

**UNIVERZA V LJUBLJANI
EKONOMSKA FAKULTETA**

MAGISTRSKO DELO

**UPRAVLJANJE PROMETNE SIGNALIZACIJE
Z UPORABO MEHKE LOGIKE**

Ljubljana, marec 2004

ALENKA MALEJ

IZJAVA

Študentka Alenka Malej izjavljam, da sem avtorica tega magistrskega dela, ki sem ga napisala pod mentorstvom prof. ddr. Janeza Usenika, in skladno s 1. odstavkom 21. člena Zakona o avtorskih in sorodnih pravicah dovolim objavo magistrskega dela na fakultetnih spletnih straneh.

V Ljubljani, dne 12. 3. 2004

Alenka Malej

Kazalo

1	Uvod – oris problematike, opredelitev predmeta in problema raziskave.....	1
1.1	Namen raziskave s temeljno hipotezo	2
1.2	Metode dela	2
1.3	Zasnova dela.....	3
2	Mehka logika	4
2.1	Uvod	4
2.2	Pripadnostne funkcije	6
2.2.1	Definicije.....	6
2.2.2	Tipi pripadnostnih funkcij.....	7
2.2.3	Konstrukcija pripadnostnih funkcij.....	10
2.3	Operacije nad mehкими množicami.....	10
2.3.1	Komplement.....	11
2.3.2	Operator preseka	11
2.3.3	Operator unije	13
2.4	Sistemi mehke logike	15
2.4.1	Uvod.....	15
2.4.2	Mehčanje.....	16
2.4.3	Mehko sklepanje	18
2.4.4	Ostrenje	21
3	Upravljanje prometne signalizacije.....	26
3.1	Cilji upravljanja prometne signalizacije.....	26
3.2	Upravljalni postopki	27
3.2.1	Od določenih časovnih intervalov do zaznave vozil.....	28
3.2.2	Metode matematične optimizacije	30
3.3	Zaključek	32
4	Simulacije v prometu	33
4.1	Področje in pristopi prometne simulacije.....	34
4.2	Razvoj simulacij v prometu.....	34
4.3	Simulacijski sistem za oceno upravljalnika svetlobne signalizacije v križišču.....	35

5	Upravljanje prometne signalizacije z uporabo mehke logike.....	39
5.1	Pregled literature o mehkem upravljanju prometne signalizacije.....	39
5.2	Struktura mehkega upravljanja signalizacije.....	42
5.3	Splošnost mehkega upravljanja signalizacije.....	43
5.3.1	Signalizirani prehod za pešce.....	43
5.3.2	Dvofazno upravljanje signalizacije za vozila.....	44
5.3.3	Večfazno upravljanje signalizacije za vozila.....	45
5.3.4	Vključitev prednosti javnega prevoza.....	46
5.3.5	Upravljanje prometne signalizacije na večjih glavnih mestnih cestah.....	46
5.4	Postopek mehkega upravljanja signalizacije.....	47
5.4.1	Spremenljivke pri mehkem upravljanju signalizacije.....	47
5.4.2	Mehčanje.....	48
5.4.3	Sistem sklepanja.....	49
5.4.4	Ostrenje.....	54
5.5	Primerjava mehkega upravljanja signalizacije z drugimi sistemi.....	54
6	Primer uporabe mehke logike pri upravljanju signalizacije izbranega križišča.....	58
6.1	Analiza problema.....	58
6.2	Modeliranje sistema mehkega upravljanja signalizacije.....	61
6.2.1	Definicija lingvističnih spremenljivk.....	62
6.2.2	Konstrukcija sistema sklepanja.....	66
6.3	Optimizacija.....	68
7	Rezultati.....	71
8	Sklep.....	74
	Slovarček tujih izrazov.....	75
	Kazalo slik.....	76
	Kazalo tabel.....	77
	Literatura in viri.....	78

1 Uvod – oris problematike, opredelitev predmeta in problema raziskave

Semaforizirana križišča so danes splošno znana struktura po vsem svetu in usmerjajo na milijone vozil v prometu. Dobro načrtovanje prometne svetlobne signalizacije pripomore k povečanju učinkovitosti cestnega omrežja, kar je pomembno tako z ekonomskega kot tudi z ekološkega vidika. Načrtovanje signalizacije ima med drugim neposreden vpliv na varnost v prometu in na gibanje pešcev.

Mehka logika postaja vse bolj razširjena metoda pri upravljanju prometne signalizacije (Niittymäki, 2002, str. 27). Mehki pristop uporablja bazo preprostih pravil oblike IF-THEN stavkov, ki glede na trenutno prometno situacijo izračunajo upravljalno akcijo. Na primer: »Če je število približujočih se vozil veliko in je število čakajočih vozil majhno, potem je interval zelenega signala dolg.« Opisi, kot so velik, majhen in dolg, so mehke množice. To pomeni, da nimajo točno določene meje in lahko vsebujejo tudi elemente z delno stopnjo pripadnosti (Klir, St. Clair, Yuan, 1997, str. 7). Recimo: 5 vozil je do neke stopnje majhno, po drugi strani pa tudi do neke stopnje veliko število vozil.

Moderni upravljalniki signalizacije uravnavajo signalne intervale glede na trenutne prometne razmere, katere zaznavajo z različnimi detektorji. Zaradi mnogih spremenljivk, ki nastopajo v prometu, je ustrezno matematično reprezentacijo pri upravljanju signalizacije težko izpeljati (Bingham, 1998, str. 57). Kompleksnost upravljalne procedure včasih omejujejo omejene kapacitete signalnih upravljalnikov, zato klasične matematične optimizacije niso primerne in iščemo enostavnejše optimizacijske procedure.

Mehka logika je primerna za upravljanje prometne signalizacije, kajti z uporabo lingvističnih spremenljivk in pravil omogoča enostaven prenos ekspertnega znanja in aproksimativne narave človeškega sklepanja. Mehka logika razume lingvistična navodila in omogoča generiranje upravljalne strategije na osnovi verbalne komunikacije. Hkrati predstavlja tudi sredstvo za prevod lingvistične upravljalne strategije, izražene z if-then stavki, v upravljalni algoritem (Zimmermann, 1996, str. 204). Motivacija za izgradnjo mehkega upravljalnega sistema prometne signalizacije je obstoj neposredne povezave med prostimi lingvističnimi izrazi pri upravljalni strategiji in njeno implementacijo (Niittymäki, 2002, str. 25).

Z lingvističnim modeliranjem sistemu povečamo transparentnost in možnost kasnejšega optimiziranja. Sistemi mehke logike so v primerjavi s tradicionalnimi sistemi tudi

enostavnejši za načrtovanje in lažji za vzdrževanje. Osnovne operacije so preproste in vključujejo manjše število parametrov (Niittymäki, 2001, str. 9).

Izboljšanje prometne signalizacije z ekonomskega vidika pomeni prihranek, saj prispeva k zmanjšanju količine časa, ki ga porabimo v prometu.

1.1 Namen raziskave s temeljno hipotezo

Namen raziskave je predstaviti uporabo mehke logike pri upravljanju prometne signalizacije. Zato v prvem delu raziskave preučimo osnove mehke logike, sisteme prometne signalizacije in metode preverjanja učinkovitosti teh sistemov. Drugi del raziskave pa je usmerjen k analizi upravljanja prometne signalizacije z uporabo mehke logike. Poudarek je na predstavitvi sistematičnega pristopa mehkega upravljanja signalizacije in na tvorjenju baze pravil sistema mehke logike glede na različne prometne situacije in na osnovi ekspertnega znanja. Namen raziskave je tudi predstaviti uporabo izpeljanega algoritma na konkretnem primeru, za izbrano križišče v Kopru.

V strokovni literaturi se zadnja leta pojavlja veliko prispevkov na temo mehkega upravljanja, ki kažejo na učinkovitost uporabe mehke logike pri upravljanju prometne signalizacije. To namreč prispeva k zmanjšanju zakasnitev in časov prehoda ter k zmanjšanju deleža ustavljanja vozil, kar je pomembno tudi za varnost v prometu.

Hipoteza magistrskega dela je: uporaba mehke logike je pri upravljanju prometne signalizacije smiselna in ekonomsko upravičena.

1.2 Metode dela

Magistrsko delo zajema več metod. Teoretični del temelji na proučevanju literature tujih in domačih avtorjev. Pomembni prispevki z najnovejšimi spoznanji so pridobljeni iz znanstvenih revij in z interneta.

V drugem delu magistrskega dela, pri opisu mehkega upravljalnega sistema signalizacije na konkretnem primeru, delo temelji na uporabi računalniškega programa za načrtovanje in optimizacijo sistemov mehke logike, kjer je uporabljena metoda grafičnega in interaktivnega prikazovanja podatkov ter metoda poizkusov in napak.

1.3 Zasnova dela

Magistrsko delo je razdeljeno na sedem poglavij. Prvo, uvodno poglavje, je oris problematike in opredeli predmet, problem ter namen raziskave. Drugo poglavje zajema osnove mehke logike. Predstavljeni bodo osnovni pojmi in definicije mehke množice, pripadnostne funkcije, tipi pripadnostnih funkcij, operacije nad mehкими množicami in sistemi mehkega upravljanja. Tretje poglavje opisuje upravljanje prometne signalizacije in metode, ki so običajno v uporabi. Med temi so metoda fiksnega časovnega intervala, metoda na zaznavo vozil in matematične optimizacije. Četrto poglavje je pregled simulacij v prometu: kakšne so, kje, zakaj in kako se uporabljajo. Peto poglavje opisuje uporabo mehke logike pri upravljanju prometne signalizacije. Predstavljen je splošni sistem upravljanja signalizacije z mehko logiko, vključno z mehčanjem (fuzifikacijo), sklepanjem (inferenco) in ostrenjem (defuzifikacijo). Poudarek je na sklepanju, in sicer na osnovi pravil glede na različne prometne situacije. V šestem poglavju je izpeljan konkreten primer uporabe mehke logike pri upravljanju signalizacije izbranega križišča v Kopru. Sedmo poglavje predstavi rezultate in ugotovitve raziskave. Sledijo zaključek, slovarček uporabljenih tujih izrazov, kazalo slik in tabel ter navedba literature in virov. V prilogi je dodan izpis datoteke sistema mehke logike za upravljanje prometne signalizacije izbranega križišča v tekstovni obliki.

2 Mehka logika

2.1 Uvod

Mehka logika ali fuzzy logic je tehnologija, ki omogoča opis sistemov z uporabo vsakdanjega govornega jezika. Lotfi Zadeh, profesor sistemske teorije na kalifornijski univerzi, je avtor prvih publikacij na temo mehke logike. V svojem članku (Zadeh, 1965) je predstavil mehke množice in pripadnostne funkcije ter definiriral nekatere operacije nad mehkiimi množicami. Pregled zgodovine mehke logike najdemo v več virih (Kosko, 1997, str. 5; Hellendoorn, 1997; *fuzzyTECH 5.3 User's Manual*, 1999).

S teoretično matematičnega vidika je mehka logika zvezna razširitev klasične logike s tem, da ni omejena na klasični računski vrednosti 0 in 1, temveč dopušča tudi delne in večvrednostne resnice. V klasični logiki je izjava lahko le pravilna ali nepravilna. Matematično to opišemo tako, da pravilnost izjave zavzame vrednost 1 ali 0. V večvrednostni logiki pravilnost izjave lahko zavzame več vrednosti, na primer 0, $\frac{1}{2}$ ali 1. V mehki logiki lahko pravilnost izjave zavzame katerokoli realno vrednost na intervalu med 0 in 1. Torej je mehka logika večvrednostna logika.

Mehka logika z uporabo lingvističnih spremenljivk in pravil omogoča enostaven prenos ekspertnega znanja in aproksimativne narave človeškega sklepanja. Mehka logika razume lingvistična navodila in omogoča generiranje upravljalne strategije na osnovi verbalne komunikacije. Hkrati predstavlja tudi sredstvo za prevod lingvistične upravljalne strategije, izražene z if-then stavki, v upravljalni algoritem (Zimmermann, 1996, str. 204).

Z lingvističnim modeliranjem sistemu povečamo transparentnost in možnost kasnejšega optimiziranja. Sistemi mehke logike so v primerjavi s tradicionalnimi sistemi tudi enostavnejši za načrtovanje in lažji za vzdrževanje. Osnovne operacije so preproste in vključujejo manjše število parametrov (Niittymäki, 2001, str. 281).

Mehka logika je orodje, s katerim skušamo posnemati človekov način odločanja. Veliko prednost predstavlja pri obravnavanju in reševanju problemov, ki jih težko matematično modeliramo zaradi pomanjkanja ali nepopolnosti podatkov, oziroma procesov, ki so sami po sebi prekompleksni.

Mehke množice omogočajo specifično matematično interpretacijo nenatančnih izrazov iz govornega jezika (von Altrock, 1997, str. 21). Mehka množica je generalizacija klasične

množice, saj lahko za razliko od običajne množice, kateri elementi pripadajo popolnoma ali pa ji sploh ne pripadajo, vsebuje svoje elemente tudi samo delno.

Mehko logiko lahko definiramo tudi kot uporabo mehkih množic v logičnih izrazih (von Altrock, 1997, str. 27).

Mehko sklepanje se ukvarja s pravili v obliki IF-THEN stavkov. Množica vseh pravil podaja relacije med vhodnimi in izhodnimi spremenljivkami sistema ter predstavlja bazo znanja. Osnovna ideja mehkega sklepanja izvira iz dejstva, da so mnogi vsakodnevni dejavniki nenatančni. Na primer, izjava »Nebo je modro« je lahko pravilna le do neke stopnje. Ko nebo postaja vse bolj sivo, se vrednost pravilnosti te izjave manjša, težko pa je reči, kdaj nebo ni več modro.

V pravilu »Če je X enak S , potem je Y enak T « nastopata X in Y kot lingvistični spremenljivki, S in T pa kot lingvistični vrednosti. Lingvistična spremenljivka je spremenljivka, ki zavzame lingvistične vrednosti. Primer: »starost« je lahko lingvistična spremenljivka, ki ima vrednosti »mlad«, »srednjih let« in »star«. Vsako od teh lingvističnih vrednosti lahko predstavimo z mehko množico. Neka oseba lahko pripada eni ali več takim množicam hkrati. Oseba, stara 26 let, je do neke stopnje še mlada, obenem pa do neke druge stopnje tudi srednjih let in tako naprej. Razdeliti »starost« na več odsekov, ki se ne prekrivajo, bi bilo zelo težko in nepraktično.

Literature s področja mehke logike je mnogo (Virant, 1992; Golob, Muškinja, 1994; Ross, 1995; Zimmermann, 1996; Klir, St. Clair, Yuan, 1997; Kosko, 1997; Virant, 1998). Mehka logika postaja vse bolj razširjena metoda na mnogih področjih (von Altrock, 1995; von Altrock, 1997; David, Malej 2002; Usenik, Malej, 2002), tudi na področju prometa in prometne signalizacije (Khiang, Khalid, Yusof, 1995; Adler, Blue, 1998; Palacharla, Nelson, 1999; Wahle et. al., 2001; Bingham, 2001; Murat, Gedizlioglu, 2002; Chou, Teng, 2002; Niittymäki, 1998 – 2002).

Žal so nekateri avtorji preveč optimistični glede možnosti mehke logike. Čeprav gre za novo tehnologijo in se izračunavanje v aplikacijah razlikuje od klasičnih sistemov, ni samoumevno, da nudijo mehki sistemi boljše rezultate. Izbira upravljalnega sistema (mehkega, nevronskega, neurofuzzy ali matematične optimizacije itd.) mora temeljiti na učinkovitosti sistema in ne na spremljajoči filozofiji.

2.2 Pripadnostne funkcije

2.2.1 Definicije

Karakteristična funkcija običajne množice A

$$\chi_A : U \rightarrow \{0,1\}$$

preslika element x univerzalne množice U v 0 ali 1 in je definirana:

$$\chi(x) = \begin{cases} 0; & x \notin A \\ 1; & x \in A \end{cases} \quad (2.1)$$

Torej karakteristična funkcija običajne množice loči le elemente in ne-elemente množice A . Pri *mehki množici*, definirani kot množica urejenih parov

$$S = \{(x, \mu_S(x)) \mid x \in U\},$$

je vsakemu elementu $x \in U$ prirejena *stopnja pripadnosti* množici S , ki jo določa *pripadnostna funkcija* (Klir, St. Clair, Yuan, 1997, str. 75)

$$\mu_S : U \rightarrow [0,1]$$

Pri tem je U običajna množica. Vrednost pripadnostne funkcije $\mu_S(x)$ je 0, ko x sploh ne pripada množici S , je 1, ko x pripada množici S popolnoma, in $0 < \mu_S(x) < 1$, če x pripada množici S delno. Stopnja nejasnosti je tako za mehko množico natančno določena s pripadnostno funkcijo μ_S .

Mehka množica S (ali ekvivalentno pripadnostna funkcija μ_S) je *normalna*, če obstaja kak element $x \in U$, da je $\mu_S(x) = 1$. Sicer je množica S *subnormalna*. V tem magistrskem delu privzamemo, da so vse mehke množice normalne.

Support množice S (ali μ_S) je množica elementov, katerih pripadnost množici S je večja od 0 (Zimmermann, 1996, str. 14):

$$\text{supp}(S) = \{x \mid \mu_S(x) > 0, x \in U\}$$

2.2.2 Tipi pripadnostnih funkcij

Najbolj pogoste pripadnostne funkcije so trikotne, trapezoidne, oblik »z« in »s« ter Gaussove.

Trikotna pripadnostna funkcija je definirana:

$$\mu(x; p_1, p_2, p_3) = \begin{cases} \frac{x-p_1}{p_2-p_1}, & x \in [p_1, p_2] \\ -\frac{x-p_3}{p_3-p_2}, & x \in (p_2, p_3] \\ 0, & \text{sicer} \end{cases}$$

Parametri p_1, p_2, p_3 določajo na abscisi koordinate oglišč (glej sliko 2.1a, str. 9).

Trapezoidna pripadnostna funkcija je splošno zapisana

$$\mu(x; p_1, p_2, p_3, p_4) = \begin{cases} \frac{x-p_1}{p_2-p_1}, & x \in [p_1, p_2] \\ 1, & x \in (p_2, p_3] \\ -\frac{x-p_4}{p_4-p_3}, & x \in (p_3, p_4] \\ 0, & \text{sicer} \end{cases}$$

Parametri p_1, p_2, p_3, p_4 določajo na abscisi koordinate oglišč (glej sliko 2.1b, str. 9).

Poseben primer trapezoidne funkcije je pravokotna pripadnostna funkcija, kjer je $p_1 = p_2$ in $p_3 = p_4$. Tukaj je funkcija ekvivalentna karakteristični funkciji (2.1) običajne množice: vrednost pripadnostne funkcije je 1 za elemente, ki množici pripadajo, in 0 za ostale.

Trikotna funkcija je tudi poseben primer trapezoidne pripadnostne funkcije. Trapezoidna funkcija pa je bolj fleksibilna od trikotne, ker ima en parameter več.

Posebna primera trapezoidne funkcije sta tudi funkciji oblike »s« in »z«. Pri funkciji oblike »s« (slika 2.1c, str. 9) sta parametra p_3 in p_4 izven definicijskega območja funkcije, pri funkciji oblike »z« (slika 2.1d, str. 9) pa sta izven definicijskega območja funkcije parametra p_1 in p_2 .

Našteti tipi pripadnostnih funkcij so zelo priljubljeni zaradi lahke izračunljivosti (Golob in Muškinja, 1994, str. 50). Pri optimizaciji so uporabne tudi kosoma linearne funkcije (glej

sliko 2.1e, na str. 9). Naštete pripadnostne funkcije pa niso povsod gladke. Zato obstajajo tudi nelinearne funkcije, te pa niso tako enostavne.

