IMPLEMENTÁCIA HEURISTIKY PRE TVORBU ROZVRHU SLUŽIEB VODIČOV AUTOBUSOV POMOCOU PYFUZZY

LADOVSKÝ, Tomáš, (SK)

Abstrakt. Článok rieši problém rozpisovania služieb vodičom autobusov pomocou fuzzifikovanej heuristiky day-by-day. Heuristika priraďuje turnusy vodičom postupne pre každý deň plánovaného časového obdobia a neuvažuje s následkami rozhodnutí, ktoré vznikli v minulosti. Tento nedostatok heuristiky vniesol do plánovaných rozhodnutí (priradení) neistotu, čo je dôvodom zavedenia fuzzifikácie. Neistotu článok modeluje pomocou fuzzy inferenčného systému a implementuje ju pomocou balíčka pyFuzzy.

1 Úvod

Problém rozpisovania služieb vodičom autobusov (The Bus Driver Rostering Problem – DRP) pozostáva zo zobrazenia množiny vodičov do množiny turnusov pre každý deň danej časovej periódy. Tento problém je v článku definovaný s ohľadom na viacero právnych predpisov, ktorými stanovené podmienky musia byť dodržané. Všetkým vodičom musíme priradiť turnus alebo deň voľna a to pre každý deň plánovaného obdobia. Preto pre každého vodiča musí tvorca rozpisu služieb vyhotoviť postupnosť turnusov a dní voľna pre uvažovanú časovú periódu. V článku je DRP obmedzovaný viacerými právnymi predpismi [1] a pravidlami spoločností pomocou nasledujúcich *ťažkých podmienok*:

(h1) každý turnus musí byť pridelený maximálne jednému vodičovi;
(h2) každý vodič môže dostať pridelený maximálne jeden turnus z vopred zvolenej podmnožiny množiny všetkých turnusov (pracovné dni, víkend, sviatky, leto, zima) alebo deň voľna;
(h3) každý vodič musí odpočívať minimálne 11 hodín medzi dvoma nasledujúcimi turnusmi.

Nazývajú sa ťažkými podmienkami, pretože sa pri vytváraní rozpisu služieb nesmú porušiť. Sú predpísané právom, zmluvami a pravidlami autobusovej spoločnosti. S ohľadom na

ťažké podmienky dobré rozpisy služieb obvykle obsahujú malé diferencie medzi celkovými rozvrhnutými pracovnými hodinami každého vodiča. Ďalej je dobrý rozpis služieb charakterizovaný tým, že vodičom sa opakujú rovnaké alebo podobné turnusy vzhľadom na trasu a/alebo obtiažnosť. V článku sa zaoberáme dvoma charakteristikami dobrého rozpisu služieb (*ľahké podmienky*) a to:

(s1) zrovnomerňovaním celkových kumulovaných denných pracovných časov vodičov za celé časové obdobie rozpisovania služieb;
(s2) maximalizáciou frekvencie opakovania rovnakých alebo podobných turnusov.

V ľahkých podmienkach sú zahrnuté ciele rozpisovania služieb a tie optimalizujeme. Na riešenie problému je zvolená heuristika day-by-day [2], ktorá najskôr nájde priradenia medzi vodičmi a turnusmi pre prvý deň plánovaného časového obdobia, potom pre druhý, atď. Hlavným jej nedostatkom je, že neuvažuje s následkami rozhodnutí, ktoré vznikli v minulosti.

2 Matematická formulácia

Rozpisovanie služieb vodičom autobusov znamená, že pre každý deň z množiny dní $\mathscr{D} = \{1,2,\ldots,n\}$ sa priradí vodičovi V_i z množiny vodičov $\mathscr{V} = \{V_1,V_2,\ldots,V_m\}$ taký turnus T_k z množiny turnusov $\mathscr{T} = \{T_1,T_2,\ldots,T_t\}$ alebo deň voľna, aby nedošlo k porušeniu žiadnej ťažkej podmienky počas optimalizovania ľahkých podmienok. Parameter n označuje počet dní, parameter m udáva počet vodičov a parameter t udáva počet turnusov. **Turnus** $T_k = (z_k, e_k, c_k)$ je definovaný usporiadanou trojicou nasledovných parametrov: časom začatia z_k; časom ukončenia e_k a denným pracovným časom vodiča c_k turnusu $T_k \in \mathscr{T}$. **Rozpis služieb** $R = (x_{ijk})$ modelujeme pomocou binárnej premennej x_{ijk} nasledovne: $x_{ijk} = 1$ ak i-ty vodič má pridelený v j-ty deň k-ty turnus; inak je $x_{ijk} = 0$ pre $(V_i, j, T_k) \in \mathscr{V} \times \mathscr{D} \times \mathscr{T}$. Táto premenná vyjadruje rozhodnutia tvorcu rozpisov služieb pri prideľovaní turnusov vodičom za celé časové obdobie rozpisovania služieb. Nech s je aktuálny deň heuristiky day-by-day, potom **rozpis služieb v aktuálnom dni** modelujeme premennou y_{ik} následovne: $y_{ik} = 1$ ak i-ty vodič má pridelený v aktuálny deň k-ty turnus; inak je $y_{ik} = 0$ pre $(V_i, T_k) \in \mathscr{V} \times \mathscr{T}$.

