

A Method for Determining Decision Tree Complexities Based on Multiobjective Optimization for Online Learning Community

Liang Jia , Nigang Sun *

School of Information Science & Engineering, Chang Zhou University, China.
 ngsun@cczu.edu.cn

This paper focuses on finding minimal complexities of decision trees representing rules defined for features and functionalities of online learning community. Because simultaneously minimizing all complexities is not always possible and this matches pattern of multiobjective optimization, Pareto optimal point is introduced for describing complexity distributions for selecting decision tree most suitable for system requirements. Unlike typical multiobjective optimization whose feasible space can be explicitly represented, feasible space for complexities of decision trees can only be implicitly reflected by relationships of subtables and associated objective functions due to the great number of possible trees. This paper provides a means of finding Pareto optimal points in implicit feasible space determined by objective functions defined for describing decision tree complexities by using graph and associated algorithm. The experimental results show that proposed algorithm yields valid results for finding Pareto optimal points.

1. Introduction

As smart phones, tablets and various types of digital mobile devices prevail in daily life, the internet expands through Wi-Fi, 4G and other mobile telecommunications standards compatible with mobile devices. Our daily lives are dramatically changed in several aspects such as learning as mobile devices flourished in past few years. With the help of new technologies, learning evolves from traditional blackboard and chalks to interactive online activities through the platforms like online learning communities. As far as mobile web can stretch, learning can occur at anytime, anywhere and with anybody by using online learning communities. Online learning communities now provide a convenient means for common people to interact with professional experts, teachers, technicians and other people of advanced knowledge and skills of some expertise. For learning itself, online learning communities promote not only knowledge learning but also idea sharing. However, an efficient solution of online learning community for providing multifarious functionalities is difficult to be implemented. For example, massive information presented in online learning communities involved with a great number of users is quite costly and difficult to classify under the considerations of what topics of users like and other features shared by certain users in online learning communities.

As an efficient and viable means for implementing automatic decision making under the sophisticated cases, decision trees are widely adapted in multifarious fields as digital image processing (Elsalamony, 2014; Grana et al., 2012), biomedicine (Czajkowski et al. 2014), firm power capacity scheduling (Moutis and Hatziaargyriou, 2014), stock analysis (Bast et al, 2015), earnings management (Chen et al., 2015), etc. These successful applications imply decision tree is a highly-efficient solution for implementing functionalities of online learning communities, e.g., information categorization. The focuses are how to construct decision trees, and more important which decision tree is most feasible for online learning communities. The most properties of decision trees need to be taken into account are time and spacial complexities of trees which determine efficiencies of functionalities depending on trees. Because minimization of both time and spacial complexities is always not attainable, this introduces concept of Pareto set which actually enumerates all Pareto optimal solutions for some objective constrains (Erfania and Utyuzhnikov, 2011; Ghosh and Chakraborty, 2014). Although there are multiple algorithms developed for finding Pareto set theoretically, a few targets decision trees (Chikalov, 2013; Hussain, 2014).

This paper presents an algorithm for determining Pareto set of time and spacial complexities of decision trees derived from decision table bearing the features of online learning communities. The time and spacial

complexities of decision tree are defined and employed as objective functions of multiobjective optimization whose desired solutions form Pareto set. The details of algorithm are visually described by UML activity diagrams (Rumbaugh et al., 2010). The algorithm essentially depends on adapting data structure of graph for preserving decision table splitting whose splitting information is employed for generating Pareto set in turn. Some experimental data is adapted for illustrating visual result of Pareto optimal points found in objective space. However, algorithm avoids constructions of decision trees corresponding to Pareto points.

The rest of paper is organized as following. Section 2 provides a brief review about Pareto set and decision tree. Section 3 defines decision tree complexities and mathematical model for generating Pareto optimal points for a given decision table. Section 4 describes algorithms implementing models discussed in Section 3 in details. Section 5 introduces three experimental results for showing validity of proposed algorithm and Section 6 draws the conclusion.