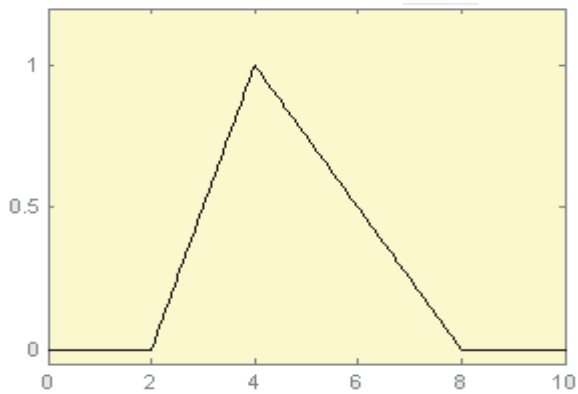
Gaussova pripadnostna funkcija

$$\mu(x; c, \sigma) = e^{-\frac{(x-c)^2}{2\sigma^2}}$$

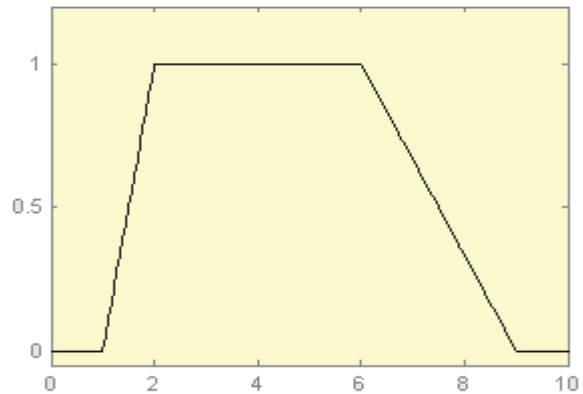
je značilne zvončaste oblike (glej sliko 2.1f, na str. 9). Parametra c , σ določata središče in širino pripadnostne funkcije.

V znanstveni literaturi je bilo predlaganih kar nekaj tipov pripadnostnih funkcij. Na primer analitične funkcije, kot je $1/(x_0 - x)^2$, posplošena zvončasta funkcija, sigmoidalna funkcija in razne empirične funkcije. Take funkcije so pomembne bolj s teoretičnega vidika. Predlagane so bile zaradi nekaterih matematičnih lastnosti, v praksi pa niso toliko uporabne, predvsem zaradi zahtevnosti glede izračunljivosti. Za potrebe mehke logike v magistrskem delu zadoščajo funkcije, ki smo jih definirali.

Slika 2.1: Pripadnostne funkcije in njihovi parametri



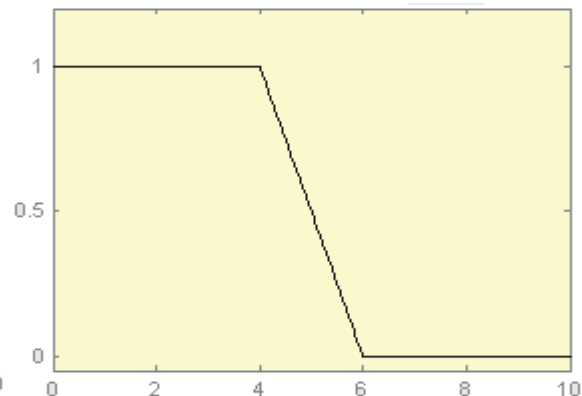
a) Trikotna pripadnostna funkcija s parametri $p_1 = 2, p_2 = 4, p_3 = 8$.



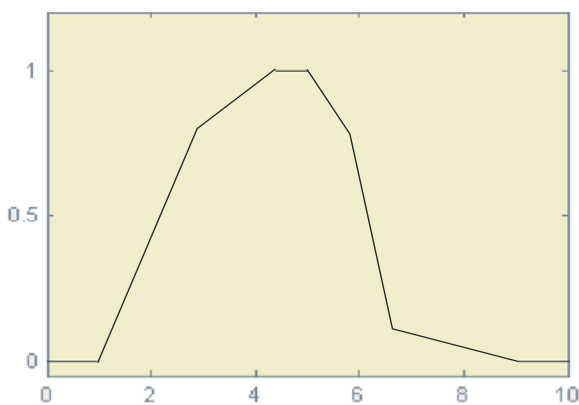
b) Trapezoidna pripadnostna funkcija ($p_1 = 1, p_2 = 2, p_3 = 6, p_4 = 9$).



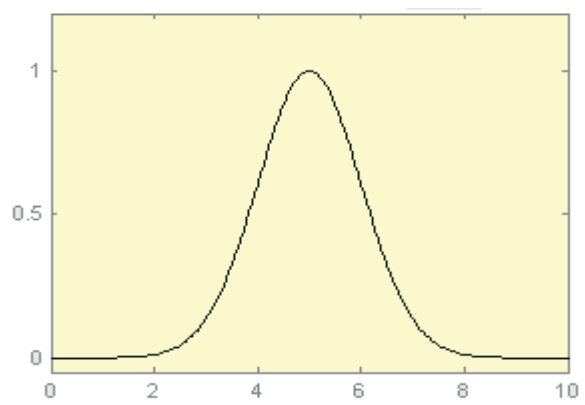
c) Pripadnostna funkcija oblike »s« ($p_1 = 4, p_2 = 6, p_3 = 11, p_4 = 12$).



d) Pripadnostna funkcija oblike »z« ($p_1 = -2, p_2 = -1, p_3 = 6, p_4 = 8$).



e) Kosoma linearna pripadnostna funkcija



f) Gaussova pripadnostna funkcija ($c = 5, \sigma = 1$)

2.2.3 Konstrukcija pripadnostnih funkcij

Pripadnostne funkcije predstavljajo lingvistične vrednosti in morajo opisati naravo ter lastnosti lingvističnih spremenljivk. Poznamo direktno in indirektno metodo konstrukcije pripadnostnih funkcij (Klir in Yuan, 1995, str. 282). Obe uporabljata ekspertno znanje o lingvističnih spremenljivkah in lingvističnih vrednostih, ki so v obravnavi.

Pri direktni metodi ekspert priredi elementu $x \in U$ stopnjo pripadnosti $\mu_S(x)$ mehki množici S . Ekspert lahko poda matematično formulo pripadnostne funkcije ali (bolj pogosto) primere vrednosti $\mu_S(x)$ za posamezne elemente x . V slednjem primeru ekspert odgovarja na vprašanje »Kolikšna je stopnja pripadnosti elementa x v S ?« ali »Kateri element x ima stopnjo pripadnosti $\mu_S(x)$ v S ?« Z odgovori na taka vprašanja dobimo množico urejenih parov $\{x, \mu_S(x)\}$, ki jih uporabimo za konstrukcijo pripadnostne funkcije (Klir, St. Clair, Yuan, 1997, str. 87). Nekatere oblike pripadnostnih funkcij so definirane že vnaprej in tedaj uporabimo urejene pare za določitev njihovih parametrov.

Pri indirektni metodi za elemente x_1, x_2, \dots, x_n iz množice U med seboj primerjamo njihove stopnje pripadnosti množici S . Včasih je lažje primerjati, kakšna je stopnja pripadnosti mehki množici, kot pa kar podati točno stopnjo pripadnosti za vsak element kot pri direktni metodi. Rezultat primerjave je matrika P , katere elementi $p_{ij} = \mu_S(x_i) / \mu_S(x_j)$ predstavljajo, koliko bolj/manj element x_i pripada množici S od elementa x_j . Ekspert torej ne poda natančnih vrednosti $\mu_S(x_i)$ in $\mu_S(x_j)$. Dejanske vrednosti pripadnostne funkcije se izračunajo, tako da poiščemo lastne vrednosti in lastne vektorje matrike P (Klir in Yuan, 1995, str. 287). Rezultat so spet pari $\{x, \mu_S(x)\}$, ki jih uporabimo pri konstrukciji pripadnostne funkcije.

Pri obeh metodah, direktni in indirektni, moramo uskladiti odgovore, če je ekspertov več in so njihova mnenja različna.

2.3 Operacije nad mehкими množicami

Znane operacije nad mehкими množicami so komplement, presek in unija. Z vsemi tremi operacijami so povezane razne matematične interpretacije (Klir, St. Clair, Yuan, 1995; Zimmermann, 1996). Nekatere izmed njih bomo obravnavali v nadaljevanju.

2.3.1 Komplement

Komplement \bar{S} mehke množice S je mehka množica s pripadnostno funkcijo (Zimmermann, 1996, str. 17)

$$\mu_{\bar{S}}(x) = 1 - \mu_S(x).$$

Ta definicija je ekvivalentna definiciji komplementa klasične množice. Stopnjo $\mu_{\bar{S}}(x)$ lahko interpretiramo ne samo kot stopnjo pripadnosti elementa x množici \bar{S} , ampak tudi kot stopnjo, s katero x *ne pripada* množici S . Podobno lahko interpretiramo, da je $\mu_S(x)$ stopnja, s katero x ne pripada množici \bar{S} .

V podpoglavju 2.1 stopnja mehкости množice S temelji na pripadnostni funkciji μ_S . Koncept komplementa pa nam daje način merjenja mehкости (Kosko, 1997, str. 351). Množica S je mehka natanko tedaj, ko je presek množice S in njenega komplementa \bar{S} neprazna množica. Pri običajnih, rečemo jim tudi ostrih množicah, je presek prazen.

2.3.2 Operator preseka

Obstaja več matematičnih operatorjev za presek mehkih množic (Zimmermann, 1996, str. 29). Slika 2.2 (str. 12) primerja nekatere. (Če so pripadnostne funkcije narisane v istem koordinatnem sistemu, kot na sliki, morajo biti funkcije iste lingvistične spremenljivke S . Formule, ki jih bomo predstavili, pa lahko direktno apliciramo tudi na pripadnostne funkcije različnih lingvističnih spremenljivk S_1, S_2, \dots, S_n .)

Standardni operator preseka je *minimum* (slika 2.2b, str. 12). Presek $S = \bigcap_{i=1}^n S_i$ mehkih množic S_i , $i = 1, \dots, n$, ima tedaj pripadnostno funkcijo

$$\mu_S = \min \{ \mu_{S_1}, \dots, \mu_{S_n} \},$$

kjer je μ_{S_i} pripadnostna funkcija množice S_i , $i = 1, \dots, n$.

Minimum je tudi operator preseka za klasične množice. Ni pa gladek in zato ga včasih nadomesti njegova odvedljiva različica, *mehki minimum*:

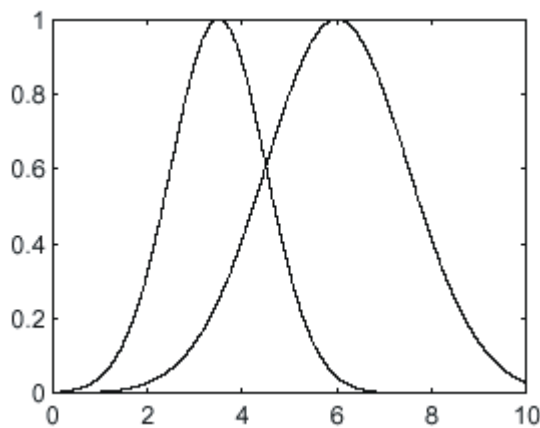
$$\mu_S = \text{softmin} \{ \mu_{S_1}, \dots, \mu_{S_n} \} = \frac{\sum_{i=1}^n \mu_{S_i} e^{-k\mu_{S_i}}}{\sum_{i=1}^n e^{-k\mu_{S_i}}},$$

kjer je k parameter. Mehki minimum se asimptotično približuje minimumu, ko gre parameter k proti neskončnosti. Pri nizkih vrednostih parametra k daje mehki minimum nepričakovane rezultate za majhne vrednosti pripadnostnih funkcij. Slika 2.2c prikazuje mehki minimum dveh Gaussovih pripadnostnih funkcij. Parameter $k = 10$ je očitno premajhen.

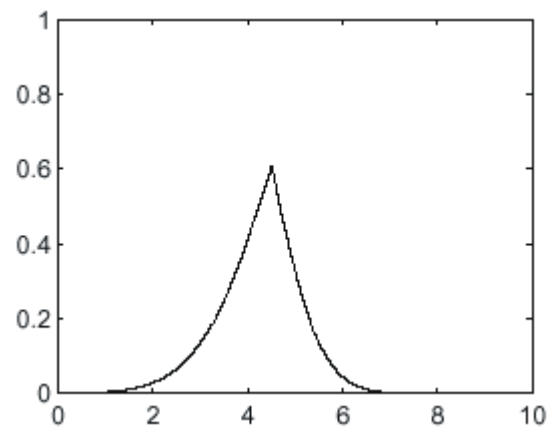
Še en operator preseka je *produkt*:

$$\mu_S = \prod_i \mu_{S_i}$$

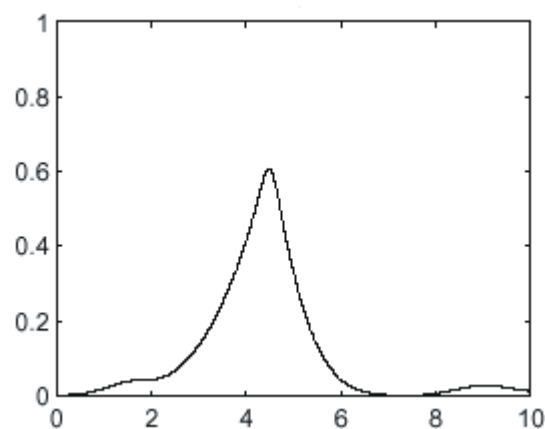
Slika 2.2: Operatorji preseka



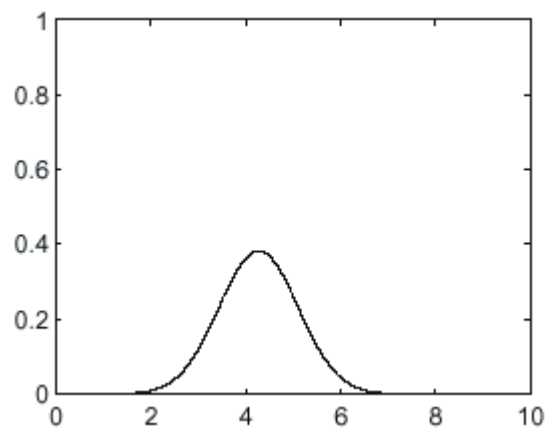
a) Gaussovi pripadnostni funkciji μ_1 in μ_2



b) Minimum μ_1 in μ_2



c) Mehki minimum μ_1 in μ_2 za $k = 10$



d) Produkt μ_1 in μ_2

Produkt je gladka funkcija (slika 2.2d, str. 12) in je v literaturi pogost zaradi svoje preprostosti. Produkt upošteva, za razliko od minimuma in mehkega minimuma, vse vrednosti pripadnostnih funkcij. To postane pomembno pri več kot dveh vrednostih pripadnostnih funkcij. Vzemimo na primer dve množici vrednosti pripadnostnih funkcij (μ_1, μ_2, μ_3) : $(0.1, 0.9, 0.9)$ in $(0.1, 0.2, 0.2)$. Operator minimum daje v obeh primerih rezultat 0.1. Mehki minimum za $k=10$ da 0.1003 za prvo množico in 0.1269 za drugo; za $k=100$ pa 0.1000 in 0.1000. Operator produkt pa daje rezultat 0.081 za prvo in 0.04 za drugo množico. Torej le produkt upošteva, da so vrednosti pripadnostnih funkcij v prvi množici v povprečju večje od vrednosti v drugi množici. Mehki minimum za $k=10$ daje ravno obraten rezultat in tako zaključimo, da je vrednost parametra $k=10$ premajhna. V aplikaciji je minimum lahko bistvenega pomena, morda pa je potrebno upoštevati tudi vrednosti drugih pripadnostnih funkcij.

2.3.3 Operator unije

Tudi operacija mehke unije ima več interpretacij (Zimmermann, 1996, str. 29). Primerjavo prikazuje slika 2.3 (str. 14).

Običajno je unija interpretirana kot *maksimum* (slika 2.3b, str. 14). Pri operatorju maksimum ima unija množic $S = \bigcup_{i=1}^n S_i$ pripadnostno funkcijo

$$\mu_S = \max \{ \mu_{S_1}, \dots, \mu_{S_n} \},$$

kjer je μ_{S_i} pripadnostna funkcija množice S_i , $i = 1, \dots, n$. To je tudi operator unije pri klasičnih množicah.

Alternativa je operator *vsota* (slika 2.3c, str. 14):

$$\mu_S = \sum_i^n \mu_{S_i}.$$

Vsota spet upošteva vse informacije, medtem ko jih maksimum, podobno kot minimum pri preseku, ne. Ker lahko vsota pripadnostnih funkcij presega 1, je običajno omejena na vrednost od 0 do 1 z uporabo *omejene vsote* ali *bounded sum* (slika 2.3d, str. 14), katere pripadnostna funkcija je

$$\mu_S = \min \left\{ 1, \sum_{i=1}^n \mu_{S_i} \right\},$$

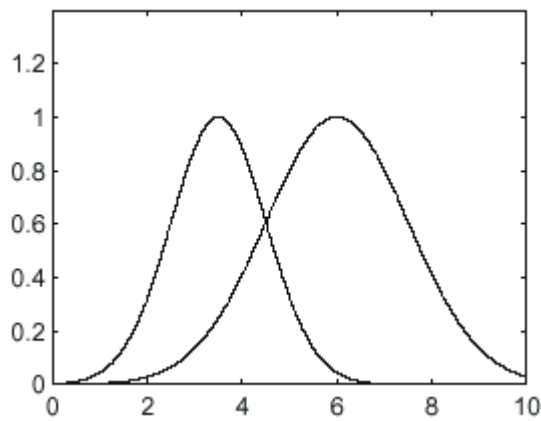
ali z uporabo *verjetnostne vsote* (Zimmermann, 1996, str. 28), katere pripadnostna funkcija za dve množici je

$$\mu_S = \mu_{S_1} + \mu_{S_2} - \mu_{S_1} \mu_{S_2},$$

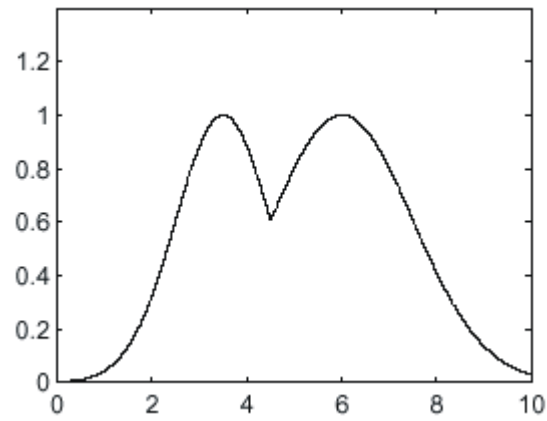
v splošnem pa

$$\mu_S = 1 - \prod_{i=1}^n (1 - \mu_{S_i}).$$

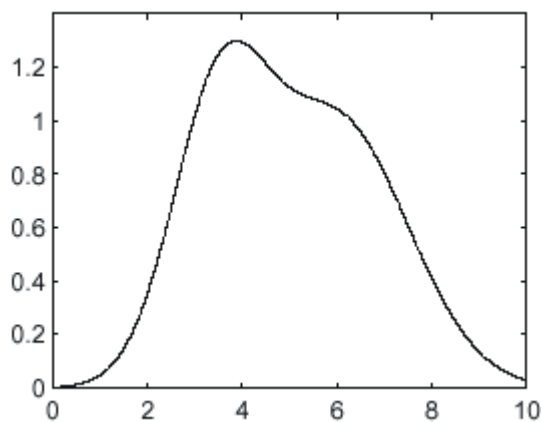
Slika 2.3: Operatorji unije



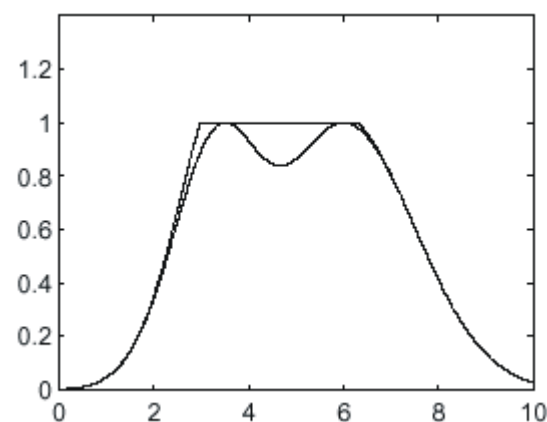
a) Gaussovi pripadnostni funkciji μ_1 in μ_2