Algoritmus day-by-day

K1:	Inicializuj aktuálny deň na prvý deň plánovaného obdobia, $s := 1$.
K2:	Pomocou vopred zvolenej *metódy rozpisovania služieb v aktuálny deň* získaj optimálne riešenie rozpisu služieb v aktuálnom dni Y^*.
K3:	Pre daný aktuálny deň s aktualizuj rozpis služieb R, teda $x_{isk} := y^*_{ik}, (V_i, T_k) \in \mathscr{V} \times \mathscr{T}$.
K4:	Ak je aktuálny deň rovný poslednému dňu obdobia rozpisovania služieb ($s = n$) potom koniec, R je rozpisom služieb na celé plánované obdobie, inak prejdi na nasledujúci deň ($s := s+1$) a goto K2.

Metóda rozpisovania služieb v aktuálny deň je postup riešenia problému rozpisovania služieb v aktuálny deň. Riešením metódy rozpisovania služieb v aktuálny deň je optimálny rozpis služieb v aktuálnom dni $Y^* = (y^*_{ik}), (V_i, T_k) \in \mathscr{V} \times \mathscr{T}$.

3 Implementovanie heuristiky day-by-day

Fuzzifikovaná heuristika je implementovaná v programovacom jazyku *Python* [4]. Dôvodom jeho výberu je množstvo balíčkov, rýchle prispôsobenie programátora k Pythonu, open source, množstvo dokumentácie a jednoduché paralelne programovanie. V dnešnej dobe existujú pre Python dva balíčky, ktoré implementujú fuzzy množiny a vykonávajú na nich operácie fuzzy logiky. Sú to *pyFuzzyLib* a *pyFuzzy*. Balíček *pyFuzzy* je zvolený z nasledujúcich dôvodov: jednoduché programovanie fuzzy množín a fuzzy pravidiel (v zdrojovom kóde alebo súbor flc); možnosť zobrazovania vstupov a výstupov graficky; možnosť grafického zobrazenia fuzzy pravidiel; množstvo príkladov. Fuzzy inferenčný systém implementovaný v pyFuzzy je typu MIMO (Multiple Inputs Multiple Outputs). Programátorovi ponúka pyFuzzy veľký výber zo vstupných a výstupných druhov fuzzy čísiel (množín). Ďalej má programátor možnosť výberu z rôznych noriem a konoriem, ktoré môže použiť pri tvorbe pravidiel, agregácií a/alebo konfigurovania defuzzifikácie. V závislosti od ich výberu je programátor schopný vytvoriť rôzne inferenčné metódy (Mamdami, Larsen, [3]). Náš fuzzy inferenčný systém pozostáva z dvoch vstupov, piatich pravidiel a jedného výstupu (MISO – Multiple Inputs Single Output). Nasledujúce kroky vedú k vytvoreniu fuzzy inferenčného systému.

3.1 Vytvorenie fuzzy inferenčného systému

Kód inicializácie systému zapíšeme v Pythone takto:

```
import fuzzy.System
system = fuzzy.System.System()
```

Trieda *fuzzy.System* koordinuje celý fuzzy systém (bázu dát, bázu pravidiel, fuzzifikačné rozhranie, defuzzifikačné rozhranie).

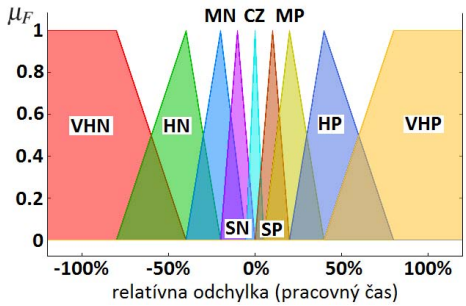
3.2 Definovanie bázy dát fuzzy množín vstupov

Nech s-ty deň mesiaca ($1 \leqq s \leqq 28$) je aktuálnym dňom fuzzifikovanej heuristiky day-by-day. Predpokladáme, že i-ty vodič by mal dostať pridelený k-ty turnus, ktorý začína v s-tý deň. Na základe tohoto predpokladu vieme spočítať nasledujúce relatívne odchýlky:

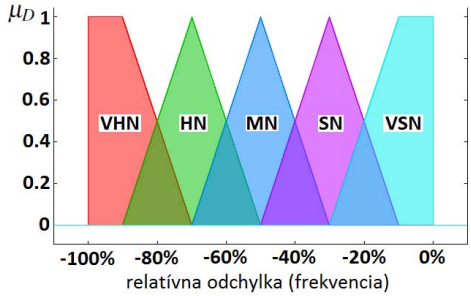
- *Relatívna odchýlka reálneho od ideálneho denného pracovného času* $d_{ik}(s,R)$. Táto relatívna odchýlka môže byť pozitívna alebo negatívna.
- *Relatívna odchýlka reálnej od ideálnej frekvencie* $f_{ik}(s,R)$ podobných turnusov. Táto relatívna odchýlka môže byť iba negatívna.