2. Related works

Constructions of decision trees based on given decision tables are constantly evolving. There are plenty of means for constructing decision trees with respect to terminal vertices and height of the tree. Different strategies are employed for decision tree constructions of different kinds of decision tables, e.g., single-valued decision tables (Amin et al., 2013; Chikalov, 2013; Hussain, 2014; Luo et al, 2015) and multi-valued tables (Azad et al., 2013; Azad and Moshkov, 2014). Construction algorithms roughly match some design pattern like dynamic programming (Azad and Moshkov, 2014), incremental algorithm (Luo et al, 2015) and greedy algorithm (Amin et al., 2013; Azad et al., 2013), etc. Normally, almost all of proposed algorithms attempt to construct decision trees of minimal height and minimal left number which are referred as time and spacial complexities. Commonly, such attempts end with compromise of optimizing only one complexity because minimization of two is not always possible. This essentially resembles the case of multiobjective optimization in which several objective functions need to be minimized, but the ideal case of minimizing all objective functions simultaneously is not possible. Hence, suboptimal solutions which minimize objective functions in some degree are derived as Pareto set. There are greatly number and types of Pareto set generation algorithms depending on actual problem patterns, e.g., utility function method (Zeng et al., 2012), global criterion method (Rao, 2009), nonsmooth convex objective function method (Attouch et al., 2015), lexicographic method (Pulido et al., 2014), goal programming method (Rao, 2009), goal attainment method (Rao, 2009), etc.

Formally, k constrains of multiobjective optimization are formalized as functions g_1, g_2, \dots, g_k , there are n objective functions F_1, F_2, \dots, F_n and their values form objective space $Z \subseteq \mathbb{R}^n$. Vectors compatible with F_i where $i = 1, 2, \dots, n$ form decision space $\mathcal{D} \subseteq \mathbb{R}^m$ and the set of vectors in \mathcal{D} satisfying g_j where $j = 1, 2, \dots, k$ is called feasible space and denoted by $\mathcal{D}^* \subseteq \mathcal{D}$. Hence, there is a map $\mathcal{F}: \mathbb{R}^n \rightarrow \mathbb{R}^m$ from \mathcal{D}^* to a subset $Z^* \subseteq Z$, i.e., image of \mathcal{D}^* under \mathcal{F} . For a given vector $\mathbb{x} \in \mathcal{D}^*$, the definition of multiobjective optimization is given as $\min_{\mathbb{x} \in \mathcal{D}^*} \mathcal{F}(\mathbb{x}) = \min_{\mathbb{x} \in \mathcal{D}^*} (F_1(\mathbb{x}), F_2(\mathbb{x}), \dots, F_n(\mathbb{x}))$ where $\mathbb{x}^* \in \mathcal{D}^*$ is Pareto optimal for $\min_{\mathbb{x} \in \mathcal{D}^*} \mathcal{F}(\mathbb{x})$ iff $\nexists \mathbb{x}'$ such that $F_i(\mathbb{x}') \leq F_i(\mathbb{x}^*)$ for $i = 1, 2, \dots, n$ and $\exists j, 1 \leq j \leq n, F_j(\mathbb{x}') < F_j(\mathbb{x}^*)$. Minimum of F_i in Z is denoted by F_i^* , a common assumption of $\min_{\mathbb{x} \in \mathcal{D}^*} \mathcal{F}(\mathbb{x})$ is that there is no $\mathbb{x}' \in \mathcal{D}^*$ satisfying $(F_1(\mathbb{x}'), F_2(\mathbb{x}'), \dots, F_n(\mathbb{x}')) = (F_1^*, F_2^*, \dots, F_n^*)$. To understand how Pareto set is related with decision trees, relative definitions of decision tree have to be introduced.