b) Maksimum μ_1 in μ_2



c) Vsota μ_1 in μ_2



d) Omejena vsota (zgoraj) in verjetnostna vsota (spodaj) μ_1 in μ_2

2.4 Sistemi mehke logike

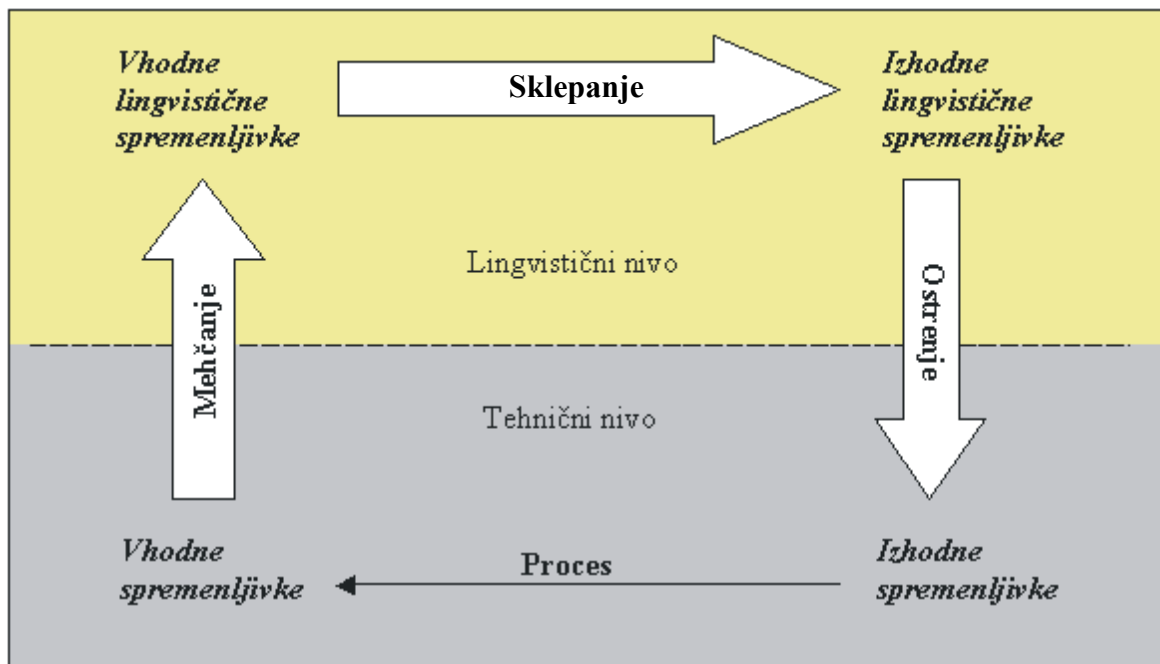
2.4.1 Uvod

V zadnjih tridesetih letih je bilo razvitih mnogo metod, ki uporabljajo mehke množice. Mehka logika se je izkazala kot najbolj praktična in najbolj uspešna aplikacija teorije mehkih množic. Stopnja pripadnosti množici tukaj postane stopnja pravilnosti izjave: vzemimo za primer vreme. V splošnem govorimo o stopnji oblačnosti: naj ta znaša recimo 40%. Torej je izjava »Nebo je oblačno« pravilna s stopnjo 0.4.

V magistrskem delu se bomo omejili le na mehko logiko na osnovi pravil. Skoraj vse aplikacije mehke logike v zadnjem času temeljijo na tej metodologiji (Fuzzy Logic Toolbox User's Guide, 2000). V tem podpoglavju bomo torej predstavili sisteme mehke logike na osnovi pravil, njihovo strukturo in osnovne elemente.

Sistem mehke logike zajema tri osnovne korake: mehčanje, sklepanje in ostrenje (Zimmermann, 1996, str. 207). Celotno strukturo sistema mehke logike prikazuje slika 2.4.

Slika 2.4: Struktura sistema mehke logike



Vir: von Altrock, 1995, str. 37.

V prvem koraku moramo prevesti vse vhode v lingvistične spremenljivke. Tokrat vzemimo primer semaforiziranega križišča: naj bo eden izmed vhodnih podatkov »število vozil, ki se križišču približujejo iz smeri zelenega signala«. Naj bo to število enako 10. Z mehčanjem lahko število deset prevedemo v »veliko število vozil« (slika 2.5, str. 17). Skratka, realne vhodne vrednosti sistema prevedemo v lingvistične vrednosti.

V drugem koraku, ko so vse vhodne vrednosti prevedene v vrednosti lingvističnih spremenljivk, sistem sklepanja z množico pravil obdela lingvistične podatke in definira obnašanje sistema. Rezultat tega koraka so spet lingvistične vrednosti, ki pa veljajo za izhodne lingvistične spremenljivke. Če je na primer izhodna spremenljivka sistema »dolžina podaljška zelenega signala«, je lahko lingvistični rezultat za to spremenljivko vrednost »malo več kot srednja«.

V zadnjem koraku, ki ga imenujemo ostrenje, se lingvistični rezultat sistema prevede v realno vrednost. V našem primeru je to določeno število sekund, za kolikor se naj zeleni signal podaljša.

2.4.2 Mehčanje

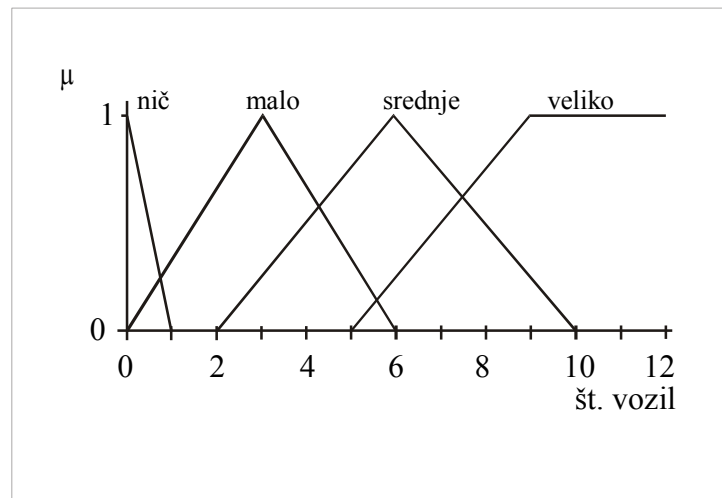
Za vse spremenljivke, ki nastopajo v sistemu, je potrebno definirati ustrezne lingvistične spremenljivke. Za vsako lingvistično spremenljivko pa moramo definirati tudi vse njene možne lingvistične vrednosti (Zimmermann, 1996, str. 131). V primeru križišča so lahko spremenljivke take, kot jih prikazuje tabela 2.1.

Tabela 2.1: Primer lingvističnih spremenljivk in njihovih lingvističnih vrednosti

Lingvistične spremenljivke	Možne lingvistične vrednosti
1. Število vozil iz smeri A	{nič, malo, srednje, veliko}
2. Število vozil iz smeri B	{nič, malo, srednje, veliko, preveč}
3. Podaljšek zelenega signala	{nič, kratek, srednji, dolg}

Za vsako posamezno lingvistično spremenljivko je vsaka lingvistična vrednost definirana s pripadnostno funkcijo (slika 2.5).

Slika 2.5: Pripadnostne funkcije za primer spremenljivke »število vozil iz smeri A«



Vhodne ostre vrednosti se v koraku mehčanja prevedejo v vrednosti lingvističnih spremenljivk. Te vrednosti so predstavljene z mehкими množicami. Prevod poteka tako, da vsaki vhodni vrednosti s pripadnostnimi funkcijami priredimo stopnjo pripadnosti ustreznim mehkim množicam. V primeru s slike 2.5 se število 7 prevede v »srednje« s stopnjo pripadnosti 0.75 in »veliko« s stopnjo pripadnosti 0.5.

Na začetku poglavja smo predstavili razne tipe pripadnostnih funkcij. Vse našete tipe pripadnostnih funkcij, razen Gaussove funkcije, lahko matematično predstavimo kot kosoma linearne funkcije. Stopnjo pripadnosti vhodne vrednosti sistema neki mehki množici tedaj izračunamo na sledeči način:

1. Definiramo pripadnostno funkcijo z množico karakterističnih točk $P(x_i, \mu_i)$, kjer je $i = 1, \dots, n$, $x_1 \leq x_i \leq x_n$ in $\mu_i \in [0, 1]$. Interval $[x_1, x_n]$ je definicijsko območje funkcije in vhodna vrednost $X \in [x_1, x_n]$.
2. Ocenimo vhodne vrednosti: $x_k \leq X \leq x_{k+1}$
3. Izračunamo stopnje pripadnosti: $\mu = \mu_i + (X - x_i)(\mu_{i+1} - \mu_i)/(x_{i+1} - x_i)$

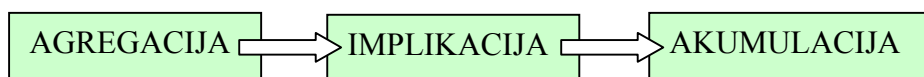
Pri funkcijah, ki niso kosoma linearne, je postopek podoben. Potrebno je definirati interval z $\mu(x) = 0$ in $\mu(x) = 1$. Stopnjo pripadnosti mehki množici izračunamo s pomočjo analitičnega zapisa pripadnostne funkcije.

2.4.3 Mehko sklepanje

Poleg lingvističnih spremenljivk so osnovni gradnik sistema mehke logike še množice pravil. Ponekod imamo en sam blok pravil, lahko pa tudi več blokov pravil, če gre za strukturiran problem. Pravila so klasične oblike **If ... Then**, le da operirajo z lingvističnimi vrednostmi – podobno kot v govorjenem jeziku. Pravila definirajo relacije med lingvističnimi spremenljivkami. S tem določajo potek akcij, ki jih izvede upravljalnik. Sestavljena so iz dveh delov: prvi del je pogoj, drugi del pa posledica. Pogoj je lahko sestavljen iz več podpogojev, ki so med sabo povezani z vezniki, kot sta IN ter ALI.

Računanje z mehкими pravili imenujemo mehko sklepanje (inferenca). Po prevodu vseh vrednosti vhodnih spremenljivk v vrednosti vhodnih lingvističnih spremenljivk lahko sistem mehkega sklepanja v množici pravil identificira tista, ki ustrezajo vhodnim podatkom, in izračuna vrednost izhodne lingvistične spremenljivke.

Mehko sklepanje tvorijo tri komponente: agregacija, implikacija in akumulacija (Virant, 1998, str. 77).



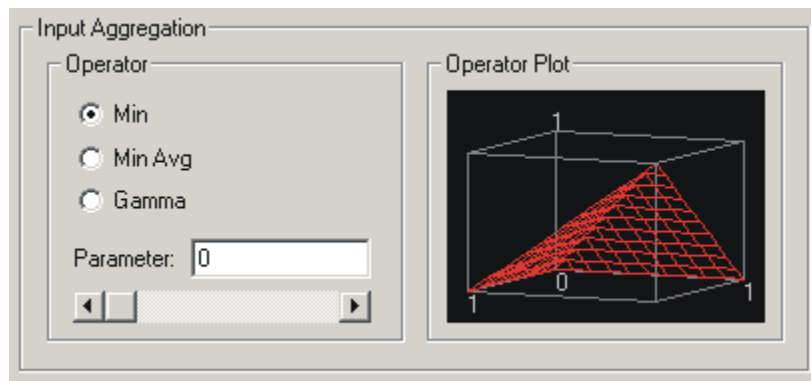
Agregacija

Pri *agregaciji* izračunamo, v kolikšni meri je zadoščeno pogojem posameznih pravil. Ker so pogoji lahko sestavljeni iz več podpogojev, uporabimo za agregacijo stopenj zadoščanja podpogojem različne operatorje, kot sta operatorja preseka in unije, ki predstavljata lingvistična IN ter ALI. Za to se najpogosteje uporabljata operatorja minimum (slika 2.6) in maksimum (slika 2.7).

$$\text{IN: } \mu_{\text{IF}} = \min_i(\mu_i)$$

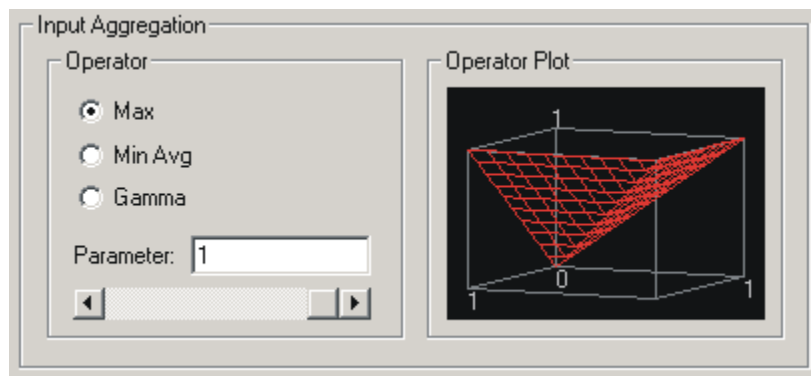
$$\text{ALI: } \mu_{\text{IF}} = \max_i(\mu_i)$$

Slika 2.6: Operator minimum



Vir: *fuzzyTECH* 5.52.

Slika 2.7: Operator maksimum



Vir: *fuzzyTECH* 5.52.

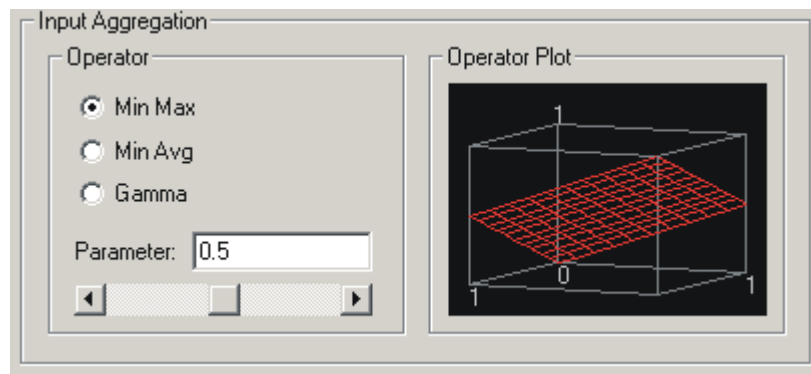
Na razpolago so tudi drugi operatorji za presek in unijo, ki smo jih opisali v podpoglavju 2.3. Pri večjih upravljalnih aplikacijah se uporabljajo nekateri kompenzacijski operatorji:

$$\text{Min-Max: } \mu_{\text{IF}} = \lambda * \min_i(\mu_i) + (1 - \lambda) * \max_i(\mu_i),$$

$$\text{Gama: } \mu_{\text{IF}} = \left(\prod_{i=0}^n (\mu_i) \right)^{1-\gamma} * \left(1 - \prod_{i=0}^n (1 - \mu_i) \right)^{\gamma},$$

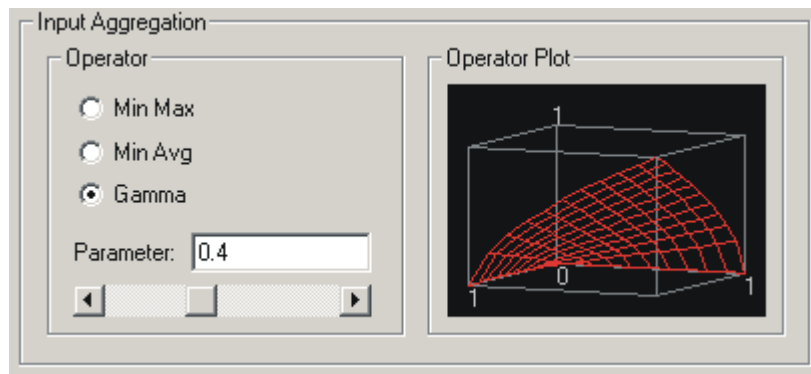
kjer sta λ in γ kompenzacijska parametra. Gre za kombinacijo že definiranih operatorjev: operator min-max je linearna kombinacija minimuma in maksimuma, operator gama pa kombinacija produkta in omejene vsote.

Slika 2.8: Operator min-max



Vir: *fuzzyTECH* 5.52, vrednost parametra $\lambda = 0.5$.

Slika 2.9: Operator gama



Vir: *fuzzyTECH* 5.52, vrednost parametra $\gamma = 0.4$.

Operatorja minimum in maksimum se uporabljata najpogosteje, in sicer zaradi njune preprostosti in razumljivosti (von Altrock, 1997, str. 339). Sta najlažje in najhitreje izračunljiva, včasih pa tudi programsko orodje ne omogoča drugih operatorjev. Imata pa tudi pomanjkljivosti, ki smo jih opisali že v podpoglavju 2.3.2. Operatorja minimum in maksimum slabo posnemata človeški način agregacije, ki teži h kompenzaciji in je predvsem odvisen od konteksta. Zato je prišlo do uvedbe kompenzacijskega operatorja gama (Zimmermann, 1996, str. 37), pri katerem pa je potrebno poiskati najbolj optimalno vrednost parametra γ .

Rezultat agregacije je torej stopnja pravilnosti prvega dela pravil.

Implikacija

Agregaciji sledi *implikacija*, ki izračuna posledico pravil. V mnogih primerih je stopnja pravilnosti pogojev pravil hkrati tudi stopnja pravilnosti celotnega pravila. Pri implikaciji se stopnja pravilnosti prvega dela pravila pomnoži z vnaprej podanim faktorjem ali t. i. utežjo. Vsako pravilo ima svojo stopnjo pomembnosti, kar v sistemu modeliramo tako, da vsakemu pravilu priredimo utež. V programu *fuzzyTECH* se ta utež imenuje DoS (Degree of Support) in lahko zavzame vrednosti na intervalu $[0,1]$.

Običajno je operator implikacije kar množenje:

$$\mu_{\text{THEN}} = \mu_{\text{IF}} * \text{DoS}$$

Primer: pravilo z utežjo 0 je za sistem popolnoma nepomembno, saj ne vpliva na rezultat.

Akumulacija

Po implikaciji se lahko zgodi, da nekatera pravila dajo isti lingvistični rezultat, toda z različno stopnjo pravilnosti.

Primer: eno pravilo da rezultat »Interval je dolg« s stopnjo 0.7, neko drugo pravilo pa »Interval je dolg« s stopnjo 0.9.

V takem primeru je potrebno rezultate pred ostrenjem uskladiti. Ta korak se imenuje *akumulacija*. Običajno se uporabi operator maksimum ali omejeno vsoto:

$$\text{MAX: } \mu_{\text{REZULTAT}} = \max_i(\mu_{\text{THEN},i})$$

$$\text{BSUM: } \mu_{\text{REZULTAT}} = \min(1, \sum(\mu_{\text{THEN},i}))$$

Akumulacija določi, katera pravila in v kolikšni meri bodo prispevala k rezultatu ostrenja.

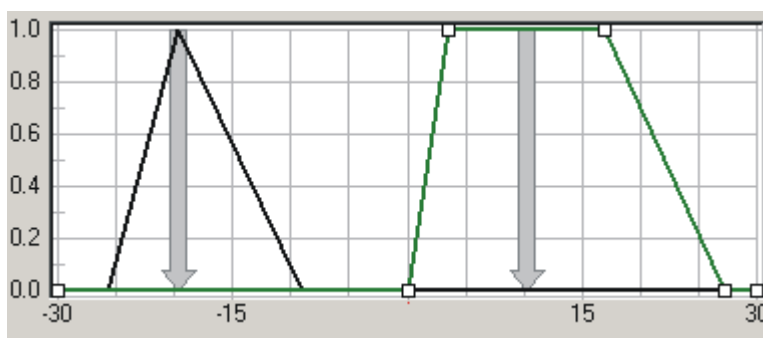
2.4.4 Ostrenje

Ob koncu sklepanja je rezultat podan z vrednostmi lingvističnih spremenljivk. Te mehke rezultate je treba uskladiti v ostro vrednost oziroma numerični rezultat. To fazo imenujemo ostrenje. Ostrenje uporabi pripadnostne funkcije izhodne spremenljivke, ki podajajo relacije med lingvističnimi in ostrimi vrednostmi.