Pre tvorcu rozpisu služieb je hodnota $d_{ik}(s,R)$ ($f_{ik}(s,R)$) veľmi dôležitá. Čím je hodnota $d_{ik}(s,R)$ bližšie k nule ($f_{ik}(s,R)$ väčšia), tým je väčšie uprednostnenie tvorcu rozpisu služieb, aby priradil i-tému vodičovi k-tý turnus na začiatku s-tého dňa.

Nech D je fuzzy lingvistická premenná vyjadrujúca relatívnu odchýlku reálneho denného pracovného času od ideálneho, ktorá nadobúda nasledujúce hodnoty: **VHN** – veľmi veľká



(a) Funkcie príslušnosti fuzzy množín VHN, HN, MN, SN, CZ, SP, MP, HP, VHP; popisujúce relatívnu odchýlku reálneho denného pracovného času od ideálneho



(b) Funkcie príslušnosti fuzzy množín VHN, HN, MN, SN, VSN; popisujúce relatívnu odchýlku reálnej od ideálnej frekvencie podobných turnusov

Obrázok 1: Funkcie príslušnosti fuzzy množín vstupov; lingvistických premenných D a F

negatívna relatívna odchýlka; **HN** - veľká negatívna relatívna odchýlka; **MN** - stredná negatívna relatívna odchýlka; **SN** – malá negatívna relatívna odchýlka; **CZ** – relatívna odchýlka blízko nule; **SP** – malá pozitívna relatívna odchýlka; **MP** – stredná pozitívna relatívna odchýlka; **HP** – veľká pozitívna relatívna odchýlka; **VHP** – veľmi veľká pozitívna relatívna odchýlka. Na obrázku 1(a) sú zobrazené jednotlivé funkcie príslušnosti k predošlým fuzzy množinám.

Nech F je fuzzy lingvistická premenná vyjadrujúca relatívnu odchýlku reálnej frekvencie od ideálnej, ktorá nadobúda nasledujúce hodnoty: **VHN** – veľmi veľká negatívna relatívna odchýlka, **HN** - veľká negatívna relatívna odchýlka, **MN** – priemerná negatívna relatívna odchýlka, **SN** – malá negatívna relatívna odchýlka, **VSN** – veľmi malá negatívna relatívna odchýlka. Na obrázku 1(b) sú zobrazené jednotlivé funkcie príslušnosti k predošlým fuzzy množinám. Kód predošlých fuzzy množín vstupov zapíšeme v Pythone nasledujúco:

```
from fuzzy.InputVariable import InputVariable
from fuzzy.Adjective import Adjective
from fuzzy.set.Polygon import Polygon
from fuzzy.set.Triangle import Triangle
from fuzzy.fuzzify.Plain import Plain
irregularity = InputVariable(fuzzify = Plain(), min = -100, max = 100)
system.variables["irregularity"] = irregularity
irregularity.adjectives['VHN']=Adjective(Polygon([(-100,1),(-80,1),(-40,0)]))
irregularity.adjectives['HN']=Adjective(Polygon([(-80,0),(-40,1),(-20,0)]))
irregularity.adjectives['MN']=Adjective(Polygon([(-40,0),(-20,1),(-5,0)]))
irregularity.adjectives['SN']=Adjective(Polygon([(-20,0),(-10,1),(0,0)]))
irregularity.adjectives['CZ']=Adjective(Triangle(0,5,5))
irregularity.adjectives['SP']=Adjective(Polygon([(0,0),(10,1),(20,0)]))
irregularity.adjectives['MP']=Adjective(Polygon([(5,0),(20,1),(40,0)]))
irregularity.adjectives['HP']=Adjective(Polygon([(20,0),(40,1),(80,0)]))
irregularity.adjectives['VHP']=Adjective(Polygon([(40,0),(80,1),(100,1)]))
```

Podobne píšeme kód aj pre fuzzy premennú „frequency“. Trieda *fuzzy.InputVariable* implementujúca vstupnú fuzzy premennú, pozostáva z niekoľkých fuzzy hodnôt, ktoré implementuje trieda *fuzzy.Adjective*. V našom prípade to môže byť napríklad hodnota „VHN“ fuzzy premennej „relatívna odchýlka reálneho denného pracovného času od ideálneho“.

