an average value of the target attribute. A model tree can be seen as an extension of the typical regression tree (Malerba et al. 2004; Quinlan 1992). The constant value in each leaf of the regression tree is replaced in the model tree by a linear (or nonlinear) regression function. To predict the target value, the new tested instance is followed down the tree from a root node to a leaf using its attribute values to make routing decisions at each internal node. Next, the predicted value for the new instance is evaluated based on a regression model in the leaf. Examples of predicted values of classification, regression, and model trees are given in Fig. 1. The gray level color of each region represents a different class label (for a classification tree), and the height corresponds to the value of the prediction function (regression and model trees).

Although regression trees are not as popular as classification trees, they are highly competitive with different machine learning algorithms (Ortuno et al. 2015) and are often applied to many real-life problems (Fakhari and Moghadam 2013; Liu et al. 2016).

In this paper, we study the evolutionary induced model trees; therefore, to go further, we must briefly describe the process of learning of DT based on the training set. The two most popular concepts for DT are a top-down induction and a global approach. The first is based on a greedy procedure known as the recursive partitioning (Rokach and Maimon 2005). In the top-down approach, the induction algorithm starts from the root node where the locally optimal split is searched according to the given optimality measure. Next, the training instances are redirected to the newly created nodes, and this process is repeated for each node until a stopping condition is met. Additionally, post-pruning (Esposito et al. 1997) is usually applied after the induction to avoid the problem of over-fitting the training data. Inducing the trees with greedy strategy is fast and generally efficient, but often produces only locally optimal solutions.

One of the most popular representatives of top-down induced regression trees is a solution called Classification And Regression Tree (CART) proposed by Breiman et al. (1984). The algorithm searches for a locally optimal split that minimizes the sum of squared residuals and builds a piecewise constant prediction with each terminal node fitted with the training sample mean. Other solutions have managed to improve the prediction accuracy by replacing single values in the leaves with more advanced models. The M5 system (Quinlan 1992) proposed by Quinlan induces a tree that contains multiple linear models in the leaves. A solution called Stepwise Model Tree Induction (SMOTI) by Malerba et al. (2004) uses two types of internal nodes: splitting nodes and regression nodes. The multiple regression model associated with a leaf is composed of straight-line regression functions found along the path from the root to that leaf. All aforementioned methods induce trees with the greedy strategy, which

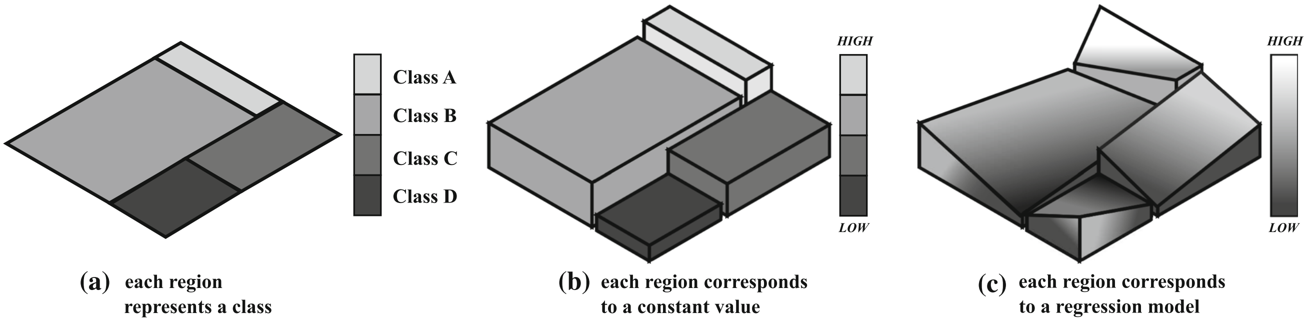

Class A
Class B
Class C
Class D
HIGH
LOW
HIGH
LOW
(a) each region represents a class
(b) each region corresponds to a constant value
(c) each region corresponds to a regression model

Fig. 1 An illustration of predicted values of **a** classification, **b** regression, and **c** model trees

is fast and generally efficient but often produces only locally optimal solutions.

The global induction for the DTs limits the negative effects of locally optimal decisions. It simultaneously searches for the tree structure, tests in the internal nodes, and models in the leaves. This process is obviously much more computationally complex but can reveal hidden regularities that are often undetectable by greedy methods. The global induction is mainly represented by systems based on an evolutionary approach (Barros et al. 2012, 2015); however, there are solutions that apply, for example, ant colony optimization (Boryczka and Kozak 2015).

In the literature, there are relatively fewer evolutionary approaches for the regression than for the classification. Popular representatives of EA-based regression trees are:

- TARGET (Fan and Gray 2005)—evolves a CART-like regression tree with basic genetic operators;
- STGP (Hazan et al. 2006)—strongly typed Genetic Programming (GP) approach;
- E-Motion (Barros et al. 2011)—globally induces model trees that implement a standard 1-point crossover and two different mutation strategies;
- GPMCC (Potgieter and Engelbrecht 2008)—evolves the model trees with nonlinear regression models in the leaves;
- GASOPE (Potgieter and Engelbrecht 2007)—composed from GP to evolve the structure of the model trees and Genetic Algorithm (GA) to evolve polynomial expressions.

2.2 Multi-objective evolutionary algorithms

Real-world optimization problems are usually characterized by multiple objectives which often conflict with each other. Main goal of multi-objective optimization is to minimize all objective functions simultaneously. As it is often impossible, a set of widely spread trade-off solutions is sought instead. Pareto dominance (Pappalardo 2008) searches not for one best solution, but rather for a group of solutions in such a way, that selecting any one of them in place of another will always sacrifice quality of at least one of its objectives, while improving it for at least one other. Let us consider m conflicting objectives that need to be minimized simultaneously. A solution $\mathbf{A} = \{a_1, a_2, \ldots, a_m\}$ is said to dominate solution $\mathbf{B} = \{b_1, b_2, \ldots, b_m\}$ (symbolically denoted by $\mathbf{A} \prec \mathbf{B}$) if and only if:

$$(\mathbf{A} \prec \mathbf{B}) \Leftrightarrow (\forall i)(a_i \leq b_i) \wedge (\exists i)(a_i < b_i), \qquad (1)$$

where a_i and b_i are the objectives ($a_i \in \mathbf{A}, b_i \in \mathbf{B}$) and $1 \leq i \leq m$. The Pareto-optimal set is constituted only of solutions that are not dominated by any other solutions:

$$\{\mathbf{A} | \neg(\exists \mathbf{B}, \mathbf{B} \prec \mathbf{A})\}. \qquad (2)$$

The set of all Pareto-optimal solutions is referred to as the Pareto front. Thus, the goal of multi-objective problems (MOP)s is to find the Pareto front and present such set of multiple alternative solutions to the decision maker for consideration.

Over the past decades, a variety of multi-objective evolutionary algorithms (MOEA)s have been developed to solve MOPs (Hiwa et al. 2015). MOEA is capable of returning a set of Pareto-optimal solutions in just a single run of the algorithm. Two most popular and state-of-the-art algorithms are Strength Pareto Evolutionary Algorithm 2 (SPEA2) (Zitzler and Thiele 1999) and Non-dominated Sorting Genetic Algorithm-II (NSGA-II) (Deb et al. 2002). Both solutions are often a benchmark procedures in most MOEA studies and show fast convergence to the Pareto-optimal set and a good spread of solutions.

The NSGA-II compares interactively pairs of alternatives solutions to identify multiple domination fronts. Each front is identified by the set of solutions with equal number of the domination count. Thus, the first front contains only those solutions which are not dominated by any other solutions (domination count equals to zero), second front contains solutions with domination count equals to one and so on. The crowded-comparison operator ($\prec_n$) helps ordering (ranking) the solutions. In NSGA-II, the crowding distance is used for diversity preservation and to maintain a well-spread Pareto front.

The main differences between NSGA-II and SPEA2 are the diversity assignment, replacement, and archiving. In contrast to the crowding distance proposed in NSGA-II, SPEA2 applies the k-nearest neighbor approach. The NSGA-II algorithm uses a population-size elitist replacement, whereas SPEA2 uses external list with non-dominated solutions. However, two algorithms are similar in that both use binary tournament as their selection method.