A decision tree is a concise representation of decision table. A decision table T is a two dimensional table of condition attributes $\{f_1, f_2, \dots, f_m\} = E(T)$ which graphically are column headers of T . Row of index i in T contains an array of values as $\{c_{i1}, c_{i2}, \dots, c_{im}\}$ corresponding to $E(T)$ and these values lead to a decision value d_i . Values of n rows in T form a set $\{\{c_{11}, c_{12}, \dots, c_{1m}\}, \{c_{21}, c_{22}, \dots, c_{2m}\} \dots \{c_{n1}, c_{n2}, \dots, c_{nm}\}\} = C(T)$ and the matched n decisions form a set $\{d_1, d_2, \dots, d_n\} = D(T)$ ($|D(T)| = n$ when $d_i \neq d_j, i \neq j; |D(T)| \leq n$, otherwise). Generally, a decision table T has the following structure.

$$T = \begin{bmatrix} f_1 & \dots & f_m & d \\ c_{11} & \dots & c_{1m} & d_1 \\ \vdots & \ddots & \vdots & \vdots \\ c_{n1} & \dots & c_{nm} & d_n \end{bmatrix} \quad (1)$$

For any $d_k \in D(T)$, $k = 1, 2, \dots, n$, there is a $\{c_{k1}, c_{k2}, \dots, c_{km}\} = r_k \in C(T)$. For two $r_{k_1}, r_{k_2} \in C(T)$ and $k_1 \neq k_2$, inequalities $c_{k_1 1} \neq c_{k_2 1}, c_{k_1 2} \neq c_{k_2 2} \dots c_{k_1 m} \neq c_{k_2 m}$ hold for r_{k_1}, r_{k_2} . For a given subset of $E(T)$, i.e., $f_{i_1}, f_{i_2}, \dots, f_{i_t}$ and their values a_1, a_2, \dots, a_t , a subtable θ can be constructed from T . θ has same condition attributes as T , but its rows is selectively chosen from T , i.e., $E(\theta) = E(T)$ and for any $r_j = \{c_{j1}, c_{j2}, \dots, c_{jm}\} \in C(\theta)$ where $j = 1, 2, \dots, |C(\theta)|$, $c_{ji_1} = a_1, c_{ji_2} = a_2 \dots c_{ji_t} = a_t$ hold. θ is also denoted by $T(f_{i_1}, a_1)(f_{i_2}, a_2) \dots (f_{i_t}, a_t)$.

Based on mentioned definitions, time and spacial complexities can be defined and mathematical formulae for modeling generation of Pareto optimal point can finally be given in following sections.

3. Decision tree complexities and generating pareto set

This section comprises two subsections: the first focuses on definitions of time and spacial complexities of decision trees; the second illustrates how Pareto optimal points are derived with respect to objective functions involving decision tree complexities.

3.1 Time and Spatial Complexities of Decision Trees

For any vertex v of a decision tree Γ generated based on a given decision table T , v denotes one condition attribute from $E(T)$, i.e., $v \in E(T)$. The edge connecting just two vertices is marked by condition values from $C(T)$. Notice, any $f_i \in E(T)$ where $i = 1, \dots, |E(T)|$ can be root vertex of Γ under the assumption $c_{ji} \in r_j \in C(T)$ where $j = 1, \dots, |D(T)|$ is processed despite of order of $f_1, f_2, \dots, f_{|E(T)|}$, and ending vertex of Γ can only be decision values from $D(T)$. A path of t steps from root vertex f_i to arbitrary $v \in E(T) \cup D(T)$ begins from f_i , passing vertices $f_{i_1}, f_{i_2}, \dots, f_{i_t}$ through their connected edges marked by values a_1, a_2, \dots, a_t and ends at v . According the vertices and values associated with t -step path, a subtable $T(v)$ can be constructed as following.

$$T(v) = \begin{cases} T & v = f_i \\ T(f_{i_1}, a_1)(f_{i_2}, a_2) \dots (f_{i_t}, a_t) & v \neq f_i \end{cases} \quad (2)$$