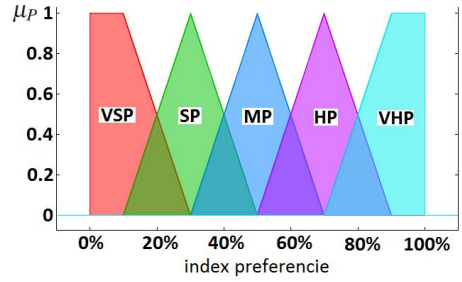
Balík *fuzzy.set* obsahuje triedy implementujúce fuzzy množiny, ktoré majú napríklad tvar trojuholníka `Triangle()`, lichobežníka `Trapez()`, ostrej hodnoty `Singleton()`, polygónu `Polygon()` a veľa iných. Balík *fuzzy.fuzzify* pozostáva z tried, ktoré slúžia na určenie stupňa, ktorým jednotlivé vstupy prislúchajú fuzzy množinám definovaných funkciami príslušnosti. Tieto triedy sa starajú o zobrazenie ostrej hodnoty do prislúchajúcej fuzzy množiny.

3.3 Definovanie bázy dát fuzzy množín výstupov

Nech P je fuzzy lingvistická premenná vyjadrujúca silu uprednostňovania (preferencie) tvorcu rozpisu služieb, ktorá nadobúda nasledujúce hodnoty: **VSP** – veľmi slabé uprednostňovanie, **SP** – slabé uprednostňovanie, **MP** – stredne silné uprednostňovanie, **HP** – silné uprednostňovanie, **VHP** – veľmi silné uprednostňovanie. Kód predošlých fuzzy množín výstupu zapíšeme v Pythone takto:

```
from fuzzy.norm.Min import Min
from fuzzy.norm.Max import Max
from fuzzy.defuzzify.COG import COG
from fuzzy.OutputVariable import OutputVariable
INF, ACC = Min(), Max()
COG = COG(INF=INF, ACC=ACC, failsafe = 0., segment_size=0.5)
preferen = OutputVariable(defuzzify = COG, min = 0, max = 100)
system.variables['preference'] = preferen
preferen.adjectives['VSP']=Adjective(Polygon([(0,0),(0,1),(10,1),(30,0)]))
preferen.adjectives['SP']=Adjective(Polygon([(10,0),(30,1),(50,0)]))
preferen.adjectives['MP']=Adjective(Polygon([(30,0),(50,1),(70,0)]))
preferen.adjectives['HP']=Adjective(Polygon([(50,0),(70,1),(90,0)]))
preferen.adjectives['VHP']=Adjective(Polygon([(70,0),(90,1),(100,1),(100,0)]))
```

Index preferencie $p_{ik}(s,R)$ vyjadruje uprednostňovanie tvorcu rozvrhu, či priradí i-tému vodičovi k-ty turnus, ktorý začína v s-tý deň. Nech $0 \leqq p_{ik}(s,R) \leqq 1$, $V_i \in V, T_k \in T, s \in D$. Čím je index uprednostnenia väčší, tým je tvorca rozpisu služieb presvedčenejší, že dôjde k priraďovaniu nad uvažovanými vodičmi a turnusmi v aktuálny deň. Na obrázku 2 sú zobrazené jednotlivé funkcie príslušnosti fuzzy množín popisujúcich silu uprednostňovania tvorcu rozpisu služieb. Balík *fuzzy.norm* pozostáva z rôznych t-noriem a t-konoriem, ktoré sú implementované triedami. Napr. operátory maxima, minima, algebraické sčítanie alebo algebraické násobenie. Tieto operátory sa používajú pri vyhodnocovaní pravidiel a pri agregácií, teda zjednotení výstupov každého pravidla. Balík *fuzzy.defuzzify* pozostáva z niekoľkých tried, ktoré vykonávajú proces defuzzifikácie. Ide o opačný proces k fuzzifikácii, teda triedy zobrazujú fuzzy množiny na ostré hodnoty. Najznámejšie sú COG (Center of Gravity) a COGS (Center of Gravity for Singletons). Trieda *fuzzy.OutputVariable*



Obrázok 2: Funkcie príslušnosti fuzzy množín VSP, SP, MP, HP, VHP určujúce index preferencie

implementuje výstupnú fuzzy premennú. Pri jej inicializácii treba určiť metódu defuzzifikácie. Pozostáva z niekoľkých fuzzy hodnôt.

3.4 Definovanie bázy pravidiel

Teraz definujme bázu pravidiel. Vieme, že s je aktuálny deň a R je rozpis služieb. Taktiež vieme, že relatívna odchýlka reálneho denného pracovného času od ideálneho je značená ako $d_{ik}(s,R)$. Ďalej relatívna odchýlka reálnej od ideálnej frekvencie podobných turnusov je značená ako $f_{ik}(s,R)$ a index preferencie je značený ako $p_{ik}(s,R)$. Kvôli čitateľnosti zapisujeme v nasledujúcom texte $d_{ik}(s,R), f_{ik}(s,R)$ a $p_{ik}(s,R)$ iba ako d, f a p.