2.3 Multi-objective optimization and the decision trees

In case of the DT induction, it is advisable to maximize the predictive performance and to minimize the complexity of the output tree. Using multi-objective optimization in comparison with a single evaluation measure results in much more acceptable overall performance of the predictor. In the context of DTs, a direct minimization of the prediction accuracy measured in the learning set usually leads to the over-fitting problem. In the typical top-down induction of DTs (Rokach and Maimon 2005), this problem is partially mitigated by defining a stopping condition and post-pruning (Esposito et al. 1997).

There are three popular multi-objective optimization strategies (Barros et al. 2012): weight formula, lexicographic analysis, and Pareto dominance. The weight formula transforms a multi-objective problem into a single-objective one by constructing a single formula that combines all objectives. The main drawback of this strategy is the need to find adjusted weights for the measures. The lexicographic approach analyzes the objective values for the individuals one by one based on the priorities. This approach also requires defining thresholds; however, adding up non-commensurable measures, such as tree error and size, is not performed. In contrast to Pareto-dominance approach, both aforementioned solutions are already applied for evolutionary induction of regression and model trees (Barros et al. 2011; Czajkowski and Kretowski 2014).

Although Pareto-optimal approach is popular in machine learning (Jin and Sendhoff 2008), it has not been explored for regression or model trees yet. However, in the literature we may find some attempts performed for classification trees (Afsari et al. 2013). In Zhao (2007), the author proposes Pareto-optimal DTs to capture the trade-off between different types of misclassification errors in a cost-sensitive classification problem. Such a multi-objective strategy is also applied to top-down induced trees (Kim 2004) to minimize two objectives: classification error rate and tree size (measured by the number of tree nodes). The Pareto optimality for greedy induced oblique DTs is investigated in Pangilinan and Janssens (2011). The authors show that an inducer, that generates the most accurate trees, does not necessarily generate the smallest trees or ones that are included in Pareto-optimal set.

3 Pareto-optimal search in GMT

In this section, we present our multi-objective approach for evolutionary induced regression and model trees. At first, we briefly describe the system called Global Model Tree (GMT). Next, we illustrate how to efficiently adapt the Pareto-based approach in the GMT's fitness function.

3.1 Global model tree

The general structure of the GMT system follows a typical EA framework (Michalewicz 1996) with an unstructured population and a generational selection.

3.1.1 Representation and selection

The GMT framework allows evolving all kinds of tree representations (Czajkowski and Kretowski 2016a) e.g., univariate, oblique, mixed; regression and model. In our description, we focus on univariate model trees (Czajkowski and Kretowski 2014); however, our study can be easily applied to different types of trees. Model trees are represented in their actual form as traditional univariate trees, so every split (test) in the internal node is based on a single attribute. Each tree leaf contains a multiple linear regression model that is constructed with learning instances associated with that leaf.

The selection mechanism is based on the ranking linear selection (Michalewicz 1996) with the *elitist strategy*, which copies the best individual found so far to the next population. Evolution terminates when the fitness of the best individual in the population does not improve during the fixed number of generations (default: 1000). In case of a slow convergence, maximum number of generations is also specified (default: 10,000), which limits the computation time.

3.1.2 Genetic operators

Tree-based representation requires developing specialized genetic operators corresponding to classical mutation and crossover. Application of the operators can modify the tree structure, tests in internal nodes, and models in the leaves. The crossover operator attempts to combine elements of two existing individuals (parents) to create a new solution. The mutation operator makes random changes in some places of selected individuals. The GMT framework (Czajkowski and

Kretowski 2014, 2016a) offers several specialized variants of crossover and mutations, e.g.,

- replace one of the following: subtree, branch, node, or test between two affected individuals or the best individual found so far;
- prune the internal node and transform it into the leaf with a new multivariate linear regression model;
- expand the leaf into the internal node;
- modify the test in internal nodes (shift threshold, replace tested attribute);
- change linear regression models in the leaves (add, remove, or change attributes).

3.1.3 Fitness function

Fitness function is one of the most important and sensitive elements in the design of EA. It drives the evolutionary search process by measuring how good a single individual is in terms of meeting the problem objectives. Currently, there are two multi-objective optimization strategies implemented in the GMT framework: weight formula and lexicographic analysis. Among various weight formulas tested within the GMT system, the BIC shows the highest performance with regression and model trees. Its fitness is given by:

$$\mathrm{Fit}_{\mathrm{BIC}}(T) = -2 * \ln(L(T)) + \ln(n) * k(T), \tag{3}$$

where $L(T)$ is the maximum of the likelihood function of the tree T, n is the number of observations in the data, and $k(T)$ is the number of model parameters in the tree. The log(likelihood) function $L(T)$ is typical for regression models and can be expressed as:

$$\ln(L(T)) = -0.5n * [\ln(2\pi) + \ln(\mathrm{SS}_e(T)/n) + 1], \tag{4}$$

where $\mathrm{SS}_e(T)$ is the sum of squared residuals of the tree T. In this measure of goodness of fit, the term $k(T)$ can be viewed as a penalty for over-parametrization. It reflects the tree complexity, which for regression trees equals to the number of nodes (denoted as $Q(T)$), whereas for model trees it also encompasses the number of attributes in the linear models in the leaves (denoted as $W(T)$).

When the lexicographic analysis is applied as a fitness function, each pair of individuals is evaluated by analyzing objectives $\mathrm{SS}_e(T)$, $Q(T)$, and $W(T)$ in order of priorities. The first priority is set to the establishment error because the researches usually seek for most accurate trees and next to the number of terminal nodes to prevent over-fitting and overgrown trees. The last measure $W(T)$ keeps the models in the leaves as simple as possible.

3.1.4 Smoothing

The GMT system uses a form of smoothing (Quinlan 1992) that was initially introduced in the $M5$ algorithm for a univariate model tree. Smoothing is applied only to the best individual returned by EA when the evolution is finished. Its role is to reduce sharp discontinuities that may occur between adjacent linear models in the leaves. For every internal node of the tree, the smoothing algorithm generates an additional linear regression model that is constituted from features that appear in subtrees. This way, each tested instance is predicted not only by a single model at a proper leaf but also by the different linear models generated for each of the internal nodes up to the root node.

In the first step of smoothing (Czajkowski and Kretowski 2014), we predict value for a test instance according to the model in the appropriate leaf. Then, this value is smoothed and updated along the path back to the root by linear models calculated in each nodes. If the instance follows branch S_i of subtree S, let n_i be the number of training instances at S_i, $\mathrm{Pred}(S_i)$ the predicted value at S_i, and $M(S)$ the value given by the model at S. The predicted value backed up to S is:

$$\mathrm{Pred}(S) = \frac{n_i * \mathrm{Pred}(S_i) + k * M(S)}{n_i + k}, \tag{5}$$

where k is a smoothing constant (Quinlan 1992) (default: 10).

3.2 GMT Pareto-based approach

The main goal of the multi-objective optimization is to find a diverse set of Pareto-optimal solutions, which may provide insights into the trade-offs between the objectives. Current GMT fitness functions: weight formula and lexicographic analysis, yield only a limited subset of solutions that may not even belong to the Pareto front.

While evolving regression trees, one can distinguish two objectives that could be minimized: prediction error measured often with Root Mean Squared Error (RMSE) and the number of nodes in the tree ($Q(T)$). In case of the model trees, one more objective occurs—the number of attributes in regression models located in the leaves ($W(T)$). The last two objectives (number of nodes and attributes) are partially depended and may fall under one single objective denoted as a tree comprehensibility.

In the GMT system, we have applied the principle of the NSGA-II workflow. However, most of its elements like sorting strategy itself, crowding and elitism are specialized in order to fit more accurately to the problem of evolutionary induction. Figure 2 shows the general GMT schema together with the proposed Pareto-based extension.