Obstajajo različne metode ostrenja (Ross, 1995, str. 134; Virant, 1998, str. 79; von Altrock, 1997, str. 356): večina jih poteka v dveh korakih. Najprej se za vsako lingvistično vrednost izbere najbolj »tipična« numerična vrednost, nato pa se te vrednosti uskladijo.

Najobičajnejši pristop pri izbiri najbolj tipične vrednosti vsake mehke množice je poiskati maksimum vsake pripadnostne funkcije. Če ima pripadnostna funkcija maksimalni interval, izberemo recimo srednjo vrednost tega intervala.

Slika 2.10: Izbira tipičnih vrednosti mehkih množic



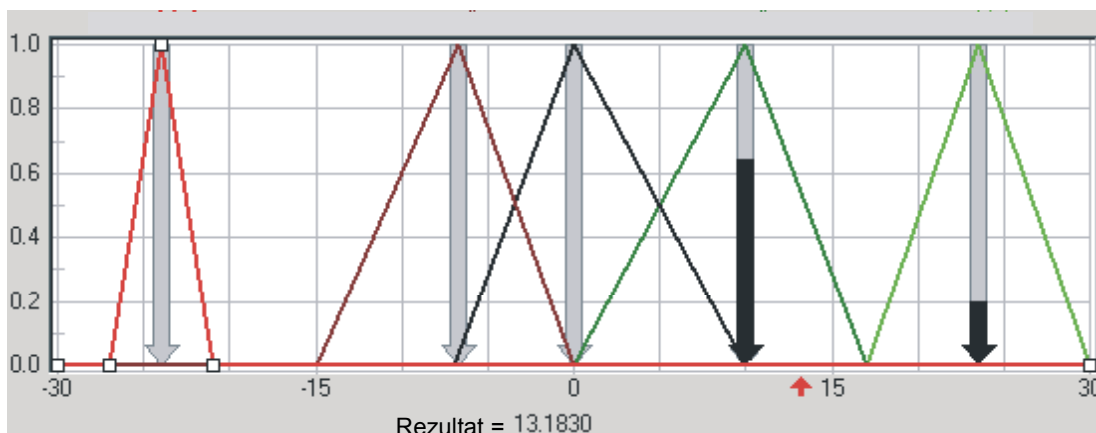
V drugem koraku uskladimo dobljene tipične vrednosti glede na rezultate mehkega sklepanja. Upošteva je ta izračun ločimo več metod ostrenja.

Metoda *center maksimumov* ali *Center of Maximum (CoM)*, imenovana tudi *metoda višine* (Virant, 1998, str. 84), izračuna uteženo povprečje maksimumov Y_i pripadnostnih funkcij μ_i :

$$Y = \frac{\sum_i (\mu_{\text{REZULTAT},i} * Y_i)}{\sum_i \mu_{\text{REZULTAT},i}} .$$

Grafično prikazuje metodo CoM slika 2.11.

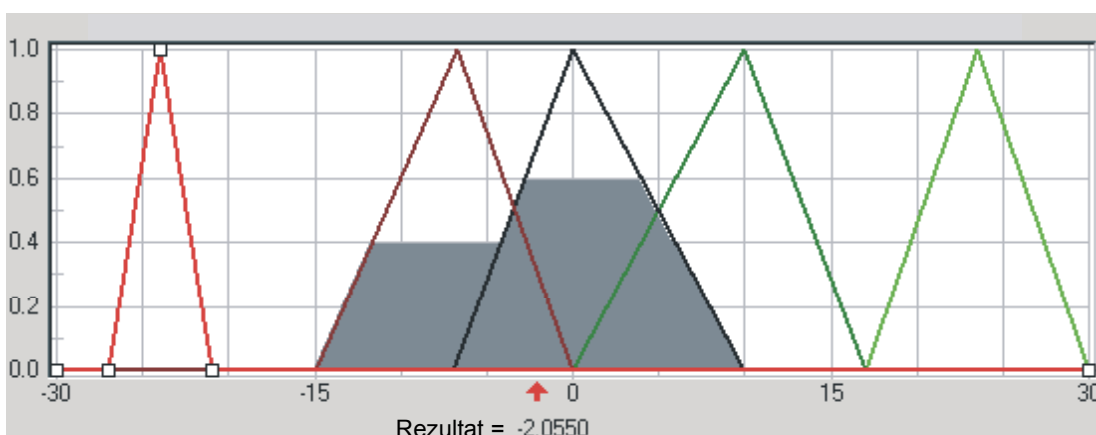
Slika 2.11: Primer ostrenja z metodo CoM



Najpogosteje uporabljena metoda ostrenja je *center ploščine* ali *Center of Area* (CoA), imenovana tudi *težiščna metoda* ali *Center of Gravity* (CoG) (Virant, 1998, str. 80). Ta metoda odreže vsako pripadnostno funkcijo pri stopnji veljavnosti tega lingvističnega rezultata. Ploščina, ki jo omejujejo tako dobljene funkcije z abscisno osjo, je uporabljena za izračun težišča (glej sliko 2.12).

$$Y = \frac{\int_U y \cdot \mu_{\text{REZULTAT}}(y) dy}{\int_U \mu_{\text{REZULTAT}}(y) dy}$$

Slika 2.12: Primer ostrenja z metodo CoA

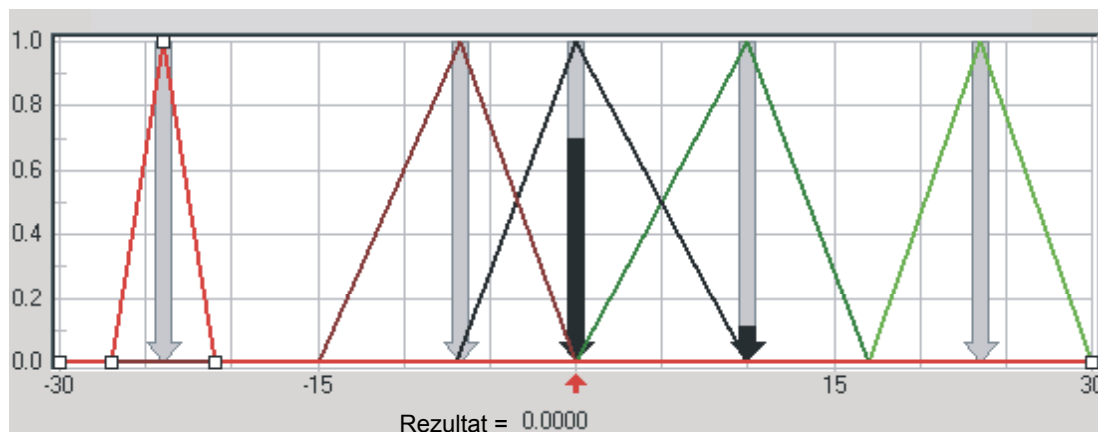


Pri metodi CoA moramo biti pozorni na to, da ploščina, ki jo omejuje pripadnostna funkcija, tukaj bistveno prispeva k rezultatu. Slaba lastnost te metode je tudi visoka časovna

zahtevnost, ki je lahko do 1000-krat večja kot pri metodi CoM, odvisno od procesorja (von Altrock, 1997, str. 361).

Omenimo še *metodo srednjega maksimuma* ali *Mean of Maximum (MoM)*. Ta metoda nam daje maksimum z največjo utežjo, kar predstavlja najbolj smiseln rezultat. Metoda MoM je med drugim primerna pri klasifikaciji.

Slika 2.13: Primer ostrenja z metodo MoM



2.4.4.1 Primerjava metod ostrenja

Pomembna lastnost metod ostrenja je zveznost. Definicija zveznosti je: »Vzemimo sistem mehke logike s polno množico pravil (pri vsaki kombinaciji vhodnih spremenljivk se uporabi vsaj eno pravilo) in s pripadnostnimi funkcijami, ki se prekrivajo. Metoda ostrenja je zvezna, če neskončno majhna sprememba vhodne spremenljivke nikoli ne povzroči ogromne spremembe v katerikoli izhodni spremenljivki.« (von Altrock, 1997, str. 362)

Metodi CoM in CoA sta zvezni, medtem ko metoda MoM ni zvezna. Računanje povprečja je zvezna operacija, pri najbolj smiselnem rezultatu pa nastopajo skoki.

Izbira metode ostrenja je odvisna od vsebine odločanja. Pri kvantitativnem odločanju uporabimo CoM, pri kvalitativnem odločanju pa MoM. Vprašanje je, če kot rešitev potrebujemo najboljši kompromis ali najbolj smiseln rezultat. Pri nekaterih aplikacijah (na primer klasifikaciji) pa ostrenje sploh ni nujno potrebno. Pripadnostne stopnje za izhodno spremenljivko so rezultat klasifikacije in definirajo podobnost izhodnega signala sistema z opazovanim objektom.

Pri izhodnih lingvističnih spremenljivkah kompleksne funkcije ne dajejo boljših rešitev. Večina aplikacij uporablja le trikotne funkcije, saj metodi ostrenja CoM in MoM od pripadnostnih funkcij uporabljata le njihove maksimume.

Naslednja tabela primerja opisane metode ostrenja.

Tabela 2.2: Primerjava metod ostrenja

metoda lastnost	CoA	CoM	MoM
lingvistična karakteristika	najboljši kompromis	najboljši kompromis	najbolj smiseln rezultat
ujemanje z intuicijo	slabo v primerih prekrivanja funkcij	dobro	dobro
zveznost	da	da	ne
učinkovitost glede na izračunljivost	zelo nizka	visoka	zelo visoka
aplikacije	upravljanje, podpora odločanju, analiza podatkov	upravljanje, podpora odločanju, analiza podatkov	prepoznavanje vzorca, podpora odločanju, analiza podatkov

Vir: von Altrock, 1997, str. 363.

3 Upravljanje prometne signalizacije

Upravljanje prometne signalizacije je v današnji družbi zelo pomemben dejavnik. Dobro načrtovanje signalizacije pripomore k povečanju učinkovitosti cestnega omrežja, to pa je pomembno z ekonomskega in z ekološkega vidika. Načrtovanje signalizacije neposredno vpliva na gibanje vozil in tudi pešcev, na varnost v prometu in na okolje, ki nas obdaja.

Pri upravljanju prometne signalizacije ločimo med upravljanjem izoliranih sistemov signalizacije in upravljanjem signalizacije več križišč hkrati. V magistrskem delu se bomo omejili na izolirana križišča. Izoliran sistem pomeni, da promet skozi križišče poteka neodvisno od ostalih križišč. To omogoča široko izbiro strategij pri načrtovanju upravljalnega sistema svetlobne signalizacije.

3.1 Cilji upravljanja prometne signalizacije

Glavni razlog za uvedbo prometnih signalov v križišču je varnost v prometu, saj signalizacija loči konfliktno prometne tokove. Po odločitvi o uvedbi prometne signalizacije v križišče se lahko postavijo razni cilji: minimum zakasnitve vozil in/ali pešcev, minimum ustavljanja vozil, minimum dolžine čakajoče kolone vozil, maksimum varnosti in/ali udobnosti tako za vozila kot za pešce, minimum negativnih vplivov na okolje, upoštevanje prednosti vozil javnega prevoza in tako naprej.

Pri upravljanju prometne signalizacije se pojavljajo trije glavni cilji: maksimalna varnost, minimalne zakasnitve in minimalne negativne posledice za okolje, kar pomeni maksimalne ekonomske učinke. Težava je v tem, da optimum vsakega cilja dosežemo z različnimi dolžinami ciklov. Problem upravljanja prometne signalizacije lahko opišemo s trikotnikom, kjer so glavni cilji v ogliščih. Vsak cilj vpliva na drugi cilj, kot prikazuje slika 3.1. Če je zmanjšanje zakasnitev glavni cilj, nekoliko negativno vpliva na ostala cilja. Edina pozitivna zveza se vzpostavi med okoljem in varnostjo. Z drugimi besedami, upravljanje, primerno za okolje, je lahko tudi varno, saj so takrat cikli dolgi, kar pomeni malo obdobje rumene luči in nevarnost trčenja je tako manjša. Največji problem je, da se tedaj zakasnitve povečajo tudi do 40% v primerjavi z optimalnimi vrednostmi (Niittymäki in Pursula, 2000, str. 17).

Z ekonomskega vidika pomeni zmanjšanje zakasnitev zmanjšanje količine časa, ki ga porabimo v prometu. Zmanjšanje negativnih vplivov na okolje pomeni manj izpušnih plinov, torej manjšo porabo goriva in s tem prihranek. Prav tako predstavlja prihranek tudi povečanje varnosti, saj se na ta način zmanjša materialna škoda.

Slika 3.1: Glavni cilji upravljanja prometne signalizacije in njihovi medsebojni vplivi



Vir: Niittymäki, 2001a, str. 229.

Običajno je glavna mera učinkovitosti signalizacijskega sistema zmanjšanje zakasnitve vozil in števila ustavljanj, saj je ostale dejavnike težko meriti. Definirajmo *zakasnitev* (delay) kot razliko med realnim časom prehoda (križišča) in željenim časom prehoda:

$$d = t_{tr} - \frac{s_{tr}}{v_{des}}$$

kjer je d zakasnitev vozila v sekundah, t_{tr} dejanski čas prehoda vozila, s_{tr} dolžina poti in v_{des} željena hitrost. Željeni čas prehoda je količnik med dolžino poti in željeno hitrostjo. Pri željenem času prehoda privzamemo, da druga vozila ali drugi dejavniki ne ovirajo prehoda. Torej zakasnitev delno zajema vpliv zasičenja in delno dodaten čas zaradi čakanja na signaliziranem križišču, vendar to ni ločeno.

3.2 Upravljalni postopki

V nalogi bomo obravnavali problem *izoliranega križišča* oziroma *izoliranega upravljanja signalizacije*. Gre za obravnavo enega samega križišča; njegova signalizacija je neodvisna in ne sodeluje z drugimi križišči. To pomeni, da je lahko upravljalni algoritem enostavnejši od algoritmov za usklajene mreže križišč, sistem upravljanja signalizacije pa ima večjo svobodo pri izbiri upravljalne strategije.

Pri upravljanju signalizacije se različni prometni tokovi borijo za isti čas in prostor. Različne prioritete so pogosto dodeljene različnim prometnim tokovom ali skupinam vozil. Seveda optimizacija sočasno vključuje več kriterijev, kot so povprečna zakasnitev, maksimalna dolžina kolone in delež vozil, ki stojijo. Tako v praksi upravljanje prometne signalizacije

temelji na rešitvah po meri in popravkih, ki jih opravijo planerji. Moderni signalni upravljalniki, ki jih je moč programirati, imajo veliko število nastavljivih parametrov in so zato zelo uporabni v tem procesu. Za dobre rezultate sta potrebna tako dober in izkušen planer kot tudi natančna prilagoditev sistema na terenu.

Pri upravljalnih procedurah je pomemben podatek o *prometnem pretoku* vozil, ki ga merimo v številu vozil na uro in podaja oceno razmer na cesti in stopnjo zasičenosti. Na primer: vzemimo križišče dveh enosmernih cest z dvema voziščema in enakim prometnim pretokom. Prometni pretok 300 vozil na uro pomeni, da se v obeh smereh na obeh voziščih generira v eni uri približno 150 vozil, kar skupaj znaša 300 vozil na uro v vsaki smeri.

Več kot 40% vseh upravljalnikov signalizacije po svetu je dvofaznih (Bingham, 1998, str. 57). Pri *dvofaznem upravljanju* promet v danem trenutku poteka bodisi v vzdolžni ali v prečni smeri. Vozil, ki zavijajo v desno ali levo, ne obravnavamo posebej, prav tako ni vključeno upravljanje signalizacije za pešce.

Poznamo več upravljalnih postopkov. Najenostavnejši je postopek s fiksnimi časovnimi intervali. V primerih uporabe detektorjev vozil za merjenje prometa se upravljalni postopek imenuje postopek na zaznavo vozil. Znani so še postopki, ki temeljijo na matematični optimizaciji, med nove postopke pa štejemo tudi upravljanje z mehko logiko.

3.2.1 Od določenih časovnih intervalov do zaznave vozil

Preden so prišli v uporabo detektorji, so upravljalniki delovali po principu vnaprej določenih *fiksni časovni intervalov*. V teh sistemih sta dolžini cikla in vsake faze konstantni. Optimalno dolžino cikla pogosto izračunamo po Webstrovi formuli, ki minimizira skupne zakasnitve za vnaprej znane prometne pretoke. Optimalna dolžina cikla se deli na časovne intervale v skladu s stopnjo zasedenosti vsake faze. Webstrova formula (1958) se glasi (Niittymäki, 2002, str. 14):

$$C = \frac{1,5F + 5}{1 - \sum \left\{ \max \left(\frac{q_i}{s_i} \right) \right\}},$$

kjer je

C = optimalna dolžina cikla (s),

F = skupni izgubljeni čas (s),

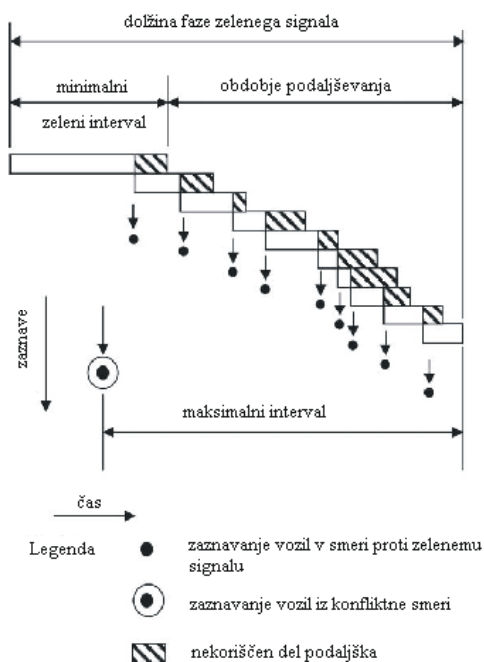
q_i = prometni pretok iz vsake smeri (vozišča) v fazi i (št. vozil/uro),

s_i = prometni pretok pri zasičenju vsake smeri (vozišča) v fazi i (št. vozil/uro).

Pri fiksnih intervalih vsaka smer dobi določeno dolžino zelenega signala, ne glede na to, koliko vozil prihaja iz te smeri. Tudi če ni nobenega vozila, ji sistem dodeli običajno dolžino zelenega signala in vozila iz druge smeri morajo čakati. Upravljanje s fiksnim časom deluje optimalno v primeru, ko je prometni pretok v obeh smereh konstanten in ni naključnih variacij, kar pa je redkost.

V mnogih primerih izoliranih križišč je upravljanje s fiksnimi intervali nadomestilo *upravljanje na zaznavo vozil*. Ta upravljalna procedura uporablja informacijo v realnem času, ki jo podajo detektorji. Uporablja vnaprej določeno strategijo, ki pri izbiri časovnih intervalov upošteva prometno situacijo (Dolenc, 2002). Sistem na zaznavo vozil deluje tako, da aktivni zeleni signal deluje vsaj za nek minimalni čas (slika 3.2). Če je povpraševanje zadostno, se interval zelene luči ustrezno podaljšuje v odvisnosti od intervala med vozili do nekega maksimalnega intervala trajanja. Potem lahko nastopi rdeč signal ali pasivna zelena, kar pomeni pripravljenost na spremembo. Pasivni zeleni signal prekine povpraševanje nasprotujočih se signalov. Prvi detektor je običajno nameščen 40 – 60 metrov pred križiščem, torej je znano, koliko vozil se bo približalo stop liniji v naslednjih 4 – 6 sekundah. Minimalni interval zelenega signala je nekaj sekund, prav toliko pa traja interval podaljšanja.