Báza pravidiel

R1:	IF d is {VHN,HN,MN,SN,SP,MP,HP,VHP} AND f is VHN THEN p is VSP
R2:	IF (d is CZ AND f is VHN) OR (d is {VHN,HN,MN,SN,SP,MP,HP,VHP} AND f is HN) OR (d is {VHN,HN,HP,VHP} AND f is MN) OR (d is {VHN,VHP} AND f is SN) THEN p is SP
R3:	IF (d is CZ AND f is HN) OR (d is {MN,SN,SP,MP} AND f is MN) OR (d is {HN,HP} AND f is SN) OR (d is {VHN,VHP} AND f is VSN) THEN p is MP
R4:	IF (d is CZ AND f is MN) OR (d is {MN,SN,SP,MP} AND f is SN) OR (d is {MN,SN,SP,MP} AND f is VSN) THEN p is HP
R5:	IF (d is CZ AND f is SN) OR (d is {SN,CZ,SP} AND f is VSN) THEN p is VHP

Predošlú tabuľku pravidiel zapíšeme v Pythone takto (iba prvé pravidlo):

```
from fuzzy.Rule import Rule
from fuzzy.operator.Compound import Compound
from fuzzy.operator.Input import Input
OR, AND = Max(), Min()
system.rules['preference index is VSP'] = Rule(
    adjective = preferen.adjectives["VSP"],
    operator=Compound(OR,
        Compound(AND,
            Compound(OR,
                Input(system.variables["irregularity"].adjectives["SN"]),
                Input(system.variables["irregularity"].adjectives["MN"]),
                Input(system.variables["irregularity"].adjectives["HN"]),
                Input(system.variables["irregularity"].adjectives["VHN"])
            ),
            Input(system.variables["frequency"].adjectives["VHN"])
        ),
        Compound(AND,
            Compound(OR,
                Input(system.variables["irregularity"].adjectives["SP"]),
                Input(system.variables["irregularity"].adjectives["MP"]),
                Input(system.variables["irregularity"].adjectives["HP"]),
                Input(system.variables["irregularity"].adjectives["VHP"])
            ),
            Input(system.variables["frequency"].adjectives["VHN"])
        )
    ),CER=CER
)
```

Podobne píšeme kód aj pre zostávajúce pravidlá R2 až R5. Trieda *fuzzy.Rule* implementuje bázu pravidiel. Balík *fuzzy.operator* pozostáva z operátorov, ktoré vytvárajú bázu pravidiel.

Napríklad trieda `Compound()` slúži na skladanie zložitejších pravidiel pomocou noriem a konoriem. Trieda `Input()` slúži na zviazanie fuzzy hodnoty k uvažovanému vstupu (tvorba výroku).

3.5 Vykreslenie grafov

Príkaz `createDoc(system)` vykresľuje grafy fuzzy premenných vstupov (irregularity, frequency) a výstupov (index preferencie) zadefinovaných v báze dát. Príkaz `create3DPlot()` vykresľuje 3D obrázok riešenia výstupu FIS vzhľadom na vstupy.

```
from fuzzy.doc.plot.gnuplot import doc
d = doc.Doc()
d.createDoc(system)
d.create3DPlot(system,"irregularity","frequency","preference")
```

Balíček *fuzzy.doc.plot.gnuplot* je potrebný na vykreslenie vstupných fuzzy množín a výstupných fuzzy premenných a to v 2D a 3D prevedení. Na vykreslenie je použitý program gnuplot [7] cez rozhranie Gnuplot.py.

3.6 Získanie výstupnej hodnoty

Inštancia vstupných hodnôt relatívnej odchýlky pracovných kumulovaných časov od ideálnych je napríklad rovná hodnote $irr = 13$ a relatívna odchýlka frekvencie opakovania podobných turnusov je rovná hodnote $freq = -54$ (priradenie jedného vodiča a jeden turnus). Výslednú hodnotu indexu preferencie spočítame pomocou príkazu `system.calculate(...)`. Ta naplní slovník "'out"' novou hodnotou výstupu.

```
irr, freq, out = 13, -54, {"preference" : 0}
system.calculate(input={'irregularity':irr,'frequency':freq},output=out)
print output
```

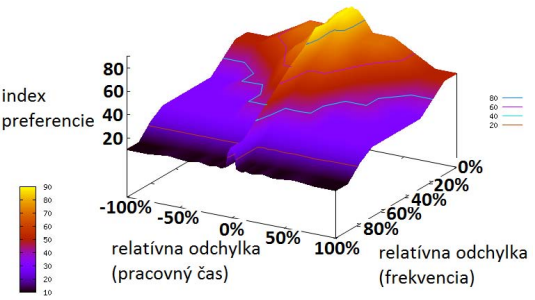
Výstupnou hodnotou FIS systému je ostrá hodnota indexu preferencie y (output). Defuzzifikáciu robíme pomocou najčastejšie používanej metódy 'ťažiska' (Center of gravity – COG). Teraz vieme spočítať index preferencie pre i-teho vodiča a k-ty turnus. Nasledujúci algoritmus počíta maticu indexu preferencie $P = (p_{ik})_{m\times t}$ pre všetkých vodičov a turnusy z daných množín $\mathscr{V}$ a $\mathscr{T}$. Nech s je aktuálny deň a R je rozpis služieb.