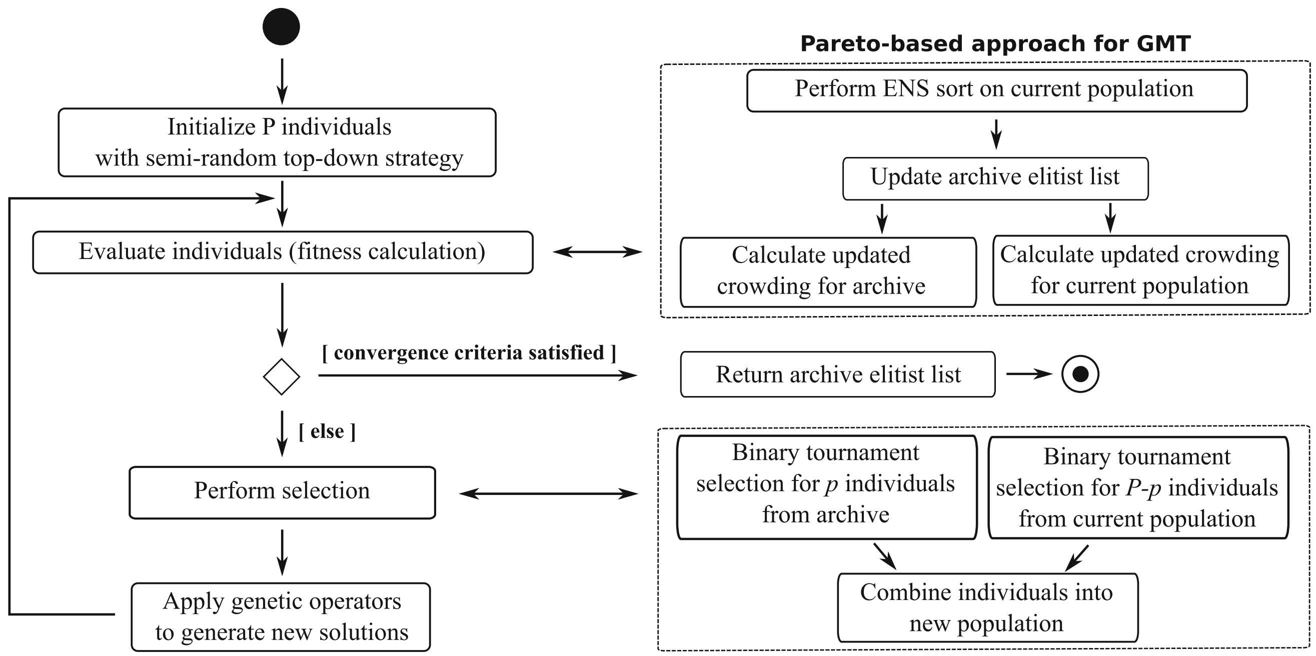

Pareto-based approach for GMT
Perform ENS sort on current population
Update archive elitist list
Calculate updated crowding for archive
Calculate updated crowding for current population
Initialize P individuals with semi-random top-down strategy
Evaluate individuals (fitness calculation)
[convergence criteria satisfied]
Return archive elitist list
[else]
Perform selection
Binary tournament selection for p individuals from archive
Binary tournament selection for P-p individuals from current population
Combine individuals into new population
Apply genetic operators to generate new solutions

Fig. 2 General schema of GMT evolutionary induction together with proposed Pareto-based extension

3.2.1 Sorting strategy

In the first step of proposed Pareto-based extension, a more recent search strategy called efficient non-dominated sorting (ENS) (Zhang et al. 2015) is applied. The ENS strategy was selected due to its efficiency. Experimental evaluation of ENS showed that it outperforms other popular non-dominated sorting approaches especially for optimization problems having a small number of objectives which is here the case.

The ENS algorithm is conceptually different from most of existing non-dominated sorting methods. Typical non-dominated sorting approaches compare a solution with all other solutions in the population before assigning it to a front. ENS determines the front each solution belongs to one by one, and compares it only with those that have already been assigned to a front. This is made possible by the fact that in ENS, the population is sorted in one objective before ENS is applied. Thus, a solution added to the front cannot dominate any solutions that are added before. As a result, ENS can avoid a large number of redundant dominance comparisons, which significantly improves the computational efficiency.

3.2.2 Elitist archive and new population

In the second step of proposed extension (see Fig. 2), the archive fronts are updated. The NSGA-II approach maintains a population-size set of non-dominated solutions that is later combined with the offspring population. However, in considered case, where population size is small (50 individuals), many possibly interesting, from the decision-maker point of view non-dominated solutions may be lost. Therefore, we have applied different strategy that allows storing all non-dominated solutions investigated so far during the search (Zitzler and Thiele 1999). Solutions from Pareto front are stored in an elitist list, which is updated each time a new solution from the current population dominates one in the list. Although this operation is more computationally expensive, it is still acceptable as for model trees the Pareto front is not very large.

In addition, the proposed approach differs from NSGA-II in a way it creates a new population. In NSGA-II, the current and offspring population are merged into a new population. Each solution is ranked according to its non-domination level (1 is the best level, 2 is the next-best level, and so on) and in case of a draw, crowding distance is considered. Next, the binary tournament is used as a selection method, but the selection criterion is based on the crowded-comparison operator. Due to storing the full list of non-dominated solutions in the archive, we are able to apply strategy proposed in Ishibuchi and Murata (1998). We reserve a room for p elitist solutions in the next population (default: half of the population size P). In this strategy, $P-p$ solutions are selected from parents and newly created offspring and p solutions are selected from the stored elitist list. Both sets use the binary tournament as a selection method. The elitist solutions are scored with the crowding distance (as they all belong to a non-dominated set), and the current population is scored alike in original NSGA-II algorithm.

3.2.3 Crowding distance

With all non-dominated solutions archived and almost full Pareto front investigated in GMT, it seems that the impact of the crowding distance on visualization of the Pareto front is small. However, crowding distance plays an important role in diversification of the population and, thus, faster constitution of the Pareto front. Within a population-size archive approach alike in NSGA-II, the crowding distance seems to have a strong impact on the visualization of the Pareto front.

In the proposed extension, we have adapted the updated crowding distance procedure (Fortin and Parizeau 2013) for the NSGA-II. The main improvement of the crowding distance calculation focuses on using unique fitness when two or more individuals share identical value. Such case in NSGA-II algorithm causes the crowding distance of the individual to either become 0, or to depend on the individuals position within the Pareto front sequence.

Algorithm 1 presents the crowding distance computation where $\mathcal{F}$ represents a Pareto front composed of $N = |\mathcal{F}|$ individuals. To keep simplicity, we assume that there are no individuals with identical fitness values. Let $\mathcal{F}[i].m$ refer to the m-th objective of the i-th individual in front $\mathcal{F}$, and the parameters $f\min_m$ and $f\max_m$ are the minimum and maximum values for objective m. At the beginning, distance of every individual is initialized to 0 (line 3). Next, for each objective m the individuals are first sorted in ascending order based on their value for this objective (line 6). Alike in NSGA-II, the solutions with smallest and largest objective value (boundary solutions) are assigned an infinite crowding distance (line 7). Finally, for each intermediary solution the algorithm calculates the crowding distance.

Algorithm 1 Crowding distance computation algorithm

```
CrowdingDistance($\mathcal{F}$)
1: $N = |\mathcal{F}|$
2: for $i \in \{1, \ldots, N\}$ do
3:     $\mathcal{F}[i]_{dist} = 0$
4: end for
5: for $m \in \{1, \ldots, M\}$ do
6:     $SORT(\mathcal{F}, m)$
7:     $\mathcal{F}[1]_{dist} = \mathcal{F}[N]_{dist} = \infty$
8:     for $i \in \{2, \ldots, N-1\}$ do
9:         $\mathcal{F}[i]_{dist} = f(\mathcal{F}[i], m)$
10:    end for
11: end for
```

In the paper, we have tested different crowding distance rankings:

(A) crowding distance based on NSGA-II procedure where algorithm computes the normalized difference between the following and preceding individuals for the current objective m, and sums it to the individual crowding distance: $f(\mathcal{F}[i]_{dist}) = \mathcal{F}[i]_{dist} + \frac{\mathcal{F}[i+1]_m - \mathcal{F}[i-1]_m}{f\max_m - f\min_m}$;

(B) crowding distance based on the BIC weight fitness value. It is calculated by Eq. 3 for each individual using the data from all objectives. This way, the Pareto front explores directly the surrounding solutions of the weight formula;

(C) combination of (A) and (B) crowding distances with percentage share to focus Pareto front on the surroundings area of the important regions (default 50%);

(D) crowding distance with the objective weight preferences, variants: (D + A), (D + B) and (D + C).