For an arbitrary row $r_k \in C(T)$, $k = 1, 2 \dots |C(T)|$, r_k is determined by some decision value $d_k \in D(T)$. In a decision tree $\Gamma_{T(d_k)}$ built on $T(d_k)$, there is a path presents r_k , the length of r_k , $\ell(r_k)$, is defined by the sum of values marked on edges of path from root of $T(d_k)$ to d_k . If $r_k = \{c_{k1}, c_{k2} \dots c_{km}\}$, then $\ell(r_k) \leq \sum_{j=1}^m c_{kj}$. For a decision tree $\Gamma_{T(v)}$ built on $T(v)$, the total length of $T(v)$ is defined as $\mathcal{L}(\Gamma_{T(v)}) = \sum_{k=1}^n \ell(r_k)$. The time complexity of $\Gamma_{T(v)}$ is defined as followings. $\bar{\mathcal{H}}(\Gamma_{T(v)}) = \mathcal{L}(\Gamma_{T(v)})/\mathcal{N}(T(v))$ where $\mathcal{N}(T(v))$ denotes the number of rows in T represented in $T(v)$. Spacial complexity of $\Gamma_{T(v)}$ is defined as terminal vertices of $\Gamma_{T(v)}$, $\mathcal{F}(\Gamma_{T(v)})$.

3.2 Pareto set generation

All condition values associated with $f_i \in E(T)$ in $C(T)$ is denoted by $C(T, f_i)$, i.e., all values in column f_i of T . All decision trees generated based on T form a set $\mathfrak{X}(T)$, and $\beta = \min_{\Gamma \in \mathfrak{X}(T)} \mathcal{L}(\Gamma)$, $\alpha = \min_{\Gamma \in \mathfrak{X}(T)} \mathcal{J}(\Gamma)$, $R_{\max} = \max_{r_k} \sum_{j=1}^{|E(T)|} c_{kj}$, then $\mathcal{R}_{\mathcal{L}, T} = \{\beta, \beta + 1 \dots |C(T)| \cdot R_{\max}\}$, $\mathcal{R}_{\mathcal{J}, T} = \{\alpha, \alpha + 1 \dots |C(T)|\}$ respectively denote ranges of \mathcal{L} and \mathcal{J} . Map $\mathcal{G}_T: \mathcal{L} \rightarrow \mathcal{J}$ and $\mathcal{F}_T: \mathcal{J} \rightarrow \mathcal{L}$ are defined as followings.

$$\mathcal{G}_T(l) = \left\{ \min_{\Gamma} \mathcal{J}(T, \Gamma) \in \mathcal{R}_{\mathcal{J}, T} \mid \Gamma \in \mathfrak{X}(T) \text{ and } \mathcal{L}(T, \Gamma) \leq l \right\} \quad \mathcal{F}_T(t) = \left\{ \min_{\Gamma} \mathcal{L}(T, \Gamma) \in \mathcal{R}_{\mathcal{L}, T} \mid \Gamma \in \mathfrak{X}(T) \text{ and } \mathcal{J}(T, \Gamma) \leq t \right\}$$

Where $l \in \mathcal{R}_{\mathcal{L}, T}$ and $t \in \mathcal{R}_{\mathcal{J}, T}$. Let data structure graph be employed to represent relationships between T and its subtables, then the nodes of graph are subtables $T(v) = T_v$, edges emitting from a node associated with $f_i \in E(T_v)$ are marked by pairs of values (f_i, a_j) where $a_j \in C(T, f_i)$ and pointing to nodes $T_v(f_i, a_j)$. Hence, edges in the graph are directed and connect two tables of parent and child relationship. The graph is thus directed acyclic graph (DAG). For a node $T_v \in \text{DAG}$, it has two possible types determined by number of decision values.

When $|D(T_v)| = 1$ for all rows $r_k \in C(T_v)$, r_k share a common decision value and T_v is leaf node of DAG of $\mathcal{F}_{T_v}(1) = 0$.

When $|D(T_v)| > 1$, $r_k \in C(T_v)$ can be categories based on their different decision values. Starting from node T_v , for any $f_i \in E(T_v)$ and $a_1, a_2, \dots, a_{|C(T_v, f_i)|} \in C(T_v, f_i)$, there are $|C(T_v, f_i)|$ edges marked by $(f_i, a_1), (f_i, a_2), \dots, (f_i, a_{|C(T_v, f_i)|})$ and leading to nodes $T_v(f_i, a_1), T_v(f_i, a_2), \dots, T_v(f_i, a_{|C(T_v, f_i)|})$. For each $T_v(f_i, a_j)$, there is a corresponding $\mathcal{R}_{\mathcal{J}, T_v(f_i, a_j)}$. For $f_i \in E(T_v)$, an ordered set of all possible compositions of $\mathcal{J}(\Gamma_{T_v(f_i, a_j)})$ where $j = 1, \dots, |C(T_v, f_i)|$ is defined as followings.