Algoritmus výpočtu matice indexu preferencie (ceny) $P = (p_{ik})_{m\times t}$

K1:	Inicializuj index vodiča $i := 1$ a index turnusu $k := 1$.
K2:	Ak je $(i = m)$ a $(k = t)$ potom **STOP** inak goto **K3**.
K3:	Pre hodnoty vstupu $x_1 := d_{ik}(s,R)$ a $x_2 := f_{ik}(s,R)$ spočítaj pomocou navrhnutého fuzzy inferenčného systému hodnotu výstupu y a zapíš ju do matice $p_{ik}(s,R) := y$. Goto **K4**.
K4:	Ak $(k = t)$ potom $i := i+1$ a $k := 0$ inak $k := k+1$. Goto **K2**.

Obrázok 3 zobrazuje výstupnú hodnotu fuzzy inferenčného systému pre každého vodiča a turnus. V podstate ide o 3D-graf matice $P = (p_{ik})_{m\times t}$.

Nech u_{ik} je binárny parameter rozpisovania služieb, ktorý je definovaný nasledovne: $u_{ik} = 1$ ak i-ty vodič môže dostať pridelený k-ty turnus; inak $u_{ik} = 0$ pre $(V_i, T_k) \in \mathscr{V} \times \mathscr{T}$. Tento parameter definuje tvorcovi rozpisu služieb, či môže danému vodičovi priradiť určitý turnus. Taktiež môžeme týmto parametrom definovať náročnosť turnusov, senioritu, atď. Nech v_i je binárny parameter rozpisovania služieb, ktorý je definovaný nasledovne: $v_i = 1$ ak i-ty vodič môže pracovať v aktuálny deň; inak $v_i = 0$ pre $V_i \in \mathscr{V}$. Tento parameter slúži na modelovanie požadovaných dní voľna zo strany vodičov, plánované školenia zo strany zamestnávateľa, zdravotnú starostlivosť a iné. Nech w_k je binárny parameter rozpisovania služieb, ktorý modeluje rozhodnutia: $w_k = 1$ ak musíme odjazdiť k-ty turnus v aktuálny deň; inak $w_k = 0$ pre $T_k \in \mathscr{T}$. Nie každý turnus sa musí odjazdiť v daný deň. Napr. turnusy pre voľné dni sa nesmú odjazdiť cez týždeň.



Obrázok 3: Správanie sa fuzzy inferenčného systému vzhľadom na jeho vstupy

Nech s je aktuálnym dňom heuristiky a r_i je čas ukončenia priradeného turnusu i-teho vodiča v $(s-1)$-vý deň a ďalej nech platí, že ak $\left(\frac{z_k+1440-r_i}{660} < 1\right)$ alebo $(u_{ik} = 0)$ potom $b_{ik} = -\infty$ inak $b_{ik} = p_{ik}(s,R)$. Pomocou tejto implikácie zaručujeme splnenie ťažkej podmienky (h3). Teraz keď poznáme ceny priraďovacej úlohy b_{ik}, môžeme spočítať zovšeobecnenú priraďovaciu úlohu priradenia turnusov vodičom pre uvažovaný aktuálny deň. Výsledkom je optimálne riešenie rozpisu služieb v aktuálnom dni Y^*, viď. druhý krok heuristiky.

$$\max\{\sum_{V_i \in \mathscr{V}} \sum_{T_k \in \mathscr{T}} b_{ik} : \sum_{V_i \in \mathscr{V}} y_{ik} = w_k, T_k \in \mathscr{T}; \sum_{T_k \in \mathscr{T}} y_{ik} \leqq v_i, V_i \in \mathscr{V}; y_{ik} \geqq 0, (V_i, T_k) \in \mathscr{V} \times \mathscr{T}\}$$