In case of decision trees, the predictive accuracy is usually considered more important than its comprehensibility. Let us consider two trees: $T1$ and $T2$ where $T1$ has 20% smaller prediction error but also 20% larger size. Most of the researches would clearly prefer the $T1$ over $T2$; however, the Pareto approach would consider them equally important as none of these two trees dominate the other. Therefore, in context of DT and the Pareto front, we considered the weights preferences (crowding function (D)) in the multi-objective optimization (Friedrich et al. 2013). With incorporating preference information into the crowding distance, we managed to focus on interesting regions in the objective space. For example, by increasing weight of the tree size objective the Pareto front is more likely to contain trees with various sizes rather than with different errors (or number of attributes in the leaves in case of model trees). In the experimental section, we show some examples how the changes in the crowding function impact the final form of non-dominated decision tree population.

4 Experimental validation

In this section, three sets of experiments are presented. First, we would like to show some visualization of Pareto front for regression trees and model trees as well as the results for the weight and lexicographic GMT fitness functions. Next, we compare the results from the proposed approach with the original GMT solution as well as with two popular top-down inducers that are most adequate greedy counterparts of GMT. Finally, we discuss how the Pareto front can be adjusted based on analytical preferences. In particular, we focus on the simplicity of the generated Pareto front by modifying the crowding function on a population-size elitist archive.

4.1 Setup

To assess the performance of the proposed approach in solving real-life problems, seven publicly available datasets (see

Table 1 Characteristics of the analyzed datasets

Dataset		Number of features	
Name	Instances	Numeric	Nominal
Abalone (AB)	4177	7	1
Ailerons (AI)	13, 750	40	0
Delta ailerons (DA)	7129	5	0
Delta elevators (DE)	9517	6	0
Kinemaics (KI)	8192	8	0
Pole (PO)	15, 000	48	0
Stock (ST)	950	9	0

Table 1) from Louis Torgo repository (Torgo 2017) were analyzed.

Datasets without provided testing sets were randomly divided into the training (66.7%) and testing (33.3%) parts.

In this experimental validation, the Pareto extension for the GMT system is denoted as pGRT when applied to generate regression trees and pGMT when the system is applied to induce the model trees. We have also tested the GMT framework with the weight fitness function (wGRT for regression and wGMT for model trees) and accordingly lGRT and lGMT for the lexicographic fitness function. In all the experiments reported in this paper and for all datasets, we used one default set of parameters as recommended in Czajkowski and Kretowski (2014): the population size equals to 50, the probability of the mutation single node is 0.8, and the probability to crossover inducers equals 0.2. The regression and model trees that were used for the purpose of comparison with GMT in the second set of experiments were tested using the WEKA system also with default settings (Hall et al. 2009).

4.2 Visualization of the Pareto front

In this set of experiments, we visualize the Pareto front for regression and model trees on three datasets: Abalone (AB), Kinematics (KI), and Stock (ST). Pareto front for the regression trees can be visualized in 2 dimensions as there are only two objectives that can be minimized: prediction error (RMSE) and the number of nodes in the tree ($Q(T)$). List of non-dominated model trees can be illustrated in 3 dimensions as one more objective can be considered ($W(T)$). However, number of nodes and attributes ($Q(T)$ and $W(T)$) may be viewed as one objective that refers to the tree comprehensibility. Therefore, in case of the model trees we present the results for the fitness function with 2 objectives where number of nodes and total number of attributes in all models in leaves are summed with equal weight; and with 3 objectives where all measures are analyzed separately.

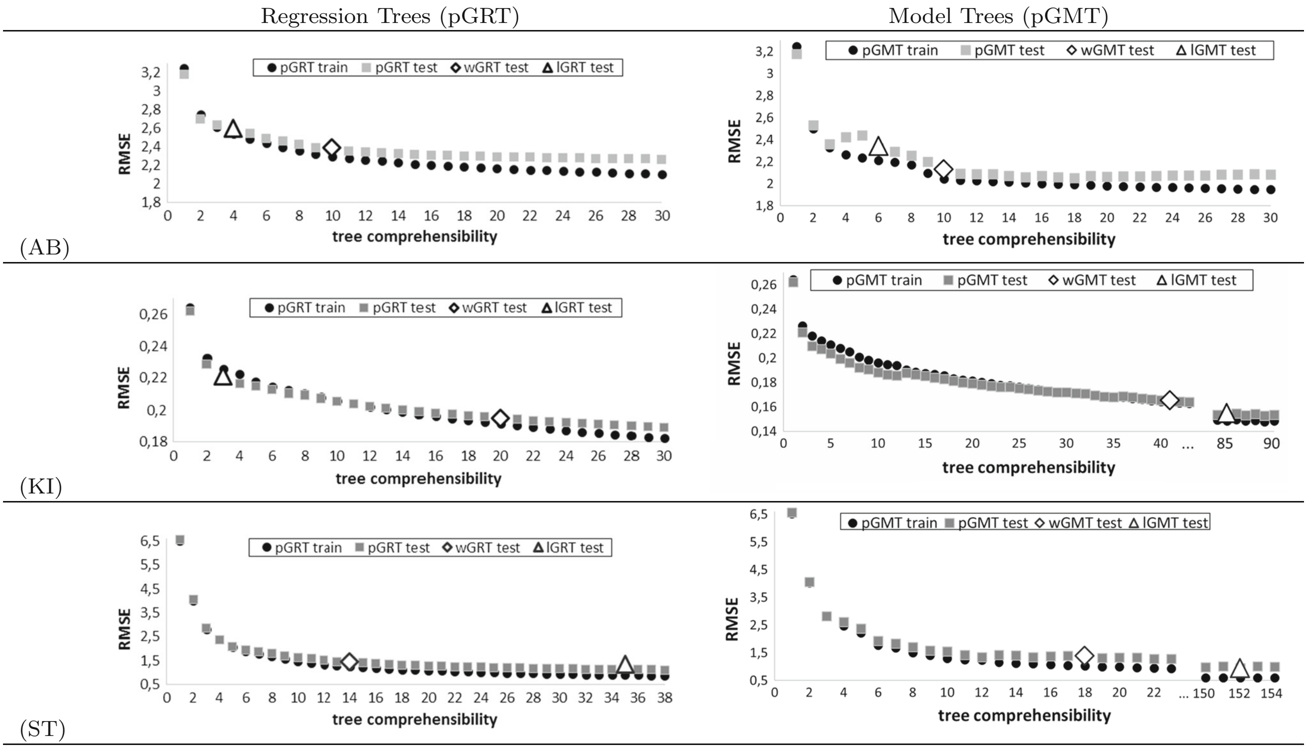


Fig. 3 Pareto front for the regression trees pGRT (left) and model trees pGMT (right) for 2 objectives on training and testing sets of Abalone (AB), Kinematics (KI), and Stock (ST) datasets. Results on testing set for the weight (wGRT/wGMT) and lexicographic (lGRT/lGMT) fitness functions are also enclosed