$$\mathfrak{R}_{\mathcal{J}, T_v, f_i} = \{v_i * v, v_{i+1} * v, \dots, v_{|C(T_v, f_i)|} * v\}$$

Where $v = [1 \dots 1]$, $*$ denotes inner product and $v_k * v < v_{k+1} * v$ where $k = i, i + 1, \dots, |C(T_v, f_i)|$. v_i is defined as followings.

$$v_i = \begin{bmatrix} \min_{\Gamma_1 \in \mathfrak{X}(T_v(f_i, a_1))} \mathcal{J}(\Gamma_{T_v(f_i, a_1)}) \\ \min_{\Gamma_2 \in \mathfrak{X}(T_v(f_i, a_2))} \mathcal{J}(\Gamma_{T_v(f_i, a_2)}) \\ \vdots \\ \min_{\Gamma_t \in \mathfrak{X}(T_v(f_i, a_{|C(T_v, f_i)|}))} \mathcal{J}(\Gamma_{T_v(f_i, a_{|C(T_v, f_i)|})}) \end{bmatrix} \quad \text{and} \quad v_j = \begin{bmatrix} \mathcal{J}(\Gamma_{T_v(f_i, a_1)}) \\ \mathcal{J}(\Gamma_{T_v(f_i, a_2)}) \\ \vdots \\ \mathcal{J}(\Gamma_{T_v(f_i, a_{|C(T_v, f_i)|})}) \end{bmatrix} \quad (3)$$

For $v_j \neq v_i$, $i + 1 \leq j \leq |C(T_v, f_i)|$, $\Gamma_1 \in \mathfrak{X}(T_v(f_i, a_1))$, $\Gamma_2 \in \mathfrak{X}(T_v(f_i, a_2))$, \dots , $\Gamma_t \in \mathfrak{X}(T_v(f_i, a_t))$, v_j is defined as followings. For $T_v(f_i, a_j)$ where $i = 1, \dots, |E(T)|$, value of $\mathcal{F}_{T_v(f_i, a_j)}$ can be derived. For $v_k * v \in \mathfrak{R}_{\mathcal{J}, T_v, f_i}$, objective function $\mathcal{F}_{T_v}^{f_i}$ associated with complexities of decision trees generated around $f_i \in E(T_v)$ is defined as followings.

$$\mathcal{F}_{T_v}^{f_i}(\mathfrak{R}_{\mathcal{J}, T_v, f_i}) = \min_{\mathfrak{v}_k * \mathfrak{v} \in \mathfrak{R}_{\mathcal{J}, T_v, f_i}} \sum_{j=1}^t \mathcal{F}_{T_v}(f_i, a_j)(\mathfrak{v}_k(j)) + |\mathcal{C}(T_v)| \quad (4)$$

Where $\mathfrak{v}_k(j)$ denotes the j th component of vector \mathfrak{v}_k . According to $\mathcal{F}_{T_v}^{f_i}(\mathfrak{v}_k)$, objective function $\mathcal{F}_{T_v}(\mathfrak{R}_{\mathcal{J}, T_v})$ of $E(T_v)$ for describing complexities of subtable T_v is given as followings where $\mathfrak{R}_{\mathcal{J}, T_v} = \{\mathfrak{R}_{\mathcal{J}, T_v, f_{i_1}}, \mathfrak{R}_{\mathcal{J}, T_v, f_{i_2}}, \dots, \mathfrak{R}_{\mathcal{J}, T_v, f_{i_t}}\}$.

$$\mathcal{F}_{T_v}(\mathfrak{R}_{\mathcal{J}, T_v}) = \left\{ \mathcal{F}_{T_v}^{f_i} \mid \min_{\mathfrak{R}_{\mathcal{J}, T_v, f_i} \in \mathfrak{R}_{\mathcal{J}, T_v}} \mathcal{F}_{T_v}^{f_i}(\mathfrak{R}_{\mathcal{J}, T_v, f_i}) \right\} \quad (5)$$