4 Reálna inštancia problému

Pre každý deň z množiny dní $\mathscr{D} = \{1,2,\ldots,28\}$ priraď vodičovi V_i z množiny vodičov $\mathscr{V} = \{V_1, V_2, \ldots, V_{107}\}$ deň voľna alebo turnus T_k z množiny turnusov $\mathscr{T} = \{T_1, T_2, \ldots, T_{179}\}$, pričom nedôjde k porušeniu žiadnej ťažkej podmienky a k maximálnemu zlepšeniu ľahkých podmienok. Časť turnusov T_k je zadaná v tabuľke 1. Turnusy s identifikačným číslom 1 až 107 sú turnusy, ktoré sa majú odjazdiť počas pracovných dní. Turnusy s číslami 108 až 179 sa majú odjazdiť počas víkendov. Vzhľadom na túto skutočnosť sú definované parametre u_{ik}, v_i, w_k pre celé obdobie rozpisovania služieb. Algoritmus day-by-day je napísaný v programovacom jazyku Python (Python Programming Language [4]). Na výpočet priraďovacej úlohy je použité GLPK (GNU Linear Programming Kit [5]). Získavame prijateľný 28 dňový rozpis služieb vodičov autobusov (tabuľka 2), ktorého relatívna odchýlka kumulovaného pracovného času od ideálneho (obrázok 4(a)) sa sústreďuje okolo nuly. Na obrázku 4(b)

Tabuľka 1: Dvadsaťosemdňová inštancia problému rozpisovania služieb

	Počet vodičov	Počet dní	Počet turnusov
	107	28	179
Turnus	Začiatok (min)	Koniec (min)	Pracovný čas (min)
T_1	298	460	239.4
T_2	793	980	227.8
⋮	⋮	⋮	⋮
T_{178}	522	711	219.0
T_{179}	260	700	480.3

Tabuľka 2: Dvadsaťosemdňový rozpis služieb vodičov autobusov

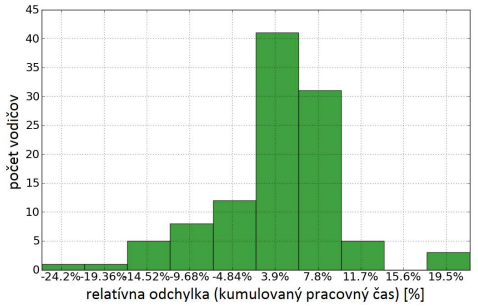
$\mathscr{V}/\mathscr{D}$	1Pon	1Uto	1Str	1Štv	1Pia	1Sob	...	4Sob	4Ned	s_i (min)
V1	T1	T29	T18	T35	T12	T150	...	T147	T154	11400.0
V2	T3	T5	T4	T8	T8	T109	...	T109	T111	10186.0
V3	T2	T11	T8	T26	T5	T110	...	T111	T109	10696.6
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋱	⋮	⋮	⋮
V105	T104	T45	T9	T45	T92	T154	...	T174	Off	9928.5
V106	T107	T94	T95	T67	T52	T152	...	T158	T168	11832.6
V107	T106	T79	T24	T88	T16	Off	...	T138	T159	10414.2

vidíme zobrazenú frekvenciu opakovania sa podobných turnusov pridelených vodičom za celé obdobie rozpisovania služieb. Povšimnime si, že turnusy T_1 až T_{72} majú značné zníženú frekvenciu opakovania priradených turnusov a turnusy T_{73} až T_{179} majú zasa zvýšenú frekvenciu priradených turnusov (diagonála od zhora dole).

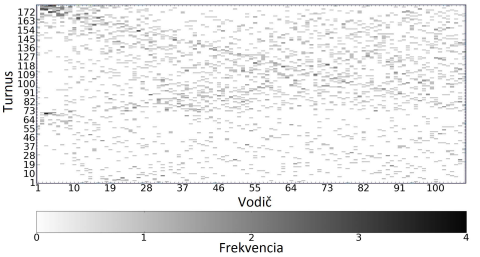
5 Záver

Článok sa venoval problematike rozpisovania služieb vodičom autobusov vo verejnej autobusovej doprave na linkách do 50 km. Na vyriešenie problému bola zvolená heuristická metóda day-by-day, ktorá vytvára rozpis služieb po rade pre každý deň plánovaného obdobia. Slabinou (neistotou) algoritmu je, či vytvorený rozpis služieb v uvažovaný deň vedie k optimálnemu výsledku. Z tohoto dôvodu je časť algoritmu fuzzifikovaná a riešená ako fuzzy inferenčný systém. Vstupom systému sú kandidáti na riešenie (odchýlky od ideálneho pracovného času a frekvencie podobných turnusov) a výstupom je index preferencie. Tento index slúži v zovšeobecnenej priraďovacej úlohe ako cena priradenia medzi turnusmi a vodičmi. Iným dôvodom zavedenia fuzzy inferenčného systému je uľahčenie tvorby rozpisu služieb vzhľadom na skúsenosti tvorcov rozpisov (fuzzy premenné a ich hodnoty, fuzzy pravidlá).