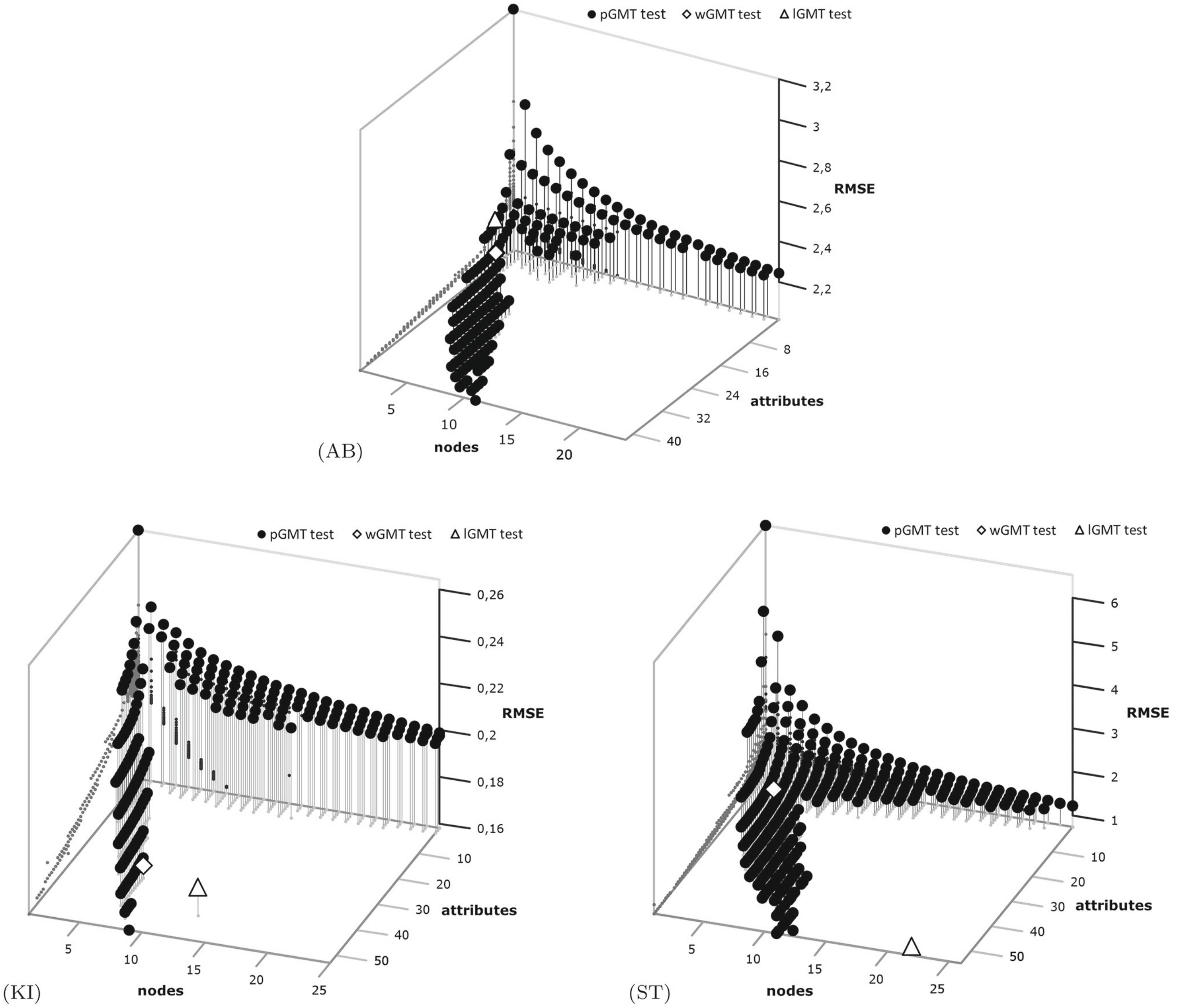


Fig. 4 Pareto front for GMT (pGMT) for 3 objectives on testing set of Abalone (AB), Kinematics (KI), and Stock (ST) datasets. Results on testing set for GMT with weight (wGMT) and lexicographic (lGMT) fitness functions are also enclosed

Figure 3 shows the results achieved for the GMT system with different fitness functions for regression (pGRT) and model (pGMT) trees. The Pareto front was achieved for bi-objective optimization problem that minimized the RMSE and the tree comprehensibility.

One can observe that for the tested datasets, the GMT system with weight (wGRT/wGMT) or lexicographic (lGRT/lGMT) fitness functions managed to find non-dominated solutions, as they belong to the Pareto front. However, open question is if the induced trees will satisfy the decision maker. In case of the results for Abalone dataset (Fig. 3(AB)), both weight and lexicographic fitness function managed to find simple regression and model trees with decent prediction performance. However, if the analyst wants to have slightly more accurate predictor, he might select trees with higher number of nodes/attributes. Opposite situation is for the Kinematics (KI), and Stock (ST) datasets where the algorithms have found accurate but relatively complex predictors which could be difficult to analyze and interpret. Although the trade-off between prediction performance and tree comprehensibility can be partially managed by ad hoc settings of the complexity term in weight fitness function and thresholds in lexicographic analysis, there is no guarantee that found solutions will belong to the Pareto front. With the proposed Pareto-based approach, the decision maker can easily balance between the tree prediction performance and its comprehensibility, depending on the purpose of the analysts goals.

The Pareto front for three-objective optimization problem can be considered only for the model trees and is illustrated in

Table 2 Performance results for evolutionary induced regression trees with different fitness functions as well as popular greedy counterpart of GMT

Algorithm	Parameter	Dataset						
		AB	AI	DA	DE	KI	PO	ST
REP Tree	RMSE	2.223	0.000203	0.000175	0.00150	0.194	8.26	1.469
REP Tree	nodes	291	93	251	229	819	223	137
REP Tree unpruned	RMSE	2.526	0.000221	0.000186	0.00171	0.2031	8.0773	1.1157
REP Tree unpruned	Nodes	526	2277	1047	1545	2301	469	261
wGRT	RMSE	2.387	0.000207	0.000180	0.00155	0.195	8.16	1.431
wGRT	Nodes	9.6	32	13	14	20	58	14
lGRT	RMSE	2.596	0.000243	0.000195	0.00166	0.223	12.10	1.339
lGRT	Nodes	3.5	11	4.0	4.9	2.8	19	35
pGRT 1*	RMSE	2.700	0.000273	0.000188	0.00160	0.216	10.96	1.938
pGRT 1*	Nodes	2.0	4.0	8.0	6.0	4.0	15	6.0
pGRT 2*	RMSE	2.389	0.000209	0.000178	0.00151	0.199	8.555	1.393
pGRT 2*	Nodes	9.0	29	46	14	25	40	15
pGRT 3*	RMSE	2.273	0.000195	0.000173	0.00149	0.179	7.984	0.995
pGRT 3*	Nodes	25	73	66	45	61	57	98

Results for three possible solutions from the Pareto front (denoted as pGRT *) are also included

Table 3 Performance results for evolutionary induced model trees with different fitness functions as well as popular greedy counterpart of GMT

Algorithm	Parameter	AB	AI	DA	DE	KI	PO	ST
M5	RMSE	2.122	0.000169	0.000170	0.00148	0.162	7.4908	0.937
M5	Nodes	12	9.0	17	2.0	106	193	47
M5	Attributes	96	149	85	10	848	1568	423
M5 unprunned	RMSE	2.151	0.000181	0.000190	0.00150	0.1578	7.5095	1.002
M5 unprunned	Nodes	1113	2676	1789	2438	2304	511	196
M5 unprunned	Attributes	9134	34941	22182	8715	15118	4627	907
wGMT	RMSE	2.127	0.000165	0.000173	0.00148	0.163	8.076	1.386
wGMT	Nodes	2.0	3.0	4.6	1.0	7.1	37	3.9
wGMT	Attributes	7.9	17	9.5	4.0	34	52	14
lGMT	RMSE	2.341	0.000177	0.000181	0.00142	0.154	7.314	0.935
lGMT	Nodes	1.0	3.2	4.9	2.0	9.7	78	41
lGMT	Attributes	5.0	11	12	5.2	4.5	82	111
pGMT 1*	RMSE	2.359	0.000175	0.000175	0.00142	0.184	16.90	1.531
pGMT 1*	Nodes	1.0	1.0	1.0	1.0	2.0	3.0	3.0
pGMT 1*	Attributes	2.0	6.0	3.0	5.0	13	6.0	8.0
pGMT 2*	RMSE	2.102	0.000168	0.000169	0.00140	0.174	9.791	0.928
pGMT 2*	Nodes	2.0	2.0	4.0	3.0	4.0	15	31
pGMT 2*	Attributes	9.0	12	14	10	25	23	38
pGMT 3*	RMSE	2.079	0.000164	0.000167	0.00138	0.149	7.354	0.782
pGMT 3*	Nodes	3.0	4.0	9.0	6.0	17	20	61
pGMT 3*	Attributes	12	27	29	24	116	157	121

Results for three possible solutions from the Pareto front (denoted as pGMT *) are also included

Fig. 4. Three-objective optimization enables obtaining much more possible variants of the output trees. For three tested datasets, one can see a trend that either induced trees are small but with large number of attributes, either large but with smaller number of the attributes. It should be noticed that in almost all cases, more compact trees (trees with smaller number of internal nodes but more complex models in the leaves) have higher prediction performance (smaller RMSE)

than larger ones but with simpler models in the leaves. Such large trees would not appear on the Pareto fronts illustrated in Fig. 3 where the number of internal nodes and the attributes in models in summed under one objective called the tree comprehensibility. In addition, we can also observe that the lGMT algorithm for Kinematics and Stock datasets finds solutions that do not belong to the Pareto front. The reason may be the priorities of the objectives and the thresholds settings.

