



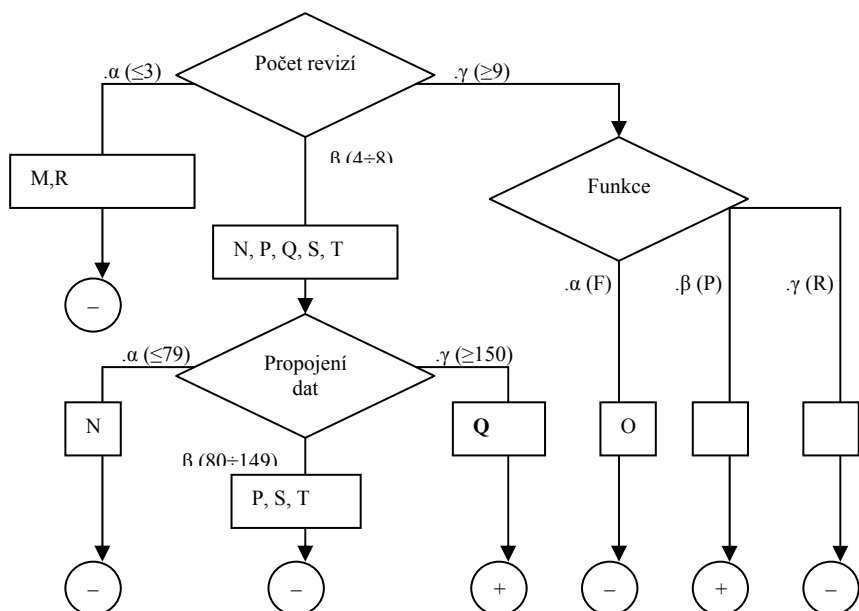
**Abstrakt:** Aplikace klasifikačních a regresních stromů v oblasti zvyšování spolehlivosti softwaru poskytuje především nástroj pro zvýšení efektivity tohoto procesu vhodným výběrem modulů s vysokým očekávaným počtem chyb a odhadem jejich rizikovosti lze snížit náklady na testování. V předchozích pracích jsou popsány nové vyvíjené metody, které usnadní a zlepší práci při tvorbě klasifikačního stromu. Jsou to metody rozšíření základní klasifikační škály na více skupin a metoda vyhledávání optimálních hranic dělení použitých metrik. Přínos těchto metod je ukázán na konečném příkladu. Pro snadnější výpočty byl vyroben program pro hledání optimálních kořenů metrik při konstrukci těchto klasifikačních stromů. Důležitými vstupními údaji jsou metriky, hodnoty metrik, moduly a třídy modulů. Uživatelé jsou umožněni získat mezivýsledky, jako jsou funkce výběru metrik F a konečný výsledek, kterým je nalezení minimálního E sloužícího k rozhodnutí o optimálním kořenu metriky. Nastavení programu je dostatečně variabilní. Program mi velice pomohl při simulacích i výpočtech na skutečných datech.

### KLASIFIKAČNÍ STROMY

Vzeme-li části programu (moduly), které jsme již v minulosti testovali, můžeme z jejich vlastností (metrik) zkonstruovat klasifikační strom, kterým pak můžeme klasifikovat části programu, u kterých chceme zjistit, zda je máme či nemusíme testovat. Tento zkonstruovaný strom nám může pomoci zkrátit dobu testování nebo dokonce zvýšit spolehlivost, protože se při testování zaměříme více na části, které jsou skutečně potřeba otestovat a odladit.

### PŘÍKLAD KLASIFIKAČNÍHO STROMU

Ukážeme názorný příklad klasifikace modulu Q v již vytvořeném stromě. Modul má tyto vlastnosti: počet revizí 7, propojení dat 172 a funkci P (řízení procesu). V prvním kroku jej tedy klasifikujeme do větve β dle metriky počet revizí a následně do větve γ dle metriky propojení dat.



Pomocí klasifikace jsme zjistili, že klasifikovaný modul Q je pravděpodobně náchylný k chybám (skončil v + listu) a měli bychom jej testovat.

### METODIKA TVORBY

- Rozklad na moduly
- Ohodnocení do skupin
- Zavedení, výběr a volba metrik
- Volba hranic
- Vhodné dělení

### OPTIMALIZACE PŘI TVORBĚ KL. STROMU

Při vytváření klasifikačního stromu předem dané (expertní) hranice metrik nemusí být vždy optimální a mohou způsobovat vytvoření stromu se slepými koncovými listy nebo s koncovými listy, které jsou stále nehomogenní přestože jsou již vyčerpány všechny metriky.