Generally, if T_v is terminal node of graph, then $\mathcal{F}_{T_v}(1) = 0$ and Pareto optimal point is $(1, \mathcal{F}_{T_v}(1))$; if T_v is not a terminal node, we first compute $\mathcal{F}_{T_v}^{f_i}(\mathfrak{R}_{\mathcal{J}, T_v, f_i})$ for each $f_i \in E(T_v)$, then choose the minimal value as $\mathcal{F}_{T_v}(\mathfrak{R}_{\mathcal{J}, T_v})$ and Pareto optimal point is $(\mathfrak{R}_{\mathcal{J}, T_v}^{-1}, \mathcal{F}_{T_v}(\mathfrak{R}_{\mathcal{J}, T_v}))$ where $\mathfrak{R}_{\mathcal{J}, T_v}^{-1}$ is objective function for retrieving $\mathfrak{v}_k * \mathfrak{v} \in \mathfrak{R}_{\mathcal{J}, T_v, f_i}$ which yields \mathcal{F}_{T_v} . For multiobjective optimization of decision tree complexities, objective space is a two-dimensional space whose objective functions are $\mathfrak{R}_{\mathcal{J}, T_v}^{-1}$ and $\mathcal{F}_{T_v}(\mathfrak{R}_{\mathcal{J}, T_v})$ which graphically are two axis of space.

4. Algorithms for generating pareto points

This section describes the procedure of generating Pareto optimal points in details. The general scheme consists of two subroutines, i.e., Subroutine 1: generate a directed acyclic graph starting from T and proceeding downward until degenerate subtables, and Subroutine 2: compute \mathcal{F}_T starting from degenerate subtables in graph and proceeding upward until T . Subsections 4.2 to 4.3 respectively describe the two phases of generating Pareto optimal points.

4.1 Generating DAG based on decision table

This section describes the procedure of generating Pareto optimal points in details. The general scheme is sketched in subsection 4.1. Subsections 4.2 to 4.3 respectively describes the two phases of generating Pareto optimal points.

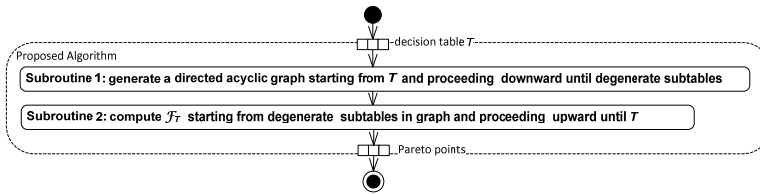


Figure 1: General scheme of proposed algorithm.

4.2 Generating DAG based on decision table

As depicted in Figure 2 left, Subroutine 1 attempts to split a given decision table T in a recursive manner. After adding T to DAG, it tries to find a table T' not split yet from DAG and insert subtables of $T'(f_i, a_j)$ where $f_i \in E(T')$ and $a_j \in C(T, f_i)$ as nodes to DAG. Node T' and all $T'(f_i, a_j)$ are connected by directed edges starting from T' and ending at $T'(f_i, a_j)$. Edges are marked by corresponding condition values (f_i, a_j) . Node insertion only occurs when subtable is not found in DAG and if there is an edge connecting T' and existing $T'(f_i, a_j)$, the mark of edge will be added by (f_i, a_j) rather than adding a new edge. This node insertion and edge modification or addition continues until there is no tables found in DAG which are not split.

4.3 Subroutine 2 generating pareto points

Subroutine 2 provides core functionality for generating Pareto points based on mathematical descriptions made in Section 3. The strategy is starting generating from terminal nodes of DAG whose Pareto points are of form $(1,0)$, then moving upward to subtables T' containing only terminal nodes providing values for $\mathcal{F}_{T'}^{f_i}$ and compute $\mathcal{F}_{T'}$ by adding the smallest $\mathcal{F}_{T'}^{f_i}$ with $|\mathcal{C}(T')|$. Before T is reached, it continues to find the next subtables T' whose subtables have completed their Pareto point computation. For T , each Pareto point attached with $f_i \in E(T)$ is considered as an optimal Pareto point as output of proposed algorithm. The details are illustrated in Figure 2 right.