4.3 Comparison with other classifiers

Tables 2 and 3 illustrate the results for evolutionary induced regression and model trees with different fitness functions as well as popular greedy counterparts of GMT:

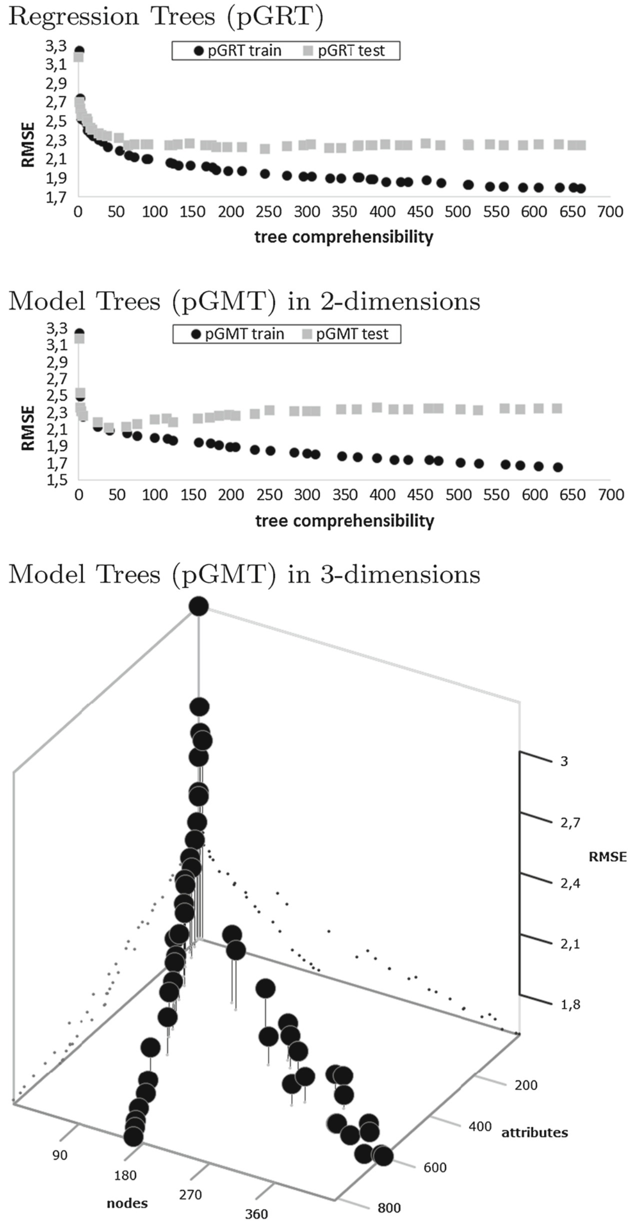
– REP Tree (REP)—popular top-down inducer that builds a regression tree using variance and prunes it using reduced-error pruning (with backfitting);
– M5—state-of-the-art model tree inducer (Quinlan 1992), the most adequate greedy counterpart of the GMT.

For all datasets, three metrics are shown: RMSE on the testing set, number of nodes, and number of attributes in regression models located in the leaves. In case of algorithms with evolutionary induction of DT, the results correspond to averages of 100 runs.

Table 2 shows the results for the regression trees. One can observe that GRT variants induce much more comprehensible predictions with smaller number of nodes, which was also noticed in Czajkowski and Kretowski (2014). The trees induced by the REP Tree are very large; however, on a few datasets (AB, AI) the algorithm managed to achieve low prediction error. With the proposed Pareto approach, there is no need to seek for the trade-off between the prediction performance and the tree size as all possible fronts of non-dominated solutions are visible. Three examples of possible predictors from the Pareto front are also included in Table 2.

Results for the model trees that are shown in Table 3 are analogical to ones for the regression trees. Again, the greedy counterpart (M5) induced overgrown trees with complex regression models in the leaves. Alike with the REP Tree algorithm, M5 managed to achieve small prediction error on a few datasets (PO, ST). It should be noticed that the solutions found by the greedy inducers hardly ever belong to the GMT Pareto front. As for the results of GRT and GMT with weight formula (wGRT/wGRT) and lexicographic analysis (lGRT/lGMT), most of the predictors occurred on the Pareto front.

In addition, we have tested for all datasets both: regression tree (REP Tree) and model tree (M5) without pruning. In both cases, trees were extremely large with often higher RMSE error. Due to over-fitting and the greedy approach, the results for unpruned trees did not coincide with the minimum RMSE solutions on the testing sets.


Regression Trees (pGRT)

Model Trees (pGMT) in 2-dimensions

Model Trees (pGMT) in 3-dimensions

Fig. 5 Complete population-size Pareto front for the regression trees pGRT and model trees pGMT with 2 and 3 objectives on training set of Abalone (AB) dataset

4.4 Fitting Pareto front based on analytical preferences

Fitting the Pareto front based on analytical preferences is not an easy task. In previously performed experiments, pGRT and pGMT systems used full list of non-dominated solutions that were stored in the elitist archive. We understand, however, that in some cases analysts may prefer smaller size of

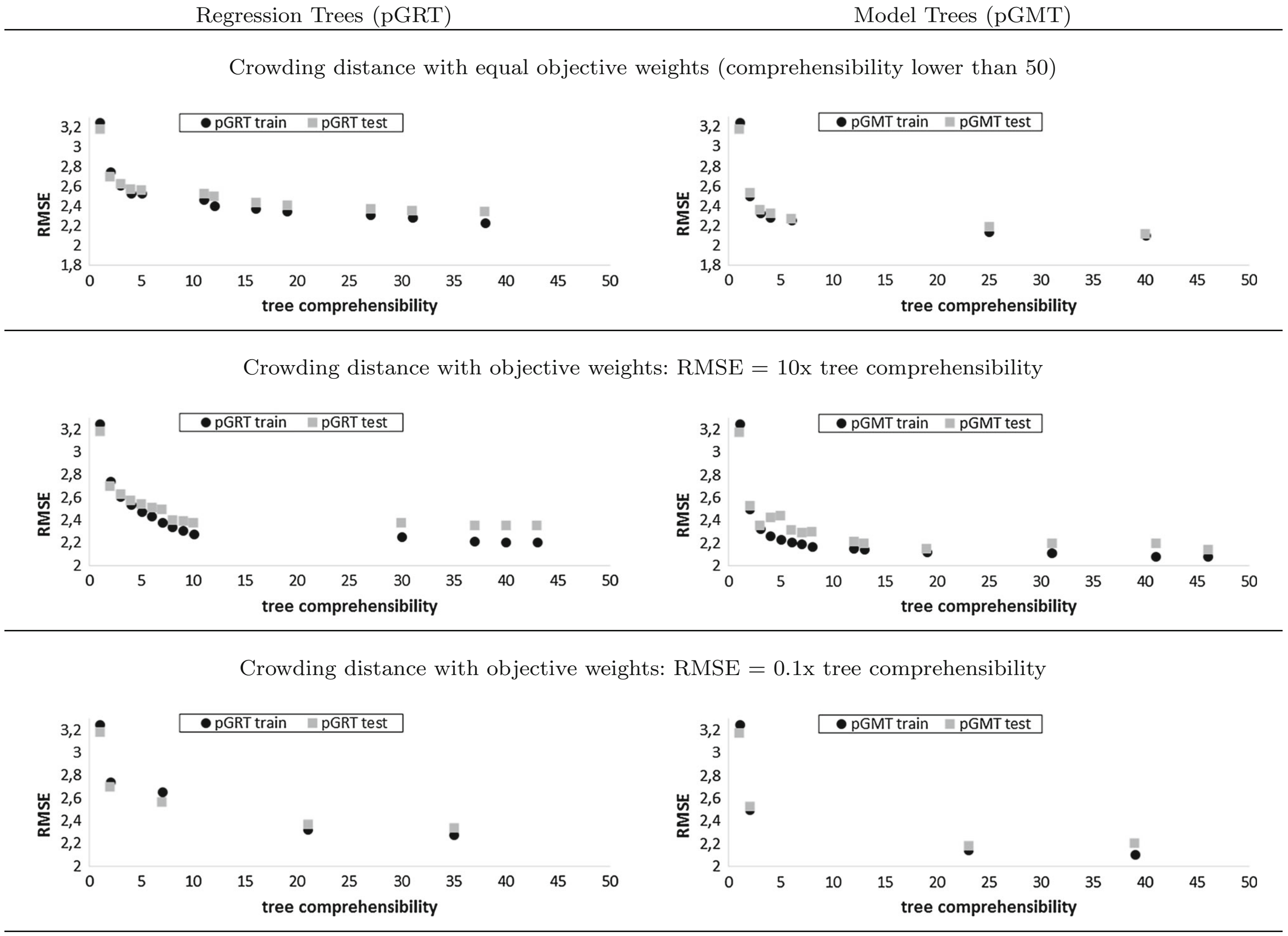


Fig. 6 Impact of the crowding distance ranking (D+A) on the Pareto front for the regression trees pGRT (left) and model trees pGMT (right) for 2 objectives on training and testing set of Abalone (AB) dataset