### OPTIMALIZACE POČTU HRANIC VÝBĚREM NEJLEPŠÍHO POČTU

Vypočteme stromy pro všechny možné počty hranic a pozorujeme maximum E() pro celý strom, počet použitých metrik a pozorujeme homogenitu v koncových listech (poměr homogenních listů ku všem listům).

- Neprovádíme v případech, kdy máme dva (a více) modulů se stejnými hodnotami všech metrik, ale s odlišnou skupinou (☒, ☒).
- Pro všechny metriky volíme stejný počet hranic. Na počátku tvorby zvolíme jednu hranici a tento počet postupně zvyšujeme.
- Tvorbu stromů ukončíme ihned, jakmile budeme mít v právě prováděném stromu všechny koncové listy homogenními.
- Dodržujeme podmínku, že metrika nemůže mít více hranic, nežli je počet naměřených odlišných hodnot mínus jedna pro tuto metriku. Pokud nastane tato podmínka, pro tuto metriku již počet hranic nezvyšujeme.
- Pokud již nemáme možnost zvýšit počet hranic u žádné z metrik, tvorbu stromů ukončíme.

Na všech získaných stromech spočítáme *použitelnost* stromu jako počet použitých metrik, dělených násobkem E() a *homogenitu* v koncových listech (násobek poměrů vyššího zastoupení skupiny ku všem modulům v listu). Ze všech stromů pak vezmeme ten, který má tuto použitelnost stromu nejvyšší.

### OPTIMALIZACE POČTU HRANIC POSTUPNÝM ZVÝŠOVÁNÍM

Budeme postupně vytvářet klasifikační strom běžným způsobem s proměnným počtem hranic a budeme zaznamenávat maximum E() pro celý strom a pozorovat homogenitu v koncových listech.

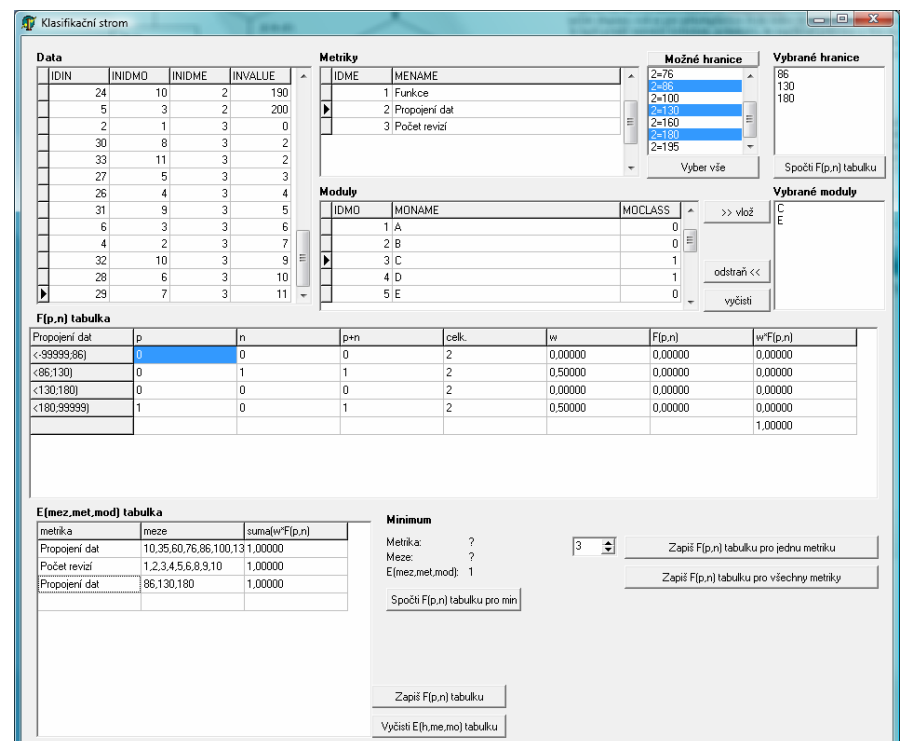
- Neprovádíme v případech, kdy máme dva (a více) modulů se stejnými hodnotami všech metrik, ale s odlišnou skupinou (☒, ☒).
- Na počátku tvorby zvolíme jednu hranici pro všechny metriky.
- Tvorbu stromu ukončíme ihned, jakmile budeme mít všechny koncové listy homogenními.
- Pokud bude existovat nehomogenní koncový list, zvýšíme počet hranic o jedničku tam, kde nám vychází nejvyšší E().
- Pokud bude existovat nehomogenní koncový list a ve stejné metrice jako v bezprostředně předchozí tvorbě stromu se nám objeví maximum E(), které je vyšší nebo rovno maximu při bezprostředně předchozí tvorbě stromu, vrátíme počet hranic v této metrice na bezprostředně předchozí. V této metrice již nikdy počet hranic nezvyšujeme a zvyšujeme počet hranic u metriky, která má druhé (popř. další) nejvyšší E().