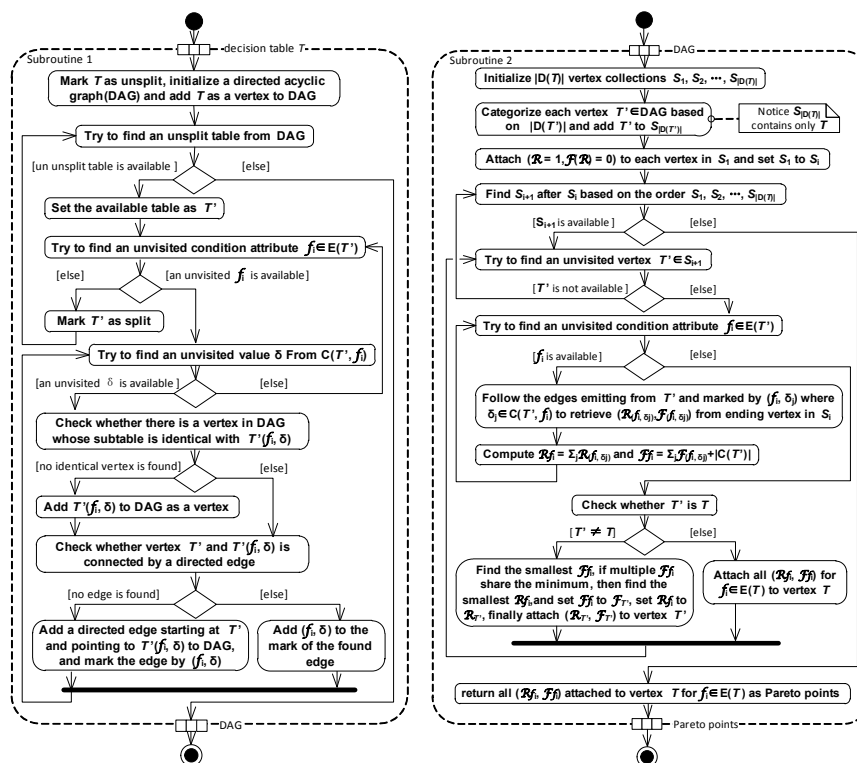


Figure 2: Subroutine 1 and Subroutine 2.

5. Experimental results

This section introduces experimental results of adapting proposed algorithm for decision tables designed for a virtual online learning community. These decision tables define some functionalities of online learning community as rules of categorizing posts based on topics, rules of identifying user of high influence, etc. A typical table has about 6 condition attributes and 50 rows. Pareto points are rendered as dots in a two dimensional space of axes leaf vertex number and F_{f_i} . Three typical results are show in Figure 3

6. Conclusions

This paper proposes mathematical model and corresponding algorithms for generating Pareto optimal points of decision tree complexities for given decision tables. Unlike traditional multiobjective optimization whose feasible space is explicitly described in some means as mathematical formulae, the feasible space of decision tree complexities is implicitly related to underlying decision tables. The implicit feasible space is revealed by employing graph to preserve relationships of subtables. Relationships are represented as nodes and directed edges in graph which lead to minimization of objective functions defined based on concepts of complexities. The validity of proposed algorithm is shown by generating Pareto optimal points of decision tables made for a virtual online learning community.

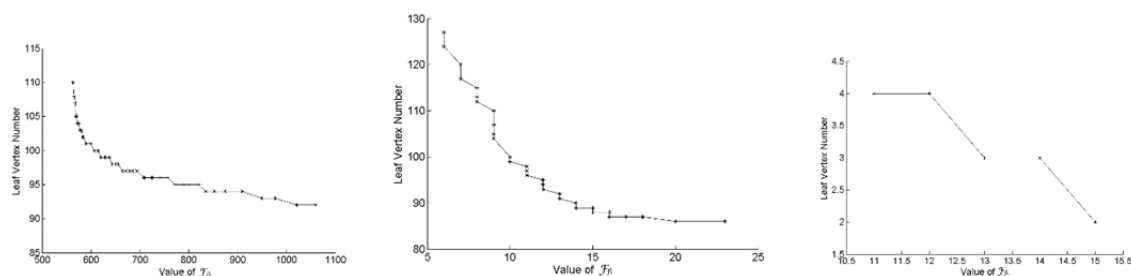


Figure 3: Pareto Set for Table 1 of Virtual Online Learning Community.