Slika 3.2: Princip podaljševanja pri upravljalnem sistemu signalizacije na zaznavo vozil



Vir: Niittymäki, 2001, str. 275.

Upravljanje na zaznavo vozil je danes najbolj običajna upravljalna procedura prometne signalizacije (Bingham, 1998, str. 57). Pri tem načinu upravljanja se pojavlja princip podaljševanja. Osnovni namen je odpraviti kolono, da se vozilom, ki se približujejo, omogoči prehod križišča brez nepotrebnega ustavljanja. Princip podaljševanja matematično ni najbolj optimalna upravljalna strategija. Njena glavna slabost je v tem, da upošteva le vozila pri zelenem signalu, ne upošteva pa števila čakajočih vozil pri rdečem signalu.

3.2.2 Metode matematične optimizacije

Uvedba upravljalnikov z mikroprocesorji je omogočila vgraditev naprednejših algoritmov, zasnovanih z matematičnim modeliranjem. Izbira optimizacijske funkcije temelji na zastavljenem cilju, ta je običajno minimum zakasnitve vozil. Miller je leta 1963 predlagal optimizacijsko strategijo, katere smoter je minimizirati skupno zakasnitev vozil (Niittymäki, 2002, str. 14). Pri tej strategiji se izpelje odločitev o podaljšanju faze ob regularnih presledkih s preverjanjem upravljalne funkcije.

Millerjeva funkcija (3.1) prikazuje razliko v zakasnitvi: na eni strani imamo dodatna vozila, ki uspejo prečkati križišče med podaljškomo (prvi del), na drugi strani pa kolono vozil, ki nastane zaradi podaljška v prečni smeri (drugi del).

$$T = \left\{ \left[\begin{array}{cc} 1 - \frac{\delta_N}{s_N} & 1 - \frac{\delta_S}{s_S} \\ \delta_N + \delta_S - q_N \frac{1 - \frac{\delta_N}{s_N}}{1 - \frac{q_N}{s_N}} - q_S \frac{1 - \frac{\delta_S}{s_S}}{1 - \frac{q_S}{s_S}} & (a + r_{NS} + l_{NS}) \end{array} \right] - \left\{ h \left[n_W + n_E + \sum_{i=1}^{k_W} q_W + \sum_{i=1}^{k_E} q_E \right] \right\} \right\}, \quad (3.1)$$

kjer je:

T = upravljalna funkcija, razlika v zakasnitvi (s),

h = ocenjen podaljšek intervala (s),

δ_I = pričakovano število vozil, ki bodo prečkala križišče v naslednjih h sekundah,

q_I = pritoki vozil v naslednjih h sekundah (št. vozil/uro),

s_I = pretoki stopnje zasičenosti v naslednjih h sekundah (št. vozil/uro),

a = dolžina rumene faze (s),

r_I = dolžina naslednje rdeče faze (s),

l_I = čas, porabljen med pospeševanjem po koncu rdeče faze (s),

n_I = število vozil, ki čakajo s strani rdečega signala,

k_I = čas, potreben za sprostitev kolone (s),

I = indeks približevanja (N – sever, S – jug, W – zahod, E – vzhod).

Bång je leta 1976 razvil TOL (Traffic Optimisation Logic), osnovano na Millerjevem delu:

$$\phi_A = r_A (a_v \delta_{Av} + a_b \delta_{Ab} + a_p \delta_{Ap}) + b_v \delta_{Av} + b_b \delta_{Ap} - h (a_v n_{Bv} + a_b n_{Bb} + a_p n_{Bp}) - (b_v \Delta n_{Bv} + b_b \Delta n_{Bp}),$$

kjer je

ϕ_A = upravljalna funkcija,

r_A = dolžina intervala, dokler faza A ne postane spet zelena, če se zaključi takoj,

a = cena zakasnitve,

δ = število dodatnih avtomobilov (v), avtobusov (b), pešcev (p) itd., ki uspejo prečkati križišče, če je zeleni signal podaljšan za h sekund,

b = cena operacije, da vozilo spet doseže normalno hitrost po popolni ustavitvi,

h = dolžina intervala med kalkulacijami upravljalne funkcije, t. j. podaljšek,

n = število vozil, ki bo dodatno utrpelo zakasnitev za h sekund v koloni pri rdečem signalu, če se trenutni zeleni signal podaljša,

Δn = število dodatnih vozil, ki se bo moralo ustaviti, če se trenutni zeleni signal podaljša za h sekund.

Britanska MOVA je edina komercialna aplikacija matematične optimizacije za izolirano križišče (Niittymäki, 2002, str. 16). MOVA (Modernised Optimised Vehicle Actuation) uporablja dva detektorja na vozišče, enega pri 40 metrih in drugega pri 100 metrih pred stop črto. Vsako vozišče dobi na začetku dovolj dolg zelen signal, da se sprazni kolona do 40 metrov. Upravljalna strategija MOVA je mešanica matematične optimizacije in hevrističnih algoritmov. Po nastopu zelene faze sledijo štiri zaporedni koraki:

1. Obstaja minimalni zeleni interval za vsako fazo (v Veliki Britaniji običajno 7 sekund).
2. Spremenljiv minimalni zeleni interval poskrbi za vozila med prvim detektorjem in stop linijo.
3. Faza ostane zelena, dokler se ne sprosti zasičeni pretok.
4. Optimizacija se začne, ko je rešena zasičenost vseh vozišč. Razlog, da je za začetek optimizacije potrebno počakati do tega trenutka, je v rezultatih prometnih simulacij, ki so pokazale, da je minimalna zakasnitev običajno dosežena, če je zeleni signal dovolj dolg, da se kolona lahko popolnoma odstrani. Ideja optimizacijskega algoritma izhaja iz Millerja, vendar je algoritem bolj kompleksen. Pri optimizaciji uporablja MOVA matematični model prometa, ki predvidi pozicijo vsakega vozila med vhodnim detektorjem (100 m) in stop linijo. Na vsake pol sekunde se izvede kalkulacija, ali bo

skupna zakasnitev minimizirana, če se trenutno stanje zelenega signala nadaljuje za 0.5 s, 1 s, 1.5 s... MOVA avtomatično zazna konec okoliščin zasičenosti in namesto Millerjevega algoritma uporabi hevristično funkcijo.

3.3 Zaključek

Težavnost procesa upravljanja prometne signalizacije je v tem, da ga je potrebno ponavljati v zelo kratkih intervalih. Prometnih razmer vnaprej ni moč natančno predvideti, zato postopki optimizirajo le trenutno stanje. S tem upravljalne akcije ne morejo dolgoročno zagotavljati optimalnih razmer. Poleg tega je potrebno upoštevati še omejitve detektorjev, saj ti ne morejo zaznati podrobnosti razmer na cestišču tako dobro kot človek. Pri tem imamo v mislih na primer tip vozila in spremembo hitrosti. Če je križišče geometrijsko kompleksno, mora upravljalni proces običajno upoštevati vzajemno konfliktne cilje, med katerimi pa je vsekakor najpomembnejša varnost.

Naslednji problem je določitev pomembnosti ciljev. Upravljanje prometne signalizacije obravnava probleme, pri katerih je prisotnih več ciljev, in vodi optimizacijo le glede na nazadnje zaznane podatke. Vse to kaže, da je pri kompleksnejših sistemih tudi težje izpeljati natančne in učinkovite rešitve. Boljšo rešitev je moč pričakovati v mehanizmu, ki bi posnemal človeško razmišljanje z lingvističnimi mehкими vrednostmi.

4 Simulacije v prometu

Za prometni sistem je značilno, da je njegove lastnosti težko analizirati, kontrolirati in optimizirati. Sistem pogosto pokriva široko prostorsko področje, število udeležencev v prometu je veliko, njihovi cilji in nameni pa so pogosto v navzkrižju. Veliko je tudi zunanjih vplivov (kot je na primer vreme), ki so izven nadzora operaterja sistema in udeležencev. Prometni sistem je po svoji naravi dinamičen in stohastičen, število enot v sistemu se s časom spreminja in je precej naključno. Veliko udeležencev v sistemu ob istem času pa pomeni tudi veliko število sočasnih interakcij.

Za prometni sistem so značilne aktivnosti, ki vključujejo tako ljudi in njihove interakcije kakor tudi interakcije ljudi z napravami (na primer voznikove interakcije s cestnim okoljem in upravljalnim sistemom). Zakonitosti interakcij so po naravi nejasno določene, saj voznikove reakcije vodi človeško zaznavanje, ne pa detektorji (Pursula, 1999, str. 2).

Prometni sistemi so tako odlično okolje za raziskave s pomočjo simulacij, saj so ostala analitična orodja, čeprav zelo pomembna, omejena le na podsisteme. Simulacije se uporabljajo za testiranje in ocenjevanje predlaganih postopkov pred implementacijo, služijo pa tudi kot orodje za demonstracijo. Glavna prednost simulacij je možnost testiranja alternativnih modelov v enakih prometnih razmerah.

Simulacijo v splošnem definiramo kot predstavitev dela realnega sveta z uporabo (računalniškega) modela, ki deluje skozi čas. Računalniški modeli se pogosto uporabljajo pri analizi vseh prometnih sistemov, v magistrskem delu pa bomo obravnavali le področje cestnega prometa.

Uporaba računalniške simulacije na področju prometa se je začela s publikacijo Gerloughove disertacije »Simulation of Freeway Traffic on a General-purpose Discrete Variable Computer« na kalifornijski univerzi leta 1955 (Pursula, 1999, str. 1). Od takrat je postala računalniška simulacija zelo uporabno orodje v povezavi z različnimi aplikacijami, namenjenimi znanstvenemu raziskovanju, planiranju, optimiziranju, treniranju in demonstraciji. Vodilne silnice pri razvoju simulacij so prednosti teorije prometa, računalniške tehnologije in programskih orodij, razvoj splošne informacijske infrastrukture ter družbeno povpraševanje po natančnejših analizah.

4.1 Področje in pristopi prometne simulacije

Aplikacije programov prometnih simulacij lahko klasificiramo na več načinov. Osnovni klasifikaciji sta delitev na mikroskopski, mezoskopski in makroskopski pristop ter delitev na zvezni in diskretni časovni pristop. Glede na prostor ločimo simulacije križišča, povezave in simulacije prometne mreže. Posebni področji sta področji varnost v prometu in vpliv sistemov ATIS (Advanced Traveler Information System).

Eden izmed najstarejših primerov uporabe simulacije v teoretičnih raziskavah je analiza sledenja vozilu. Tudi analiza križišča je ena osnovnih problematik teorije prometnega pretoka in simulacije. Po 40 letih od prvih poskusov na tem področju še vedno potekajo aktivne raziskave.

Osrednja problematika simulacij v analizah cestnega prometa se nanaša na prometni pretok. Ocenjujeta se na primer dolžina kolone in časovna zakasnitev v križiščih.

Simulacije so zelo uporabne pri proučevanju svetlobne signalizacije v semaforiziranih križiščih. Moderni sistemi svetlobne signalizacije delujejo v odvisnosti od prometa in ne le s fiksnimi časovnimi zamiki. V tradicionalnih sistemih je promet reagiral na signale, zdaj pa tudi signalizacija reagira na promet. Zato je pri analizah pomembna povezava med napravami za signalizacijo in simulacijskim sistemom. Še nekoliko širši problem nastopi pri analizi prometnih strategij v coni, oblikovani iz več križišč. Simulacija postane v tem primeru precej zapletena. Simulacijskih orodij za mrežne analize je precej manj kot programov za izolirana križišča.

Vprašanja varnosti v prometu predstavljajo velik problem pri simulacijah. V tradicionalnih simulacijskih programih so vozniki programirani, da se izogibajo trčenjem. Torej trčenj in prometnih nesreč ni. Opravljenih je bilo nekaj poskusov simulacije konfliktnih situacij, vendar ta problem ostaja odprt (Pursula, 1999, str. 4).

V zadnjih letih se je pojavilo novo področje prometne simulacije, in sicer simulacija prometnega povpraševanja. Gre za individualno usmerjene modele izbire. Uporabni so pri načrtovanju poti ob upoštevanju prometnih razmer v realnem času. Pri tem so uporabne GIS podatkovne baze za vhodne podatke in prikazovanje rezultatov.

4.2 Razvoj simulacij v prometu

V drugi polovici prejšnjega stoletja in še posebej v zadnjih letih je bil razvoj na področju prometne simulacije ogromen. To seveda pripisujemo razvoju računalniške tehnologije in

programskih orodij. Po drugi strani je zaznati tudi napredek v teoriji prometa in prometnih sistemov (Nam in Drew, 1999, str. 438). Simulacije tako postajajo vsakodnevno orodje na vseh področjih raziskav v prometu.

Število aplikacij neprestano narašča. Od aplikacij, ki obravnavajo samo eno enoto, prehajamo na aplikacije, ki pokrivajo omrežje več enot. Primer mrežnega pristopa je ameriški TRANSIMS, ki v enem simulacijskem okolju kombinira celoten prometni sistem (Smartest, 2004). Poleg tega narašča potreba po vse večji natančnosti pri opisu cest in cestnega okolja, posebej pri simulaciji križišč. Uporaben je pristop grafičnih vmesnikov in integracije v sisteme GIS.

Po drugi strani se kaže potreba po odprtih okoljih, kjer je možno interaktivno uporabiti več analitičnih orodij hkrati. Primer takega dela je sodelovanje finskih in britanskih partnerjev v zvezi s finskim programom HUTSIM za prometno modeliranje (Kosonen in Bargiela, 1999).

Vse pomembnejše postaja simuliranje prometnih upravljalnih sistemov. Prometni upravljalni sistemi so v interakciji s prometom, torej je v teh primerih potrebno opisati tako reakcije voznikov kot tudi odzive upravljalnih sistemov. Posebno pomembna pri voznikovi reakciji je izbira poti, ki jo je potrebno obravnavati dinamično. V prihodnosti bo v upravljalne sisteme vgrajenih vse več simulacijskih sistemov za predvidevanje prometnega pretoka.

Kombinacija simulacij in realnih podatkov nudi potrebne informacije za upravljanje in odločanje v prometu. Napredek v elektroniki in telekomunikaciji prinaša vse naprednejša orodja za nadzor prometa in s tem pripomore k večji natančnosti in zanesljivosti podatkov v primerjavi s preteklostjo. Danes so že na voljo podatki v realnem času, kar omogoča tudi simulacijo v realnem času. S tem postaja simulacija vse bolj realistična.

4.3 Simulacijski sistem za oceno upravljalnika svetlobne signalizacije v križišču

Na področju upravljanja signalizacije se uporablja računalniška mikrosimulacija, tako v praksi za primerjave planiranih in alternativnih modelov kakor tudi pri teoretičnih raziskavah in razvoju novih modelov, metod in strategij. V obeh primerih je glavna prednost simulacij možnost testiranja alternativ v identičnih prometnih razmerah.

Glavni problem pri načrtovanju signalizacije je merjenje raznih količin pri ocenjevanju delovanja sistema. Poleg tradicionalnih meritev učinkovitosti sistema in zakasnitve vozil merimo še dolžino čakajoče vrste, količino emisij, porabo goriva in delež ustavljanja vozil.

Vse to lahko izračunamo s simulacijo, ki omogoča analizo funkcij upravljalnika signalizacije, dolžino faze zelenega signala in drugo.

Skratka, simulacije omogočajo nadzor nad spremenljivkami prometnega sistema. Podobno kot pri laboratorijskih eksperimentih lahko tudi pri simulaciji izvajamo številne teste z vnaprej izbranimi vrednostmi posameznih spremenljivk. To nam pri upravljanju prometne signalizacije daje nove možnosti testiranja in ocenjevanja strategij.

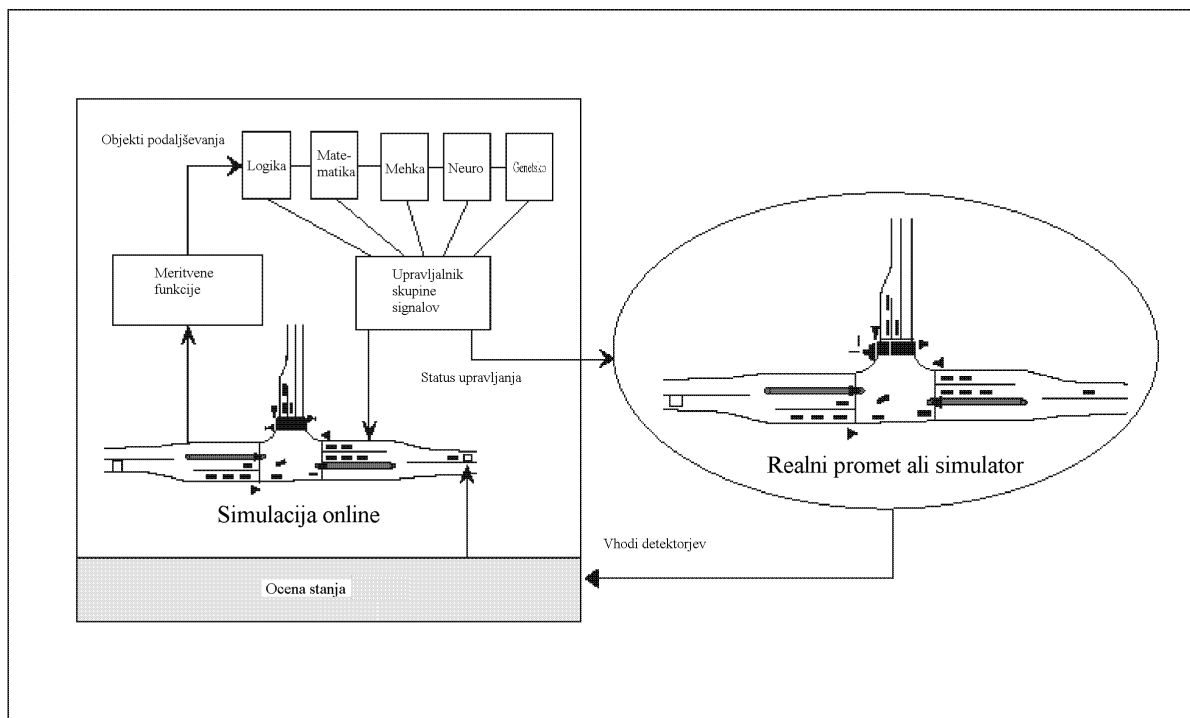
Pri testiranju nove upravljalne sheme je torej poleg upravljalnega algoritma potreben mikroskopski simulacijski model, ki omogoča testiranje veliko upravljalnih shem pri realističnih podatkih. Na trgu je več simulacijskih sistemov. Naštajmo nekatere (Vir: Smartest, 2004):

- Aimsun2,
- Anatoll,
- Autobahn,
- Casimir,
- Corsim,
- Dracula,
- Flexsyt Ii,
- Freevu,
- Fresim,
- Hutsim,
- Integration,
- Melrose,
- Microsim,
- Micstran / Tras-Tsc,
- Mitsim,
- Mixic,
- Nemis,
- Netsim,
- Padsim,
- Paramics,
- Pharos,
- Plansim-T,
- Shiva,
- Sigsim,
- Simdac,
- Simnet,
- Sism,
- Sitra-B+,
- Sitras,
- Transims,
- Thoreau,
- Vissim.

Eden znanih simulacijskih modelov je simulator HUTSIM. To je mikrosimulacijski model, ki so ga razvili na Helsinški univerzi za tehnologijo na Finskem. Glavne tri komponente simulacije so geometrija križišča, prometni pretok in upravljanje signalizacije.

Upravljanje prometne signalizacije preko simulacije in osnove on-line simulacije je v svojem delu predstavil Kosonen (Kosonen, 1999). Osnovna ideja je v tem, da simulirani promet predstavlja za upravljalno enoto realni promet, hkrati pa upošteva tudi učinke upravljalnih operacij.