- Pokud bude existovat nehomogenní koncový list a ve stejné metrice jako v bezprostředně předchozí tvorbě stromu se nám objeví maximum E(), které je nižší než při bezprostředně předchozí tvorbě stromu, zvýšíme počet hranic o jedničku u této metriky jen tehdy, pokud nám nevyšel stejný strom, jinak vrátíme počet hranic v této metrice na bezprostředně předchozí. V této metrice již nikdy počet hranic nezvyšujeme a zvyšujeme počet hranic u metriky s druhým (popř. dalším) nejvyšším E().
- Pokud při zvyšování počtu hranic v dané metrice narazíme na počet všech možných (odlišných) hodnot pro danou metriku mínus jedna, počet hranic u této metriky nezvyšujeme a zvyšujeme počet hranic u jiné.
- Nemáme-li již možnost zvýšit počet hranic u žádné z metrik, tvorbu stromu ukončíme.

### PROGRAM – METODA PLOVOUCÍCH MEZÍ

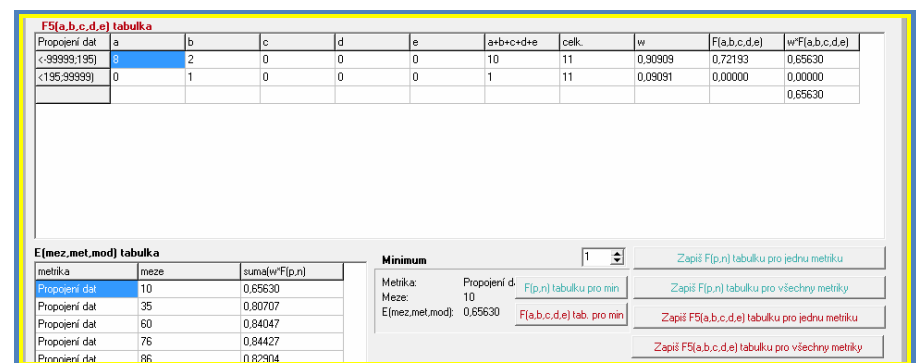
Při vytváření klasifikačního stromu můžeme procházet všechny možné meze (také i počet mezí) a dostaneme menší hodnoty E() a tím i lepší dělení jednotlivých větví stromu. Tuto metodu nazveme metoda plovoucích mezí.

Protože by bylo náročné procházet ručně všechny možné meze a zjišťovat vhodnou metriku a meze vytvořili jsme pomocný program, který propočítává minimum E() přes všechny metriky. Program generuje tyto meze v polovinách mezi všemi naměřenými hodnotami pro danou metriku. Výsledkem (výstupem) programu je vyhledání optimální metriky pro kořen (přes minimum E()). Následně je pak možné vybrat pouze moduly jdoucí určitou větví a znovu určit optimální metriku pro další kořen.



### PROGRAM – ROZŠÍŘENÍ KLASIFIKAČNÍ ŠKÁLY NA 5 SKUPIN

Další verzi pomocného programu pro výpočet je jeho rozšířená verze, která umožňuje zadat klasifikaci modulů v pětistupňové škále. Třídou modulu tedy do programu zadáváme v hodnotách 0,1,2,3,4.



### ZÁVĚR

V příspěvku jsem se zabýval konstrukcí klasifikačních stromů pro použití ve spolehlivosti softwaru. V první části jsem popsal základní konstrukci, která počítá s expertním počtem i hodnotami hranic metrik, v další části jsem popsal vylepšení této základní konstrukce vyhledáním optimálních hranic metrik. V příspěvku jsem také popsal rozšíření základní klasifikační škály na více skupin. Program přinesl ulehčení výpočtů. Bylo řešeno v rámci projektu MŠMT 1M06047 Centrum pro jakost a spolehlivost výroby.

### ODKAZY

- [1] Strouhal J.: Spolehlivostní modely při tvorbě softwaru, Disertační práce 2008
- [2] Dohnal G.: Markovské modely spolehlivosti softwaru, Sborník mezinárodní konference ROBUST'06, JČMF, 2006, ISBN 80-7015-073-4, str. 453-460
- [3] Dohnal G.: Markovské řetězce a jejich aplikace v QC, Analýza dat 2005/II, Trilobyte statistical software, 2006, ISBN 80-239-6552-2, str. 119-124