References

- Amin T., Chikalov I., Moshkov M., Zielosko B., 2013, Dynamic programming approach to optimization of approximate decision rules, *Inform. Sci.*, 221, 403-418.
- Attouch H., Garrigos G., Goudou X., 2015, A dynamic gradient approach to Pareto optimization with nonsmooth convex objective functions, *J. of Math. Anal. and Applicat.*, 422, 741-771.
- Azad M., Moshkov M., 2014, Minimization of decision tree average depth for decision tables with many-valued decisions, *Procedia Comput. Sci.*, 35, 368-377.
- Azad M., Zielosko B., Moshkov M., Chikalov I., 2013, Decision rules, trees and tests for tables with many-valued decisions—comparative study, *Procedia Comput. Sci.*, 22, 87-94.
- Bastı E., Kuzey C., Delen D., 2015, Analyzing initial public offerings' short-term performance using decision trees and SVMs, *Decision Support Syst.*, 73, 15-27.
- Chen F., Chi D., Wang Y., 2015, Detecting biotechnology industry's earnings management using Bayesian network, principal component analysis, back propagation neural network, and decision tree, *Econ. Modelling*, 46, 1-10.
- Chikalov I., Hussain S., Moshkov M., 2013, Totally optimal decision trees for monotone boolean functions with at most five variables, *Procedia Comput. Sci.*, 22, 359-365.
- Czajkowski M., Grześ M., Kretowski M., , 2014, Multi-test decision tree and its application to microarray data classification, *Artificial Intell. in Medicine*, 61, 35-44.
- Elsalamony H. A., 2014, Detecting distorted and benign blood cells using the Hough transform based on neural networks and decision trees, in *Emerging Trends in Image Process.*, *Comput. Vision and Pattern Recognition*, 1st ed. USA: Morgan Kaufmann.
- Erfania T., Utyuzhnikov V. S., 2011, Directed Search Domain: A method for even generation of the Pareto frontier in multiobjective optimization, *Eng. Optimaztion*, 43, 467-484.
- Ghosh D., Chakraborty D., 2014, A new Pareto set generating method for multiobjective optimization problems, *Operations Research Lett.*, 42, 514-521.
- Grana C., Montangero M., Borghesani D., 2012, Optimal decision trees for local image processing algorithms, *Pattern Recognition Letters*, 33, 2302-2310.
- Hussain S., 2014, Total path length and number of terminal nodes for decision trees, *Procedia Comput. Sci.*, 35, 514-521.
- Luo C., Li T., Chen H., Lu L., 2015, Fast algorithms for computing rough approximations in set-valued decision systems while updating criteria values, *Inform. Sci.*, 299, 221-242.
- Moutis P., Hatzigaryriou D. N., 2014, Decision trees aided scheduling for firm power capacity provision by virtual power plants, *Int. J. of Elect. Power & Energy Syst.*, 63, 730-739.
- Pulido J. F., Mandow L., Cruz P. L. J., 2014, Multiobjective shortest path problems with lexicographic goal-based preferences, *European Journal of Operational Research*, 239, 89-101.
- Rao S. S., 2009, Practical aspects of optimization, in *Engineering Optimization Theory and Practice*, 4th ed., USA: John Wiley & Sons.
- Rumbaugh J., Jacobson I., and Booch G., 2010, *Unified Modeling Language Reference Manual*, 2nd ed. MA, USA: Addison-Wesley.
- Zeng X., Wong W., Leung Y. S., 2012, An operator allocation optimization model for balancing control of the hybrid assembly lines using Pareto utility discrete differential evolution algorithm, *Comput. & Operations Research*, 39, 1145-1159.