Slika 4.1: Upravljanje prometne signalizacije na osnovi on-line simulacije

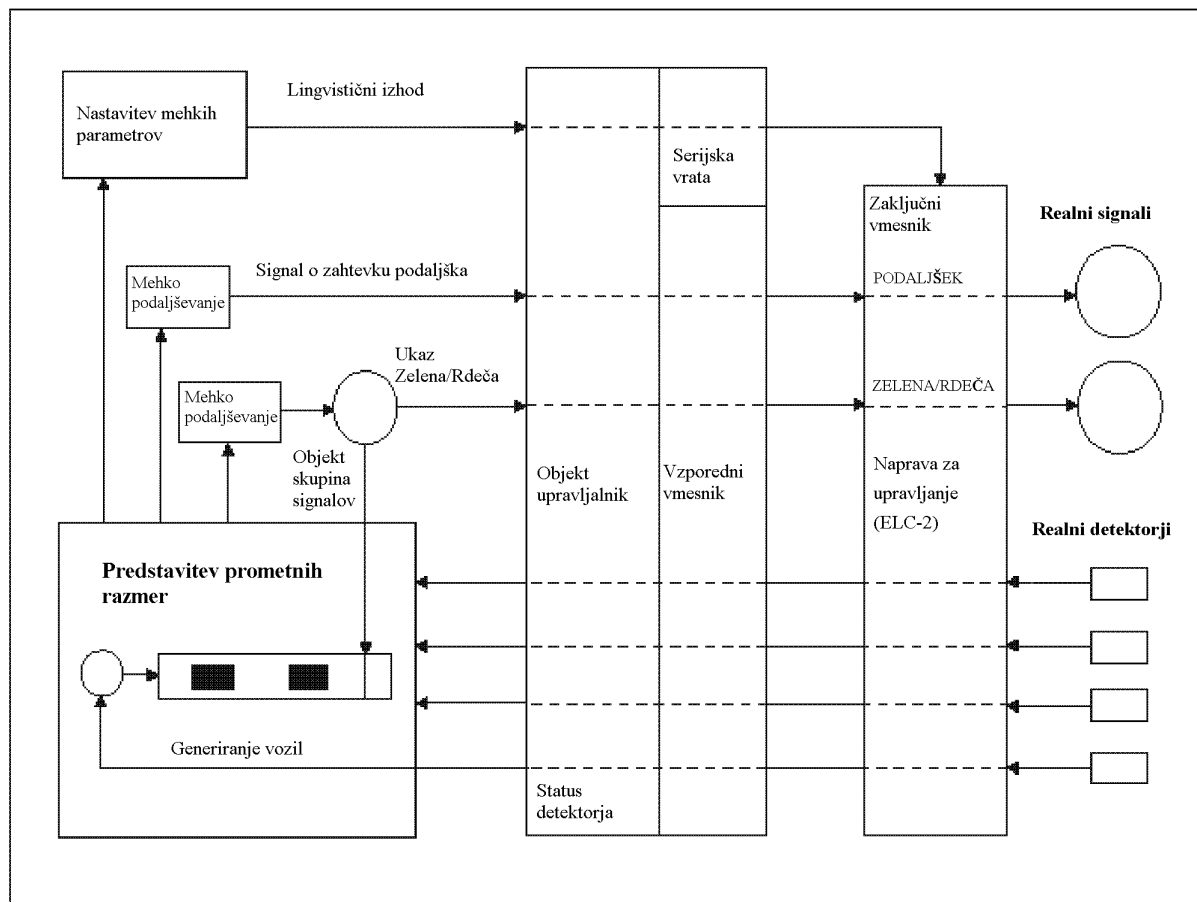


Vir: Niittymäki, 2002, str. 32.

Za čimbolj natančno oceno stanja so uporabljeni detektorji v realnem času (Kosonen in Kokkinen, 1992, str. 252), pa tudi programsko orodje HUTSIM, prirejeno za uporabo v realnem času. V trenutku, ko vozilo prečka detektor, se v simulacijskem modelu generira vozilo, o katerem potem ni več zbranega nobenega podatka. Simulacijski model sam usmerja vozila od točke generiranja do cilja. Ta ideja vsebuje ključni pomen za sistem, da je mogoče upravljati celotno križišče z zelo majhnim številom detektorjev.

V simulacijski sistem je vgrajen mehki upravljalnik signalizacije, ki sprejema vhodne podatke, sprejema mehke odločitve in generira izhodne podatke. Strukturo sistema in potek simulacije prikazuje slika 4.2.

Slika 4.2: Mehki upravljalnik signalizacije in povezave s simulacijskim sistemom



Vir: Niittymäki, 2002, str. 33.

5 Upravljanje prometne signalizacije z uporabo mehke logike

Upravljanje z uporabo mehke logike se je izkazalo kot uspešno zlasti pri problemih, kjer je natančno matematično modeliranje težko ali celo nemogoče, vendar pa uspe tak proces upravljati izkušen operater. Upravljanje prometne signalizacije spada v to kategorijo (Niittymäki, 2000, str. 17). Upravljanje z mehko logiko je posebej primerno pri problemih, kjer nastopa več med seboj konfliktnih ciljev in so informacije podane z lingvističnimi izrazi. Efektiven sistem mora ustrezno uravnovežiti prometno pretočnost in preprečevanje nastajanja dolgih čakalnih vrst. S tem problemom se srečujejo načrtovalci upravljalnih sistemov prometne signalizacije, predvsem pri zapletenih križiščih z mnogimi cestami z več vozišči.

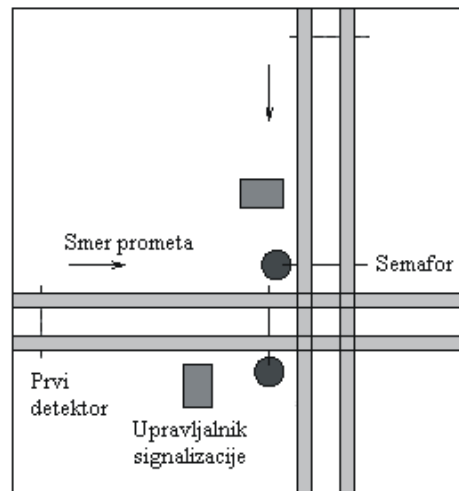
Mehka logika deluje tudi v primeru nepopolnih podatkov, kar je pri pridobivanju podatkov v realnem času velikega pomena, saj v vsakem trenutku lahko nastopi kakšna okvara, nesreča ali le motnja pri komunikaciji.

Mehka logika omogoča manipuliranje z lingvističnimi in ne dovolj točnimi podatki. Predstavlja sredstvo za prevod lingvistične upravljalne strategije, ki je izražena z if-then stavki, v upravljalni algoritem. Mehka logika razume lingvistična navodila in omogoča generiranje upravljalne strategije na osnovi verbalne komunikacije. Motivacija za izgradnjo mehkega upravljalnega sistema je obstoj neposredne povezave med prostimi lingvističnimi izrazi pri upravljalni strategiji in njeno ročno implementacijo. Mehki algoritmi imajo prednost, da se ne nanašajo na matematično funkcijo za formuliranje upravljalne strategije. Namesto te so pri načrtovanju mehkega upravljalnika signalizacije potrebni ekspertno znanje in izkušnje pri upravljanju prometa za formuliranje lingvističnega protokola (Niittymäki, 2002, str. 25).

5.1 Pregled literature o mehkem upravljanju prometne signalizacije

Prvi znani poizkus uporabe mehke logike pri upravljanju prometne signalizacije sta leta 1977 naredila Pappis in Mamdani. Naredila sta teoretično simulacijsko študijo mehkega upravljalnika v izoliranem semaforiziranem križišču dveh enosmernih cest z dvema voziščema in enakim prometnim pretokom (Pappis in Mamdani, 1977). To je tudi ena najstarejših aplikacij mehke logike.

Slika 5.1: Konfiguracija križišča pri poizkusu Pappisa in Mamdanija



Vir: Bingham, 1998, str. 56.

V tem delu sta primerjala svojo mehko metodo s prilagodljivim sistemom upravljanja signalizacije, ki minimizira zakasnitve z optimalno dolžino cikla. Po rezultatih je bil sistem upravljanja z mehko logiko vsaj tako dober ali za malenkost boljši od prilagodljive metode, s katero sta ga primerjala (Niittymäki, 2002, str. 25). Za kvantitativno interpretacijo lingvističnih navodil in za proces odločanja sta uporabila osnovne ideje mehkih množic.

Primer mehkega pravila:

If	T (čas) = srednji	
And	A (prihodi) = vk (srednji)	vk = več kot
And	Q (kolona) = mk (majhna)	mk = manj kot
Then	E (podaljšek) = srednji	

Skupno je 25 pravil, 5 pri vsakem koraku. Vsako pravilo je mehka povezava med vhodnimi spremenljivkami T, A, Q in izhodno spremenljivko E. Glede na to, da vsako mehko pravilo predstavlja 4-dimenzionalna matrika, je v tej aplikaciji mehki algoritem pri vsakem koraku (oz. odločanju o upravljalni akciji) predstavljen z unijo petih takih matrik, saj je pri vsakem odločanju 5 pravil. 25 pravil zagotovi največ 5 korakov, ki se zgodijo v 7., 17., 27., 37. in 47. sekundi. Torej je maksimalni učinkoviti zeleni interval dolg 57 s. Ob vsakem koraku so pravila poklicana 10-krat (za vsako od naslednjih sekund). Pri tem je pomembno, da so detektorji postavljeni dovolj daleč od križišča, da so podatki o prometnih razmerah na razpolago za vsako od naslednjih 10 sekund.

V devetdesetih letih prejšnjega stoletja so se začele vrstiti raziskave in študije uporabe mehke logike pri upravljanju prometne signalizacije. Sodobne mehke sisteme v prometnem inženirstvu je predstavil Teodorovic (Teodorovic, 1999). V njegovem članku je veliko referenc; v tem pregledu omenimo le prilagodljivo upravljanje prometne signalizacije.

Kim je preučeval mehke algoritme v izoliranih križiščih (Kim, 1997). Dolžino zelenega signala je prilagodil razmeram prometa ob koncu vsake faze. Favilla, Machion in Gomide so predstavili dve različni metodi ostrenja in kriterije pri sprejemanju odločitev (Favilla, Machion in Gomide, 1993). Trabia, Kaseko in Ande so predstavili mehki sistem upravljanja prometne signalizacije za izolirano križišče, ki je temeljil na dvonivojski proceduri, ki je odločala o podaljšanju ali prekinitvi faze (Trabia, Kaseko in Ande, 1999). V prvi fazi je sistem ocenil intenzivnost prometa, nato pa rezultat upošteval v drugi fazi, kjer se odloča o prekinitvi ali podaljškju. Sistem je produciral manjše zakasnitve vozil in enak delež ustavljanja kot v primeru metode na zaznavo vozil. Sayers, Anderson in Bell so se ukvarjali z mehkim upravljanjem prometne signalizacije z vidika več ciljev in tudi z ozirom na to, kje naj bi se upravljalni sistem uporabljal (Sayers, Anderson in Bell, 1998). Uporabili so genetski algoritem (MOGA) kot optimizacijsko tehnologijo. Nakatsuyama, Nakahashi in Nishizuka so obravnavali mehko logiko pri upravljanju faz signalizacije dveh zaporednih križišč (Nakatsuyama, Nakahashi in Nishizuka, 1984).

Z vidika mrež križišč določajo upravljanje prometne signalizacije trije parametri: dolžina cikla, dolžina faze in zamik. Chiu je uporabil mehki sistem sklepanja za nastavitvev teh treh parametrov glede na podane lokalne informacije (Chiu, 1992). Blok 40 pravil je upravljal parametre signalizacije mreže 3*3 križišč. Efektivnost je bila prikazana s simulacijami.

S projektom FUSICO na Finskem so Niittymäki in sodelavci v obdobju med letoma 1998 in 2002 naredili splošno analizo sistema upravljanja prometne signalizacije z uporabo mehke logike. Obravnavali so signalizirani prehod za pešce, dvofazno in večfazno upravljanje signalizacije, problem upoštevanja prednosti javnega prevoza in upravljanje signalizacije na večjih glavnih mestnih cestah. Njihove študije temeljijo predvsem na uporabi simulacije in primerjavi sistema mehke logike s sistemom na zaznavo vozil. Sistem upravljanja signalizacije z mehko logiko so testirali tudi na konkretnih križiščih.

Upravljanje prometne signalizacije z uporabo mehke logike se je razvijalo z vidika mehkega sklepanja, kar omogoča modeliranje znanja in izkušenj človeka kot operaterja. Torej je upravljanje signalizacije naloga, primerna za mehko upravljanje.

5.2 Struktura mehkega upravljanja signalizacije

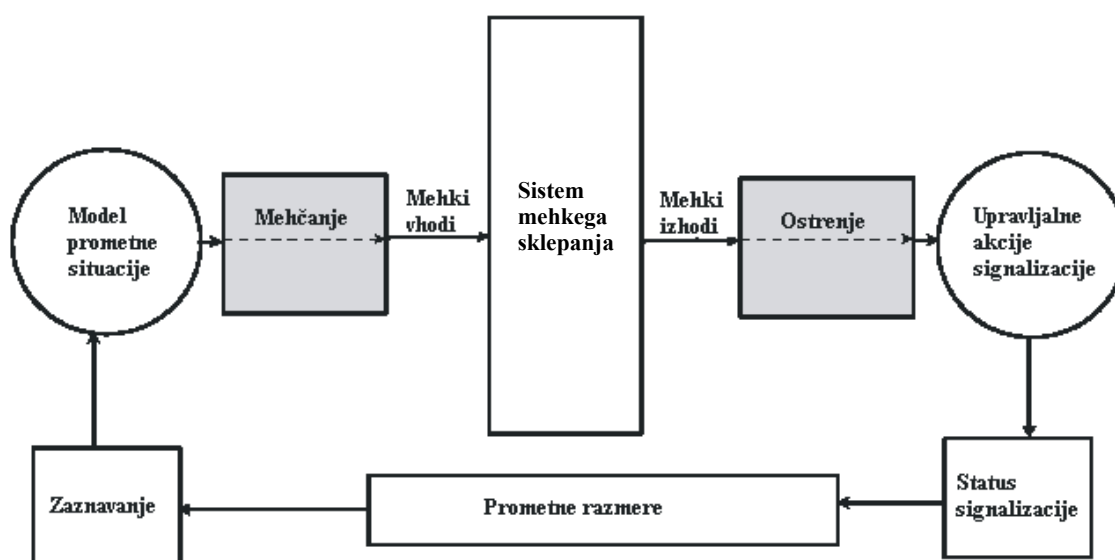
Upravljanje prometne signalizacije zajema neprestano sprejemanje naslednjih odločitev:

- a) zaključiti trenutno fazo in preskočiti na naslednjo najustreznejšo fazo ali
- b) podaljšati trenutno fazo.

Z drugimi besedami, upravljalnik neprestano (ali ob rednih intervalih) zbira informacije, ocenjuje stanje na posameznih voziščih in izbere najprimernejšo opcijo (Niittymäki in Könönen, 2000, str. 3578). Kot večina upravljalnih problemov vključuje ta upravljalni proces naslednje elemente: vhod, proces, izhod, cilj, ocenjevalni kriterij in povratno zanko. Pri upravljanju s povratno zanko vhodi predstavljajo informacije o stanju sistema. Proces je osnovan na sistemu pravil, ki ob danih vseh sprejme odločitev, ali nadaljevati ali zaključiti trenutno fazo. Izhod je napovedana posledica upravljanja procesorja, cilj pa so razmere v mejah tolerance, preden je potrebno zamenjati fazo. Ocenjevalni kriterij in povratna zanka predstavljata proces primerjanja izhoda in cilja, kateremu sledi pošiljanje izhodnih podatkov nazaj, da postanejo novi vhod (Niittymäki, 2002, str. 29).

Mehki upravljalni proces upravljanja prometne signalizacije je sestavljen iz sedmih delov: trenutne prometne razmere s statusom signala, pridobivanje (ostrih) vrednosti merjenih količin (vhod), modeliranje prometnih razmer, mehčanje, sklepanje, ostrenje in upravljalne akcije.

Slika 5.2: Struktura sistema za upravljanje prometne signalizacije



Vir: Niittymäki, 2002, str. 29.

Pri gradnji sistema za mehko upravljanje prometne signalizacije uporabljamo tri glavne metode: izpeljava mehkih pravil, simulacija in meritve na terenu. Potek gradnje sistema pa delimo na 5 korakov: cilji, mere, spremenljivke, mehke množice in pravila. Najprej je potrebno obravnavati cilje in načine, kako jih meriti. V splošnem imamo pri upravljanju prometne signalizacije tri glavne cilje (tekoč promet, varnost, zaščita okolja) in vrsto dejavnikov, ki jim lahko rečemo delni cilji, kot so zakasnitev, čas čakanja, delež ustavljanja, nevarnost trčenja, količina emisij itd. Na osnovi tega lahko izberemo spremenljivke, X za vhod in Y za izhod. Eno izmed pravil, ki sledi, je lahko »če X , potem Y «. Naj bo na primer X čas čakanja pešcev na zeleni signal in Y status signala za pešce (rdeča, zelena). Rdečo sprejmemo, če je čas čakanja kratek, sicer pa moramo, če je čas čakanja dolg, pešcem dati zeleno luč (načelo varnosti). Ko so spremenljivke izbrane, je potrebno definirati mehke množice zanje. Odločimo se lahko, da je čas čakanja pešcev na zeleni signal kratek, dolg ali zelo dolg. Nato je potrebno definirati pripadnostne funkcije za mehke množice vsake spremenljivke. Končno lahko definiramo pravila, ki vhodnim množicam priredijo izhodne množice. Za vsako možno kombinacijo pogojev je potrebno definirati posledico.

Pomembna metoda pri modeliranju prometa je uporaba računalniške simulacije. Simulacije imajo to prednost, da omogočajo nadzor nad spremenljivkami. Podobno kot pri laboratorijskih eksperimentih lahko tudi pri simulaciji izvajamo številne teste z vnaprej izbranimi vrednostmi posameznih spremenljivk. To nam daje nove možnosti testiranja in ocenjevanja strategij pri upravljanju prometne signalizacije.

5.3 Splošnost mehkega upravljanja signalizacije

Pri mehkem upravljanju prometne signalizacije modeliramo akcije policista, ki usmerja promet. Razvoj sistema pravil zato poteka ob sodelovanju izkušenih načrtovalcev prometne signalizacije. S sistematičnim pristopom je možno izpeljati funkcionalni upravljalni algoritem prometne signalizacije za kakršnokoli izolirano križišče (Niittymäki, 2002, str. 35). V tem podpoglavju obravnavamo pet primerov signalizacije in njihove cilje: signalizirani prehod za pešce, dvofazno upravljanje signalizacije za vozila, večfazno upravljanje signalizacije za vozila, vključevanje prednosti javnega prevoza in upravljanje prometne signalizacije na večjih glavnih mestnih cestah.

5.3.1 Signalizirani prehod za pešce

Najenostavnejši med osnovnimi problemi pri upravljanju prometne signalizacije je upravljanje prehoda za pešce. Osnovno vprašanje v tem primeru je (glede na približevanje vozil in glede na željo pešcev, da prečkajo cesto) kdaj zaključiti fazo zelenega signala za vozila in dodeliti zeleno luč pešcem (interval je lahko konstanten). Glavni cilj upravljanja je

pešcem ponuditi možnost varnega prehoda ceste z najkrajšim čakalnim časom, hkrati pa minimizirati nevarnost trčenja vozil, torej minimizirati število približujočih se vozil v trenutku prekinitve zelenega signala. Cilj je tudi, da se pešci ne bi odločali za prečkanje ceste v času zelenega signala za vozila.

Z naraščanjem števila pešcev in vozil postajajo odločitve vse težje. Mehka logika pri tem deluje tako, da uporabi več mehkih pravil hkrati in sprejme odločitve, ki predstavljajo kompromis med cilji.

Opis delovanja signalizacije: ko ni pešcev, ki bi želeli prečkati cesto, poteka faza zelenega signala za vozila, ko pa detektor zazna pešca ali pa ta pritisne na gumb, se začne ocenjevanje prihodov vozil in čakalnega obdobja pešcev. Zeleni signal za vozila se prekine, ko nastopijo razmere, določene z različnimi kriteriji, in tedaj se vklopi zeleni signal za pešce. Sicer pa se nadaljuje zeleni signal za vozila (Niittymäki in Kikuchi, 1998, str. 32).