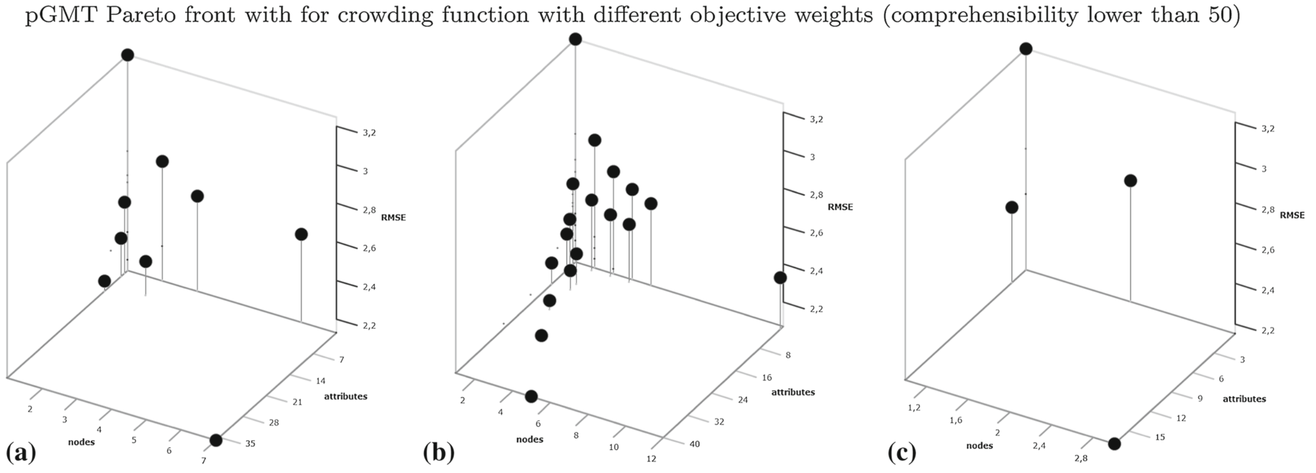


Fig. 7 Impact of the crowding distance ranking (D+A) on the Pareto front for the model trees pGMT for 3 objectives on training and testing set of Abalone (AB) dataset

the Pareto front. Therefore, in this set of experiments we have tested the population-size Pareto front which is also used in, e.g., NSGA-II algorithm.

The crowding distance has a significant impact on the final form of the Pareto front. Therefore, in this section we discuss how to change the crowding function to meet the analysts preferences. In all performed experiments, Abalone (AB) dataset is used for the illustration purposes. Figure 5 shows the Pareto fronts generated for pGRT and pGMT with 2 and 3 objectives. We can observe that for all charts the non-dominant solutions are well arranged according the tree comprehensibility objective thanks to the updated crowding distance procedure (Fortin and Parizeau 2013) (crowding distance ranking: (A)). However, in the real-life problems such almost uniform distribution of the Pareto front may not be desired by the analysts.

It is important to remember that one of the strengths of the decision trees lies in their interpretability which is simply lost when the tree size is too large. On the other side, larger and more complex trees are often more accurate. Therefore, in order to focus on interesting from the analysts point of view regions in the objective space the weights of the objectives need to be introduced (crowding distance ranking: (D+A)). Figures 6 and 7 illustrate the impact of weights in the crowding distance on the Pareto front of regression and model trees. For the illustration purposes, in both figures the tree comprehensibility which is calculated as the sum of the tree size and the number of attributes in the model leaves (in case of model trees) is limited to 50. From Figs. 6 and 7, one can observe that the objective weights have a major impact on the number of solutions that appear in the Pareto fronts. The results are consistent for all tested algorithms: pGRT, pGMT with 2 objectives, and pGMT with 3 objectives. Setting higher weight for the objective connected with the prediction error increases the number of solutions that appear in the Pareto fronts (see Figs. 6 and 7) in comparison with the neutral objective weights settings (A). Such behavior can be explained by the fact that for the small decision trees the prediction error may significantly differ. Opposite situation can be noticed when the higher weights are set for the tree comprehensibility. We can observe that the number of solutions in the Pareto front with comprehensibility lower than 50 is strongly decreased. It is because the region of interests is focused on larger, more complex trees where the changes in RMSE are usually much smaller.

Finally, we want to share some of the results for the concept of the pGMT Pareto front for crowding function based on weight formula described in Sect. 3.2.3 (crowding distance ranking: (C)). Figure 8 shows the Pareto fronts for pGRT and both pGMTs solutions with tree comprehensibility up to 50 (nodes + attributes). We can observe that most of the solutions gathered in the neighborhood of the best individual that could be found using the weight fitness function. Almost all solutions fall under the tree comprehensibility limitation; however, we can observe that predictors with small tree complexity were omitted. Setting crowding distance to the weight formula's value is an interesting proposition to those analysts that want to find Pareto front in a particular region, e.g., near preferred value. In addition, a hybrid solution that would combine both objective weights and weight formula in the crowding distance seems also an interesting idea (crowding distance ranking: (D+B) and (D+C)). Omitted predictors with small tree complexity (see Fig. 8) could be included with the RMSE weight increase. This way, the analysts can have additional control on the final view of the Pareto front.

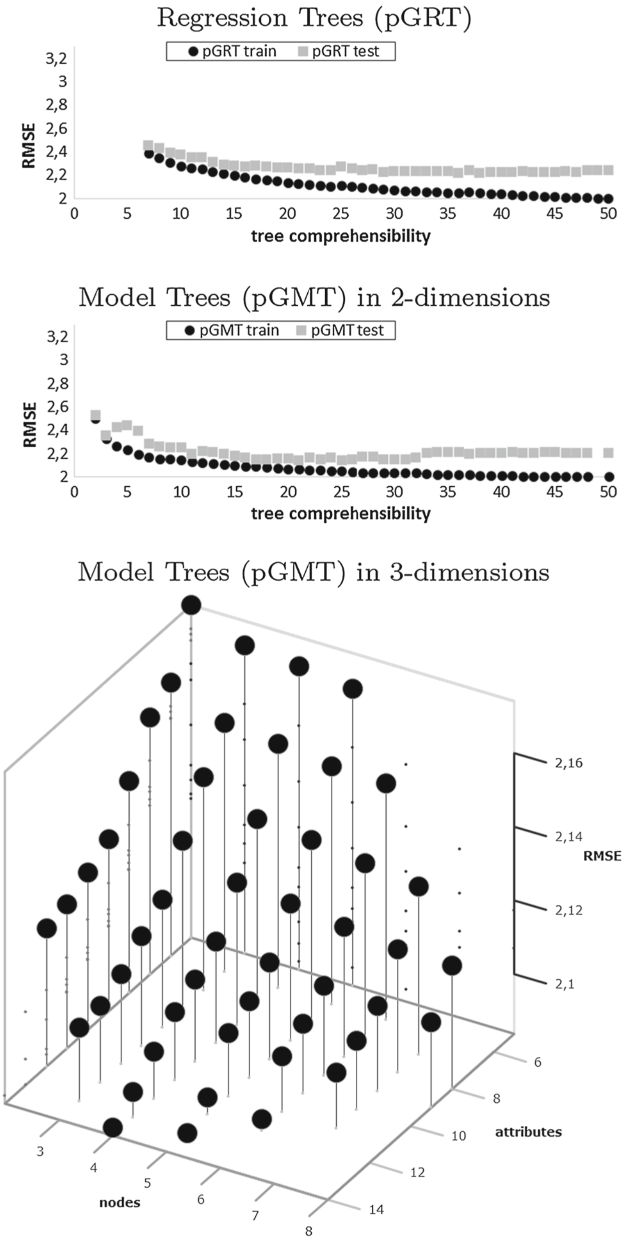


Fig. 8 The pGMT Pareto front for crowding function based on the BIC weight formula (crowding distance ranking: (C))

5 Conclusion and future works

In the paper, we discuss a multi-objective fitness function to evolutionary induced decision trees. Our approach covers the evolutionary induced regression trees and the model trees with two or three objectives that relate to tree error and tree comprehensibility. Performed experiments show that our solution is capable of finding Pareto front for the GMT framework. This is a first step toward searching for efficient and easy to analyze Pareto front for the regression and model trees that can be adjusted based on user preferences.