Prečo sa v článku snažíme minimalizovať nerovnomernosti v pracovných výkonoch vodičov autobusov? Dôvodov je viacero. Jedným z nich je vplyv na morálku vodičov. Mo-



(a) Relatívna odchýlka kumulovaného pracovného času od ideálneho za celé obdobie rozpisovania služieb



(b) Frekvencie opakovania podobných turnusov priradených vodičom za celé obdobie rozpisovania služieb

Obrázok 4: Relatívna odchýlka a frekvencia navrhnutého 28 dňového rozpisu služieb vodičov autobusov

rálka vodičov a tak isto aj práceschopnosť (menej absencií) sa zvyšuje s rovnomernejšími pracovnými výkonmi. Prečo článok maximalizuje u vodičov frekvenciu rovnakých alebo podobných turnusov? Tak isto aj v tomto prípade je dôvodov viacero. Napríklad bezpečnosť a pohodlie vodičov autobusov (návyk na trasu turnusu, tieto turnusy sú odjazdené bezpečnejšie). No netreba zabúdať aj na to, že sa turnusy nesmú opakovať príliš často. Je možné, že samotná trasa turnusu a jej rôznorodosť pôsobí na vodičov tak, že zostávajú pozorní a že neupadajú do mdlôb z dlhého času sedenia za volantom. Potom by sa ponúkalo viac priestoru pre tvorcu rozpisov služieb, aby sa početnosť podobných turnusov mohla zvýšiť.

Balíček pyFuzzy je veľmi užitočný nástroj. Uľahčuje tvorbu fuzzy množín a prácu s fuzzy logikou pri tvorbe aplikácií. Jeho výhodami je napríklad ľahké modelovanie neistoty, jednoduchý návrh bázy pravidiel a jednoduchý návrh bázy dát (oproti pravdepodobnostným rozdeleniam). Jeho nevýhodami sú napríklad výpočtová rýchlosť inferenčného systému, chýba užívateľská dokumentácia a čiastočná kompatibilita s Windows.

Poďakovanie Táto práca vznikla s podporou grantovej agentúry VEGA vrámci riešenia projektu 1/0135/08 „Optimalizačné problémy v logistických a dopravných systémoch“.

Literatúra

[1] POLIAK, M., GNAP, J., *Práca vodičov nákladných automobilov a autobusov a používanie tachografov*, Žilinská univerzita v Žiline/EDIS-vydavateľstvo ŽU, ISBN 978-80-8070-989-1, (2009).

[2] THEODOROVIC, D., LUČIĆ, P., *A fuzzy set theory approach to the aircrew rostering problem, Fuzzy Sets and Systems, Vol 95, Issue 3*, 261–271,(1998).

[3] LEE, K.H., *First Course on Fuzzy Theory and Applications*, str. 236–243, ISBN 3540229884, (2005)

[4] Python Programming Language, `http://www.python.org/`

[5] GNU Linear Programming Kit (GLPK), Package for solving large-scale linear programming (LP) and mixed integer programming (MIP), `www.gnu.org/software/glpk`

[6] Python fuzzy package, http://pyfuzzy.sourceforge.net/
[7] Gnuplot, A portable command-line driven graphing utility, http://www.gnuplot.info/

Kontaktná adresa

Tomáš LADOVSKÝ (Ing.),
Fakulta riadenia a informatiky, Žilinská univerzita,
Univerzitná 1, 010 26 ŽILINA,
tomas.ladovsky@fri.uniza.sk

Otvorený softvér vo vzdelávaní, výskume a v IT riešeniach
1.–4. júla 2010, Žilina, Slovensko

Organizátori: Miloš Šrámek, Spoločnosť pre otvorené informačné technológie
Tatiana Šrámková, Katedra fyziky, FEI STU Bratislava
Michal Kaukič, Aleš Kozubík, Tomáš Majer, Žilinská univerzita
Lýdia Gábrišová, Ľubica Michálková, Žilinská univerzita
Juraj Bednár, Digmia, Slovensko
Miloslav Ofúkaný, GeoCommunity, Slovensko
Peter Mráz, Kremnica
Slavko Fedorik, SOŠ elektrotechnická, Poprad
Peter Štrba, Spojená škola/Gymnázium M. Galandu, Turčianske Teplice
Ladislav Ševčovič, FEI, Technická univerzita v Košiciach

Editori: Michal Kaukič
Miloš Šrámek
Slavko Fedorik
Ladislav Ševčovič

Recenzenti: Mgr. Juraj Bednár
Mgr. Rudolf Blaško, PhD.
RNDr. Ján Buša, CSc.
Ing. Slavko Fedorik
Ing. Karol Grondžák, PhD.
Mgr. Michal Kaukič, CSc.
Ing. Tomáš Kliment
RNDr. Aleš Kozubík, PhD.
Mgr. Juraj Michálek
doc. RNDr. Štefan Peško, CSc.
Ing. Pavel Stříž, PhD.
RNDr. Ladislav Ševčovič
Ing. Michal Žarnay, PhD.

Vydavateľ: Spoločnosť pre otvorené informačné technológie – SOIT, Bratislava

ISBN 978-80-970457-0-8

Sadzba programom pdfTEX Ladislav Ševčovič