Vhodni parametri temeljijo na ciljnih upravljanja: čakalni čas pešcev (WT) in število približujočih se vozil (A). Dodamo lahko še dodaten parameter, kot je indikator sproščanja kolone vozil (S – presledek med vozili pri stop liniji), saj ni primerno zaključiti zelenega signala vozilom, medtem ko se kolona še sprošča (metoda enkratnega ustavljanja na križišču) (Niittymäki, 2001a, str. 235).

Formulacija pravil je:

ČE je WT (*kratek / dolg / zelo dolg*) IN A je (*nič / nekaj / mnogo*) IN S je (*nizek / visok*)

POTEM (T -prekini / E -podaljšaj).

Vseh pravil je 18 ($3*3*2$), več o tem pa sledi v podpoglavju 5.4.3.

5.3.2 Dvofazno upravljanje signalizacije za vozila

Dve fazi sta pri signaliziranem križišču najmanjše možno število faz. Takih križišč je tudi največ. Z dvema fazama signalizacija opravlja osnovno nalogo dodeljevanja prednosti vozilom brez upoštevanja pešcev in kolesarjev.

Glavna cilja mehkega upravljanja signalizacije sta:

- določiti dolžino cikla,
- razdeliti cikel na faze.

S primerjanjem prometnih razmer sočasno pri zelenem in rdečem signalu poiščemo najboljši možni trenutek za prekinitev faze. Pri mehki signalizaciji ima faza zelenega signala neko minimalno dolžino. Ob zadostnem povpraševanju je možno zeleni signal po korakih podaljšati. Glede na človeško sklepanje zadoščata za sistem mehke logike pri upravljanju dvofazne signalizacije dve vhodni spremenljivki:

A = število vozil, ki se v trenutku t približujejo zelenemu signalu,

Q = število vozil, ki v trenutku t čakajo na zeleni signal.

Iz tega je razvidna bistvena lastnost mehkega načina upravljanja signalizacije: upoštevajo se prometne razmere iz različnih smeri, torej je poleg vozil, ki se približujejo zelenemu signalu, pomembno tudi, koliko vozil čaka pri rdečem signalu.

Formulacija pravil je podrobno opisana v podpoglavju 5.4.3.

5.3.3 Večfazno upravljanje signalizacije za vozila

Sistem mehke logike za večfazno upravljanje signalizacije deluje na treh nivojih (Niittymäki, 1999, str. 12):

Prvi nivo je ocena prometnih razmer, ki nam pove, če je promet redek, normalen ali zasičen.

Drugi nivo je izbira faze ali sekvence. Glavni cilj je maksimalna kapaciteta s čimmanj intervali med zelenimi signali. Drugi cilj je določitev pravilnega vrstnega reda faz. Osnovni princip je izpustiti fazo, če ni potrebna. Če je vrstni red faz A-B-C-A normalen, je lahko mehka razporeditev faz A-B-A-C-A ali A-C-A-B-C. Pravila so pri štirih fazah bolj zapletena, vendar je načelo enako kot pri treh fazah. Splošno načelo pravil je:

Če je utež $W(f)$ za fazo f zelo visoka, potem bo f naslednja faza.

Če je $W(f_i)$ visoka in $W(f_j)$ nič, potem bo f_i naslednja faza.

Vozila ne smejo predolgo čakati.

Tretji nivo je nivo podaljševanja zelenega signala. Glavna odločitev pri tem je, kako izbrati pravi trenutek za njegovo prekinitev. Potrebno je poiskati prvi možni trenutek za prekinitev faze glede na prometno situacijo. Osnovna ideja je v tem, da ne prekinemo signala med sproščanjem kolone (Niittymäki, 2002, str. 37). To pomeni, da se vozila ustavijo samo enkrat na vsakem križišču. Ta metoda se imenuje tudi metoda podaljševanja zelene.

5.3.4 Vključitev prednosti javnega prevoza

Prednost javnega prevoza je običajno posebej prilagojena križišču ob upoštevanju posebnosti: podaljšanje faze, preklic faze, dodatna faza in hitri cikel (Niittymäki, 2002, str. 37). Javnemu prevozu se dodeli prednost, ko detektorji zaznajo vozilo javnega prevoza.

Osnovna ideja mehkega pristopa pri dodeljevanju prednosti vozilom javnega prevoza je, da zaznava takega vozila požene kombinacijo pravil. Javni prevoz (*JP*) postane najpomembnejša lingvistična spremenljivka v času zahtevanja prednosti javnega prevoza; tukaj gre za čas prehoda vozila javnega prevoza od prvega (zaznavnega) do izhodnega detektorja. Če se približuje več takih vozil naenkrat, je lingvistična spremenljivka manj pomembna, saj lahko vsi avtobusi dobijo prednost.

Glavni cilji sistema pravil za vključitev prednosti javnega prevoza so:

- dodeliti pravilno funkcijo prednosti kot funkcijo trenutka zahtevka po prednosti,
- pravilna odločitev za prednost glede na prometno situacijo križišča,
- minimalne motnje drugih prometnih tokov zaradi uveljavljanja prednosti javnega prevoza.

Z mehko logiko lahko vključevanje prednosti javnega prevoza opišemo bolje kot z običajno binarno logiko (Niittymäki in Mäenpää, 2001, str. 23).

5.3.5 Upravljanje prometne signalizacije na večjih glavnih mestnih cestah

Upravljanje prometne signalizacije na večjih glavnih mestnih cestah je pomemben del upravljanja izolirane signalizacije, vendar ima povsem svojske lastnosti. Seveda upravljanje signalizacije križišča z glavno mestno arterijo daje prednost prometnemu toku vzdolž glavne ceste. Niittymäki je v svoji študiji izbral 4 funkcije (HOVR) takšnega upravljanja (Niittymäki, 2002, str. 37). Prva funkcija H daje prednost glavnemu toku in pripada drugemu nivoju podaljševanja zelenega signala. Ostale tri funkcije, OVR (O – zaznava nesreč, V – spremenljivi čas rumene luči, R – spremenljivi čas rdeče luči), se osredotočajo na varnost v prometu. Funkcija H je povezana z izbiro trenutka za prekinitve faze zelenega signala. Funkcija O pri metodi mehkega upravljanja poudarja varnost, medtem ko funkciji V in R služita optimizaciji upravljanja in ne vplivata na trenutek prekinitve zelene faze. Funkcija V zmanjšuje zakasnitve, funkcija R pa nevarnost večjih nesreč med vozili z glavne in stranske ceste.

5.4 Postopek mehkega upravljanja signalizacije

V sistemu mehke logike za upravljanje prometne signalizacije je vhodnih parametrov malo. Običajno nastopajo prometni pretok (ali število vozil) in časovni parametri. Postopek mehkega upravljanja je razdeljen na mehčanje, sklepanje in ostrenje. Natančneje bomo obravnavali le proceduro podaljševanja faze, saj jo bomo uporabili na praktičnem primeru pri izpeljavi algoritma za konkretno križišče.

5.4.1 Spremenljivke pri mehkem upravljanju signalizacije

Število vozil predstavlja število vozil v okolici obravnavanega križišča, na območju med detektorji. Ena od vhodnih spremenljivk je število vozil A , ki se približujejo križišču v smeri zelenega signala. To spremenljivko lahko uporabimo na več načinov:

- kot skupno število vozil, ki se približujejo iz vseh smeri zelenega signala,
- kot maksimalni A – število vozil na najbolj obremenjenem vozišču,
- kot kombinacijo obeh.

Specifikacija je odvisna od namena in ciljev pravil sklepanja. Spremenljivka A je v osnovi namenjena štetju vozil, ki se približujejo križišču v času zelenega signala, v splošnem pa lahko pomeni tudi število približujočih se vozil v času rumenega ali rdečega signala.

Dolžino kolone pri rdečem signalu opisuje spremenljivka Q . Merjenje vrednosti spremenljivke Q temelji na zaznavi pri oddaljenem detektorju in na oceni prometnega modela (model upočasnitve in model maksimalne upočasnitve). Q je podobna spremenljivki A , le da vključuje vozila v stoječi ali skoraj stoječi koloni ($v < 5$ km/h). Lahko označuje tudi število vozil med detektorjema.

Spremenljivka GAP opisuje sproščanje kolone. GAP je časovni interval od konca zaznavanja prejšnjega vozila do trenutka naslednje zaznave. Zadnja izmerjena vrednost se ohranja. V primeru več vozišč je izbrana minimalna vrednost. Konec zaznave vozila je hkrati začetek ponovne meritve. GAP služi prepoznavanju kolone. Uporablja se tudi indikator sproščanja kolone S (presledek med vozili pri stop liniji).

Čakalni čas (WT) opisuje, koliko časa čaka prvo vozilo v koloni ali pešec. Njegovo merjenje se začne, ko pešec pritisne gumb ali ko prvo vozilo v koloni pripelje do detektorja. Čakalni čas se meri v celih sekundah. Začetek zelenega signala povrne vrednost spremenljivke WT na nič in naslednja prva zaznava povzroči začetek ponovnega merjenja. Vrednost te spremenljivke se lahko uporabi tudi za zahtevo nove faze.

Čas zelenega signala (*GRN*) opisuje dolžino trenutnega zelenega signala. Začetek novega zelenega signala se meri od 0 naprej in se zaključi ob nastopu rumenega signala.

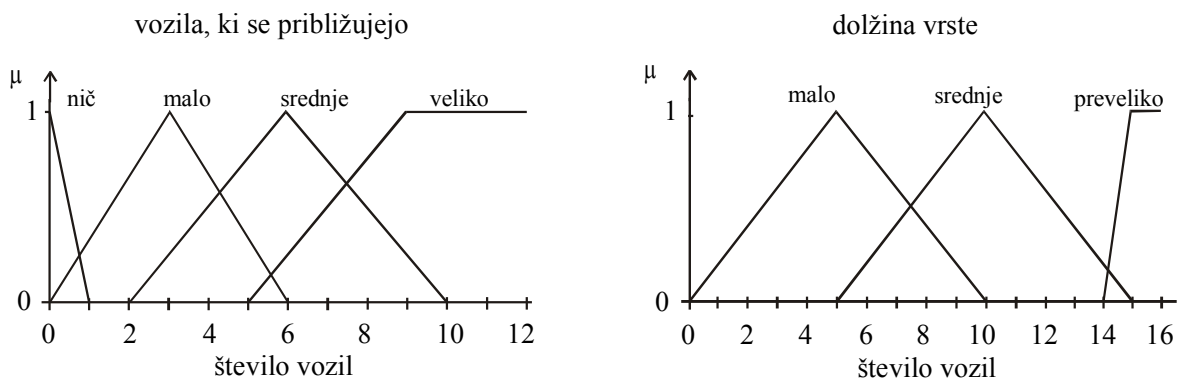
Pri ocenjevanju prometnih razmer se uporablja tudi spremenljivka zasedenost detektorja (*ZD*), ki v odstotkih izraža zasedenost detektorja.

Kot izhodni spremenljivki nastopata dolžina podaljška (*POD*) in spremenljivka prometnih razmer (*PR*), ki služi tudi kot vhodna spremenljivka pri izbiri sistema sklepanja za podaljševanje signala.

5.4.2 Mehčanje

V sistemu mehke logike za upravljanje prometne signalizacije lahko uporabimo lingvistične vrednosti, kot so *dolg*, *kratek*, *srednji*. Izbira lingvističnih spremenljivk in mehkih množic ima velik vpliv na občutljivost sistema upravljanja, vendar pa zanjo ne obstaja natančna metoda. Največkrat se uporablja metoda poizkusov in napak. Primer pripadnostnih funkcij prikazuje slika 5.3. Osnovna ideja je, da za različne smeri lahko uporabimo različne pripadnostne funkcije – odvisno od pretoka. Razlog za različne pripadnostne funkcije je tudi v drugačni postavitvi detektorjev.

Slika 5.3: Primer pripadnostnih funkcij dvofaznega upravljanja signalizacije



Obstajajo metode za optimizacijo pripadnostnih funkcij z nevronskimi mrežami in genetskimi algoritmi, vendar ne dajejo vedno dobrih rezultatov (Bingham, 2001, str. 240). Te metode so se izkazale kot učinkovite v primerih konstantnega pretoka, kot neučinkovite pa v primerih naraščajočega pretoka (Bingham, 1998, str. 90).

Pri mehčanju se v postopku upravljanja pojavljata operatorja *mk* (manj kot) in *vk* (več kot), ki sta za mehko množico *S* definirana (Niittymäki in Pursula, 2000, str. 18):

$$\mu_{mk(S)}(x_i) = \begin{cases} 0, & x_i \geq x_0 \\ 1 - \mu_S(x_i), & x_i < x_0 \end{cases},$$

$$\mu_{vk(S)}(x_i) = \begin{cases} 0, & x_i \leq x_0 \\ 1 - \mu_S(x_i), & x_i > x_0 \end{cases},$$

kjer je x_0 element v S z največjo pripadnostjo.

5.4.3 Sistem sklepanja

5.4.3.1 Pravila

Sistem sklepanja (inferenčni sistem) zajema blok pravil ali več blokov pravil. To so množice pravil, ki podajajo relacije med vhodnimi in izhodnimi spremenljivkami sistema. Množica vseh pravil predstavlja bazo znanja, ki zajema celotno znanje systemskega načrtovalca o procesu upravljanja signalizacije. Torej imajo pravila temeljni pomen za obnašanje sistema in jih je zato potrebno izpeljati čimbolj natančno. To delo pa je lahko časovno zelo zahtevno (Niittymäki, 2001a, str. 233).

Mehki upravljalni algoritem za dvofazno in večfazno signalizacijo, kot je FUSICO, deluje na več nivojih. Prvi nivo ocenjuje prometne razmere *RAZ* (zasičenost, normalne razmere ali razmere nizkega povpraševanja), kjer nastopata dve vhodni spremenljivki:

- prometni pretok *PP* v zadnjih minutah (št. vozil na minuto),
- zasedenost *ZD* prvega detektorja v zadnjih minutah (%).

Mehka pravila so na primer:

- če je $\min(ZD)$ visoka, potem *RAZ* zasičenost,
- če je *PP* nizek in $\max(ZD)$ nizka, potem *RAZ* nizko p.,
- če je $\max(ZD)$ srednja, potem *RAZ* normalne,

kjer je $\min(ZD)$ minimalna zasedenost detektorja, $\max(ZD)$ pa maksimalna zasedenost.

Nivo, ki ureja dolžino cikla in interval zelenega signala oziroma podaljšek *POD* (kratek, srednji, dolg), se poslužuje dveh vhodnih spremenljivk:

- število vozil *A*, ki se bliža zelenemu signalu (nič, malo, srednje, veliko),
- število vozil *Q*, ki čaka pri rdečem signalu (malo, srednje, veliko, preveliko).

Pravila tega nivoja pa so lahko na primer taka:

1. Po minimalni zeleni (5 s):
 - če je A nič, potem je POD nič;
 - če je A malo in Q srednje, potem je POD kratek;
 - če je A več kot malo, potem je POD srednji;
 - če je A srednje, potem je POD dolg.

2. Po prvem podaljšku:
 - če je A nič, potem je POD nič;
 - če je A malo in Q malo, potem je POD kratek;
 - če je A srednje, potem je POD srednji;
 - če je A veliko, potem je POD dolg.

3. Po drugem podaljšku:
 - če je A nič, potem je POD nič;
 - če je A malo in Q malo, potem je POD kratek;
 - če je A srednje in Q manj kot srednje, potem je POD srednji;
 - če je A veliko in Q manj kot srednje, potem je POD dolg.

4. Po tretjem podaljšku:
 - če je A nič, potem je POD nič;
 - če je Q preveliko, potem je POD nič;
 - če je A več kot malo in Q malo, potem je POD kratek;
 - če je A srednje in Q manj kot srednje, potem je POD srednji;
 - če je A veliko in Q manj kot malo, potem je POD dolg.

5. Po četrtem podaljšku:
 - če je A nič, potem je POD nič;
 - če je Q preveliko, potem je POD nič;
 - če je A več kot malo in Q malo, potem je POD kratek;
 - če je A srednje in Q manj kot malo, potem je POD srednji;
 - če je A veliko in Q manj kot malo, potem je POD dolg.

V tem primeru nastopa 5 blokov pravil. Izbira bloka je odvisna od tega, koliko podaljškov je bilo že dodeljenih. Pravila ne pokrivajo nujno celega prostora vhodnih podatkov; za take podatke, ki ne ustrezajo nobenemu pravilu, je izhodni podatek – podaljšek enak 0. Če je podaljšek 0, se faza takoj prekine. Seveda je možno dodati še kaka vzporedna ali varnostna pravila. Glavni cilj pravil podaljševanja je urejanje cikla in faze zelenega signala.

5.4.3.2 Preslikave

Vsak inferenčni sistem lahko predstavimo tudi s preslikavo ali z grafom preslikave. Preslikave služijo več namenom, recimo ugotavljanju učinkovitosti in primerjanju metod ostrenja. Cilj je, da bi sistem posnemal eksperta.

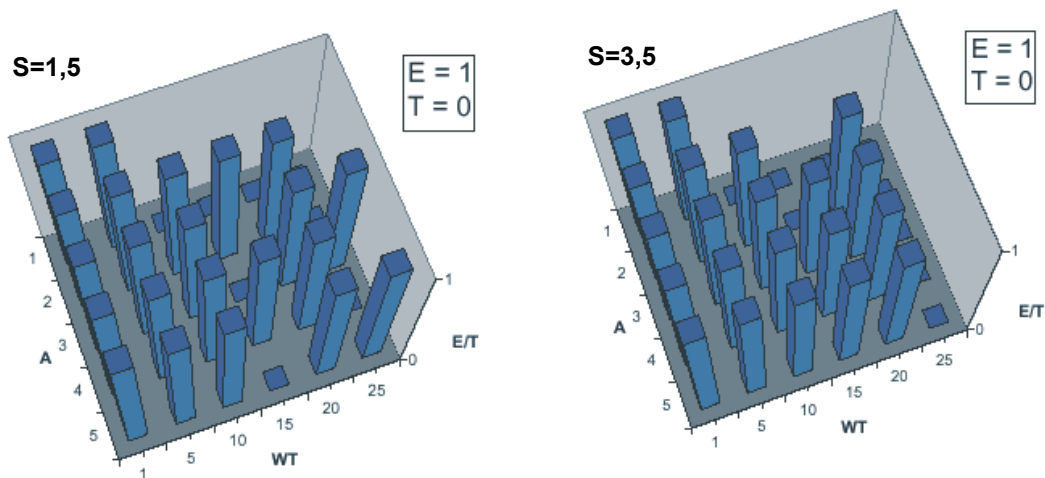
Preslikave za sistem signaliziranega prehoda za pešce prikazuje slika 5.4 (glej str. 52), in sicer za visok in nizek indeks sproščanja kolone. Prvi primer ($S=1,5$) je tipičen primer procesa sproščanja kolone, kjer je zelo težko prekiniti zeleni signal. Le ko je število vozil, ki se približujejo (A), majhno in pešci čakajo že dolgo (WT), lahko nastopi odločitev o prekinitvi. Sicer lahko, ko se kolona sprosti (primer $S=3,5$; A =malo), dosežemo zaključek faze s srednjim časom čakanja (približno 15 s). Običajna metoda upravljanja signalizacije na zaznavo vozil podaljša fazo do maksimalnega intervala zelenega signala, če zaznava vozila ($A>0$).

Preslikave in grafi preslikav so metoda za analizo pravil. Z njimi lahko preverjamo napake pravil ali pa si z njimi pomagamo pri izpeljavi pravil.

Sistem sklepanja dvofaznega upravljanja deluje, kot prikazuje slika 5.5 (glej str. 53). V prvih intervalih je enostavno dobiti srednji podaljšek; število vozil, ki čakajo v koloni, ni tako pomemben faktor. V naslednjih intervalih je podaljšek pri visokem številu približujočih se vozil A daljši. Kasneje, če naraste število čakajočih vozil Q , je podaljšek kratek ali celo enak nič. Kriterij podaljšanja določa razmerje med A in Q .