We see many promising directions for the future research. The proposed approach increases the calculation time of each evolutionary loop and may affect the convergence of EA. Additional efficiency improvements, especially in context of storing and preprocessing full list of non-dominated solutions, need to be considered. Performance issue may also be partially mitigated with parallelization of GMT with, e.g., MPI-OpenMP (Czajkowski et al. 2015), GPGPU (Jurczuk et al. 2017), or Apache Spark (Reska et al. 2018) approaches. In addition, we are constantly working on further comprehensibility improvement of the generated Pareto front and plan to extend our research to cover all types of the decision trees.

Acknowledgements This project was funded by the Polish National Science Center and allocated on the basis of decision 2013/09/N/ST6/04083. The second author was supported by the grant S/WI/2/18 from Bialystok University of Technology founded by Ministry of Science and Higher Education.

Compliance with ethical standards

Conflict of Interest The authors declare that they have no conflict of interest.

Human and animal participants This article does not contain any studies with human participants or animals performed by any of the authors.

References

Afsari F, Eftekhari M, Eslami E, Woo PY (2013) Interpretability-based fuzzy decision tree classifier a hybrid of the subtractive clustering and the multi-objective evolutionary algorithm. Soft Comput. 17:1673–1686

Barros RC, Ruiz DD, Basgalupp M (2011) Evolutionary model trees for handling continuous classes in machine learning. Inf Sci 181(5):954–971

Barros RC, Basgalupp MP, Carvalho AC, Freitas AA (2012) A survey of evolutionary algorithms for decision-tree induction. IEEE Trans SMC Part C 42(3):291–312

Barros RC, Carvalho AC, Freitas AA (2015) Automatic design of decision-tree induction algorithms. Springer, Berlin

Boryczka U, Kozak J (2015) Enhancing the effectiveness of ant colony decision tree algorithms by co-learning. Appl Soft Comput 30:166–178

Breiman L, Friedman J, Olshen R, Stone C (1984) Classification and regression trees. Wadsworth Int Group, Davidson

Czajkowski M, Kretowski M (2014) Evolutionary induction of global model trees with specialized operators and memetic extensions. Inf Sci 288:153–173

Czajkowski M, Kretowski M (2016a) The role of decision tree representation in regression problems—an evolutionary perspective. Appl Soft Comput 48:458–475

Czajkowski M, Kretowski M (2016b) Multi-objective evolutionary approach to Pareto optimal model trees. A preliminary study. In: Proceedings of the TPNC'16. LNCS 10071, pp 85–96

Czajkowski M, Czerwonka M, Kretowski M (2015a) Cost-sensitive global model trees applied to loan charge-off forecasting. Decis Support Syst 74:57–66

Czajkowski M, Jurczuk K, Kretowski M (2015b) Parallel approach for evolutionary induced decision trees. MPI+OpenMP implementation. In: Proceedings of the ICAISC 2015. LNCS 9119, pp 340–349

Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evol Comput 6(2):182–197

Esposito F, Malerba D, Semeraro G (1997) A comparative analysis of methods for pruning decision trees. IEEE Trans PAMI 19(5):476–491

Fakhari A, Moghadam AME (2013) Combination of classification and regression in decision tree for multi-labeling image annotation and retrieval. Appl Soft Comput 13(2):1292–1302

Fan G, Gray JB (2005) Regression tree analysis using TARGET. J Comput Graph Stat 14(1):206–218

Fayyad U, Piatetsky-Shapiro G, Smyth P, Uthurusamy R (1996) Advances in knowledge discovery and data mining. AAAI Press, Palo Alto

Fortin FA, Parizeau M (2013) Revisiting the NSGA-II crowding-distance computation. In: Proceedings of the 15th annual conference on genetic and evolutionary computation. GECCO '13, pp 623–630

Friedrich T, Kroeger T, Neumann F (2013) Weighted preferences in evolutionary multi-objective optimization. Int J Mach Learn Cybern 4(2):139–148

Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH (2009) The WEKA data mining software: an update. SIGKDD Explor 11(1):10–18

Hazan A, Ramirez R, Maestre E, Perez A, Pertusa A (2006) Modelling expressive performance: a regression tree approach based on strongly typed genetic programming. Appl Evol Comput LNCS 3907:676–687

Hiwa S, Nishioka M, Hiroyasu T, Miki M (2015) Novel search scheme for multiobjective evolutionary algorithms to obtain well-approximated and widely spread Pareto solutions. Swarm Evol Comput 22:30–46

Ishibuchi H, Murata T (1998) A multi-objective genetic local search algorithm and its application to flowshop scheduling. IEEE Trans SMC Part C 28(3):392–403

Jin Y, Sendhoff B (2008) Pareto-based multiobjective machine learning: an overview and case studies. IEEE Trans SMC Part C 38(3):397–415

Jurczuk K, Czajkowski M, Kretowski M (2017) Evolutionary induction of a decision tree for large scale data. A GPU-based approach. Soft Comput 21:7363–7379

Kim D (2004) Structural risk minimization on decision trees using an evolutionary multiobjective optimization. LNCS 3003:338–348

Kotsiantis SB (2013) Decision trees: a recent overview. Artif Intell Rev 39:261–283

Liu J, Sui C, Deng D, Wang J, Feng B, Liu W, Wu C (2016) Representing conditional preference by boosted regression trees for recommendation. Inf Sci 327:1–20

Loh W (2014) Fifty years of classification and regression trees. Int Stat Rev 83(3):329–348

Malerba D, Esposito F, Ceci M, Appice A (2004) Top-down induction of model trees with regression and splitting nodes. IEEE Trans PAMI 26(5):612–625

Michalewicz Z (1996) Genetic algorithms + data structures = evolution programs. 3^{rd}, ed. edn. Springer, Berlin

Ortuno FM, Valenzuela O et al (2015) Comparing different machine learning and mathematical regression models to evaluate multiple sequence alignments. Neurocomputing 164:123–136

Pangilinan J, Janssens G (2011) Pareto-optimality of oblique decision trees from evolutionary algorithms. J Glob Optim 51(2):301–311

Pappalardo M (2008) Multiobjective optimization: a brief overview. Springer Optim Appl 17:517–528

Potgieter G, Engelbrecht A (2007) Genetic algorithms for the structural optimisation of learned polynomial expressions. Appl Math Comput 186(2):1441–1466

Potgieter G, Engelbrecht A (2008) Evolving model trees for mining data sets with continuous-valued classes. Expert Syst Appl 35(4):1513–1532

Quinlan J (1992) Learning with continuous classes. In: Proceedings of the AI'92. World Scientific, Singapore, pp 343–348

Reska D, Jurczuk K, Kretowski M (2018) Evolutionary induction of classification trees on Spark. In: Proceedings of the ICAISC 2018. LNCS 10841, pp 514–523

Rokach L, Maimon OZ (2005) Top-down induction of decision trees classifiers—a survey. IEEE Trans SMC Part C 35(4):476–487

Rokach L, Maimon OZ (2008) Data mining with decision trees: theory and application. Machine perception artificial intelligence. World Scientific, Singapore, p 69

Schwarz G (1978) Estimating the dimension of a model. Ann Stat 6:461–464

Torgo L (2017) Regression DataSets repository. http://www.dcc.fc.up.pt/~ltorgo/Regression/DataSets.html. Accessed 21 Nov 2018

Zhang X, Tian Y, Cheng R, Jin Y (2015) An efficient approach to non-dominated sorting for evolutionary multiobjective optimization. IEEE Trans Evol Comput 19(2):201–213

Zhao H (2007) A multi-objective genetic programming approach to developing Pareto optimal decision trees. Dec Support Syst 43(3):809–826

Zitzler E, Thiele L (1999) Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. IEEE Trans Evol Comput 3(4):257–271

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.