Rezultati dvofaznega upravljanja pri projektu FUSICO kažejo, da sistem deluje dobro (glej poglavje 7). Pomanjkljivost takega upravljanja pa je, da ni posebne faze nadaljnega podaljšanja do preklica, recimo v primeru, ko sta A in Q enaka nič ali malo.

Slika 5.4: Preslikave za sistem signaliziranega prehoda za pešce

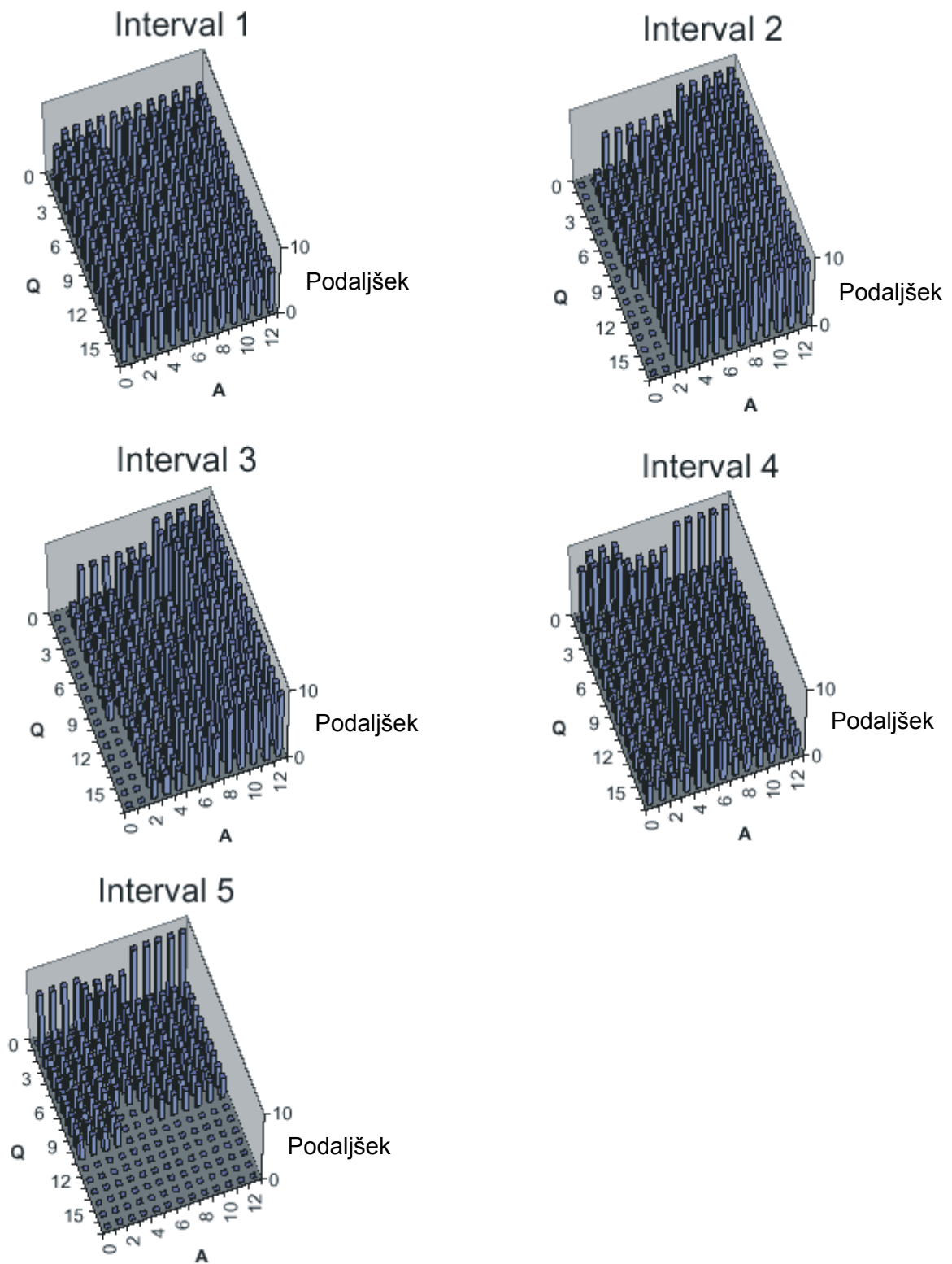


S=1.5		Čakalni čas pešcev (s)					
		1	5	10	15	20	25
Vozila v približevanju (voz.)	0	E(13)	E(13)	E(14)	E(14)	E(14)	T(7)
	1	E(10)	E(10)	T(4)	T(4)	T(5)	T(5)
	2	E(11)	E(11)	E(12)	E(12)	E(12)	T(6)
	3	E(18)	E(17)	E(18)	T(9)	E(18)	E(18)
	4	E(15)	E(17)	E(18)	E(16)	E(18)	T(8)
	5	E(18)	E(17)	E(18)	T(9)	E(18)	E(18)

S=3.5		Čakalni čas pešcev (s)					
		1	5	10	15	20	25
Vozila v približevanju (voz.)	0	T(1)	E(14)	E(14)	T(2)	E(14)	T(3)
	1	E(10)	E(10)	T(4)	T(4)	T(5)	T(5)
	2	E(10)	E(11)	E(12)	T(4)	E(12)	T(5)
	3	E(11)	E(11)	E(12)	E(12)	E(12)	T(6)
	4	E(11)	E(11)	E(12)	E(12)	E(12)	T(6)
	5	E(11)	E(11)	E(12)	E(12)	E(12)	T(6)

Vir: Niittymäki, 2002, str. 41.

Slika 5.5: Preslikave sistema sklepanja za podaljševanje ali prekinitvev faze

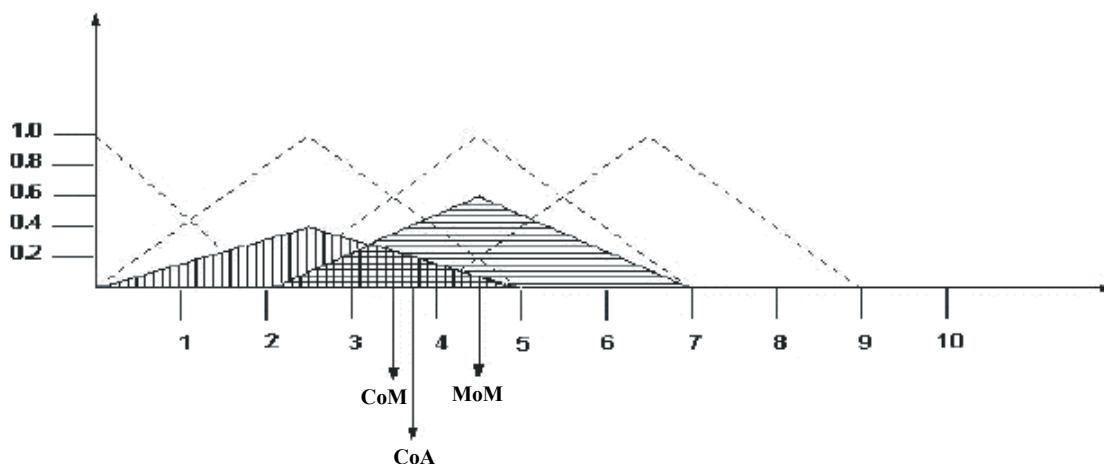


Vir: Niittymäki, 2002, str. 43.

5.4.4 Ostrenje

Različne metode ostrenja dajo različne rezultate. Primer na sliki 5.6 kaže, da ne obstaja ena sama rešitev. Za izbiro primerne metode ostrenja pa je potrebno upoštevati praktične izkušnje in meritve pri testiranju sistema. V splošnem sta najbolj uporabljeni metodi v sistemih upravljanja metoda povprečja maksimumov ali Center of Maximum (CoM) in metoda center ploščin ali Center of Area (CoA) (glej podpoglavje 2.4.4). Ti metodi prideta v poštev tudi pri upravljanju signalizacije, saj naredita kompromis med različnimi podaljški.

Slika 5.6: Primerjava metod ostrenja



Pri mehkem upravljanju prometne signalizacije je možno izpeljati algoritem upravljanja za različne tipe izoliranih križišč, vendar ni posebnih priporočil za mehčanje in ostrenje. Zato so preslikave upravljalnega sistema ključnega pomena pri analizi sistema sklepanja, saj opisujejo realno obnašanje sistema in relatije med vhodi ter izhodi.

5.5 Primerjava mehkega upravljanja signalizacije z drugimi sistemi

Da bi lahko sistem mehkega upravljanja prometne signalizacije čim bolje testirali, je potrebno uporabiti simulacijo. Dober simulacijski program je pri načrtovanju upravljalnega sistema ključnega pomena. Pri projektu FUSICO helsinške univerze so uporabili simulacijski sistem HUTSIM, ki so ga razvili na isti univerzi. Trenutna verzija programa ima že vgrajeno osnovno metodo mehke logike za upravljanje signalizacije (HUTSIM – spletna stran, 2004).

Simulacijski sistem nam omogoča primerjanje raznih sistemov upravljanja prometne signalizacije. Niittymäki je v svojem delu (Niittymäki, 2002, str. 45) primerjal prej opisane primere izolirane signalizacije glede na različne metode upravljanja, kot sta metoda na zaznavo vozil (običajna in modificirana) in metoda mehke logike. Modificirana metoda na zaznavo vozil po zaznavi pešca dodeli le en podaljšek. Običajna metoda na zaznavo vozil pa predvidi vnaprej definiran maksimalni podaljšek (30 s).

Pri signaliziranem prehodu za pešce so obravnavali različne pretoke pešcev: 15 pešcev/uro, 50 pešcev/uro in 150 pešcev/uro. Ugotovili so, da je pri mehkem upravljanju povprečna zakasnitev pešca bistveno manjša kot pri konvencionalni metodi na zaznavo vozil, ampak nekoliko večja kot pri modificirani metodi na zaznavo vozil. Pri vozilih je zakasnitev v vseh primerih približno enaka, nekoliko večja pri modificirani metodi na zaznavo vozil in najmanjša pri mehki logiki. Rezultati temeljijo na več kot 120 urah simulacij.

Pri dvofaznem upravljanju prometne signalizacije rezultati simulacij kažejo, da je področje aplikacij mehke logike zelo široko. Pri primerjanju upravljalnih metod so opravili identične simulacije za vse metode. Vozila so bila generirana v istih trenutkih, sprememba v njihovem obnašanju pa je bila rezultat različnih upravljalnih metod. Rezultati so pokazali, da je, kar se tiče zakasnitev, metoda običajnega podaljševanja kar dobra metoda upravljanja signalizacije pri zelo nizkih prometnih pretokih in da je mehka logika konkurenčna alternativa. Kapaciteta je pri upravljanju na zaznavo vozil nekoliko večja kot pri mehki logiki zaradi daljših ciklov.

Pri večfaznem upravljanju signalizacije so opravili primerjavo metode na zaznavo vozil in metode mehke logike na pravem križišču v Helsinkih. Rezultati simulacij so pokazali, da je mehko upravljanje konkurenčno tradicionalni metodi upravljanja. Efektivnost so merili s povprečnimi zakasnitvami in z deležem ustavljanja. V primerih zelo nizkega prometnega pretoka je še vedno boljši sistem na zaznavo vozil. V primeru, ko je glavni pretok prometa večji od 500 vozil/uro (to je približno 8 vozil/minuto), so bili rezultati mehkega upravljanja vsaj tako dobri kot rezultati običajnega upravljanja. Glede na terenske meritve je bil pretok na glavni cesti 600 – 900 vozil/uro in razmerje pretokov glavne in stranske ceste 5:1. Na osnovi tega lahko trdimo, da je princip mehkega upravljanja konkurenčen za izolirane večfazne sisteme prometne signalizacije.

Boljše rezultate pri mehkem upravljanju ob nizkih pretokih prometa je moč doseči z uporabo mehkega izbiranja faze. Simulacije so temeljile le na odločanju o podaljševanju signalov. Glavni cilj mehkega izbiranja faze je maksimizacija kapacitete z minimizacijo intervalov med zelenimi signali.

Pri signalizaciji na večjih glavnih mestnih cestah so primerjali mehko upravljanje, upravljanje na zaznavo vozil in metodo fiksnih časovnih intervalov. Kar se tiče zakasnitev, so različne metode dale ekvivalentne rezultate, glede števila ustavljanj pa se je najbolje izkazala metoda

mehke logike. Metodo mehke logike so primerjali tudi s švedsko metodo matematične optimizacije SOS. Rezultati kažejo, da enostaven sistem mehke logike deluje na istem nivoju kot kompleksna SOS metoda. Slednja pa daje veliko boljše rezultate v prometnih konicah.

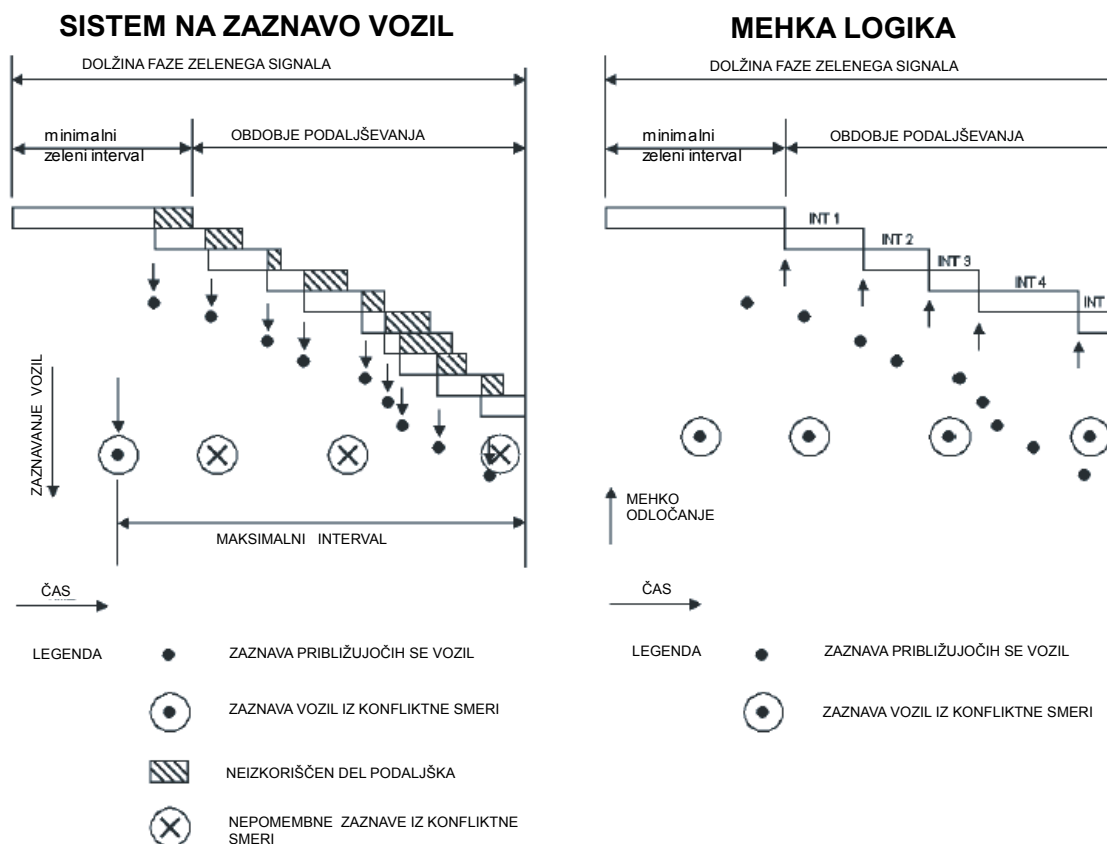
Po rezultatih testiranja s simulacijami se je metoda mehkega upravljanja prometne signalizacije izkazala kot potencialna metoda za izolirana križišča. Področje aplikacije je široko, običajna metoda na zaznavo vozil pa je še konkurenčna, posebej v primerih zelo nizkega prometnega pretoka ali prenasičenosti. To pomeni, da potrebujemo vsaj 2 do 3 različne mehke upravljalne algoritme za dnevno upravljanje signalizacije, kar pa je še vedno manj kot pri klasičnem upravljanju na zaznavo vozil.

Pomembno pa je poudariti dejstvo, da upravljanje z mehko logiko temelji na podatkih o številu vozil iz obeh smeri. Pri običajnem upravljanju na zaznavo vozil je podaljševanje signala daljše kot pri mehkem upravljanju. Pri običajnem principu podaljševanja se dodeli podaljšek pri zaznavi vozila, pri mehkem upravljanju pa lahko aktiven signal prekinemo, če je kolona vozil v konfliktni smeri dovolj dolga ali predolga (slika 5.7, str. 57). Z vidika varnosti je običajna metoda podaljševanja verjetno boljša, saj so cikli daljši; s tem je vozil v opcijski coni na posamezni cikel manj.

Torej je glavna razlika med upravljalnima algoritmoma način, kako se sprejme odločitev o podaljškju. Pri običajnem podaljševanju odločitve temeljijo le na prometni situaciji v smeri zelenega signala, dodeljeni podaljški pa so konstantni. Obe metodi sicer predvidevata nek minimalni interval za sprostitev kolone. Metoda mehke logike predstavlja bolj demokratičen način upravljanja, saj upošteva podatke o prometni situaciji iz smeri zelenega in tudi rdečega signala. S tem spada metoda mehke logike med večobjektivne metode upravljanja.

Pri načrtovanju sistema mehke logike za upravljanje signalizacije je potrebno ekspertno znanje za formulacijo lingvističnega protokola. Mehki upravljalni sistemi zajemajo ključne faktorje procesov, ki jih upravljajo, in ne potrebujejo mnogih natančnih matematičnih funkcij. Zaradi tega imajo veliko prednosti pri aplikacijah v realnem času. Upravljalniki imajo enostavno strukturo, saj ne zahtevajo mnogo numeričnih kalkulacij. Logika na osnovi »if-then« pravil pa tudi ni časovno zahtevna. Če sistemsko znanje zajamemo z mehкими pravili, je tak sistem upravljanja tudi enostavnejši.

Slika 5.7: Primerjava principov podaljševanja pri metodi na zaznavo vozil in metodi mehke logike



Vir: Niittymäki, 2002, str. 53.

Rezultati, pridobljeni z meritvami na terenu, kažejo, da mehki upravljalni algoritmi v večini primerov delujejo bolje od upravljanja po metodi na zaznavo vozil (Niittymäki, 2002, str. 58). Povprečni časi prehodov so približno 4 – 10 sekund krajši, delež ustavljanja je 2 – 12% manjši, zakasnitve avtobusov so krajše v osmih od devetih primerov, simulacije pa kažejo tudi manjšo porabo goriva in manj emisij. Vsi ti rezultati dokazujejo, da lahko mehke upravljalne algoritme uspešno uporabljamo na pravih križiščih. Po drugi strani so zaradi krajših ciklov (povprečno do 8 sekund) na boljšem tudi pešci, saj to pomeni manj čakanja.

Sistemi mehke logike za upravljanje prometne signalizacije lahko delujejo v realnosti. Seveda pa je pred njihovo instalacijo potrebnih veliko malih sprememb in popravkov; do tedaj lahko govorimo le o prototipu.

6 Primer uporabe mehke logike pri upravljanju signalizacije izbranega križišča

V tem poglavju bomo pridobljeno znanje iz mehke logike in mehkega upravljanja prometne signalizacije v križišču uporabili na konkretnem križišču. Izbrano križišče je potrebno najprej analizirati, ugotoviti pretoke vozil in trenutni upravljalni postopek. Nato se lahko lotimo modeliranja sistema za mehko upravljanje, tako da definiramo vse osnovne sestavne elemente sistema. V dobljeni prototip nato prenesemo bazo znanja v obliki pripadnostnih funkcij in pravil. Sistem na koncu še optimiziramo, tako da bo ustrezno vračal pričakovane rezultate.

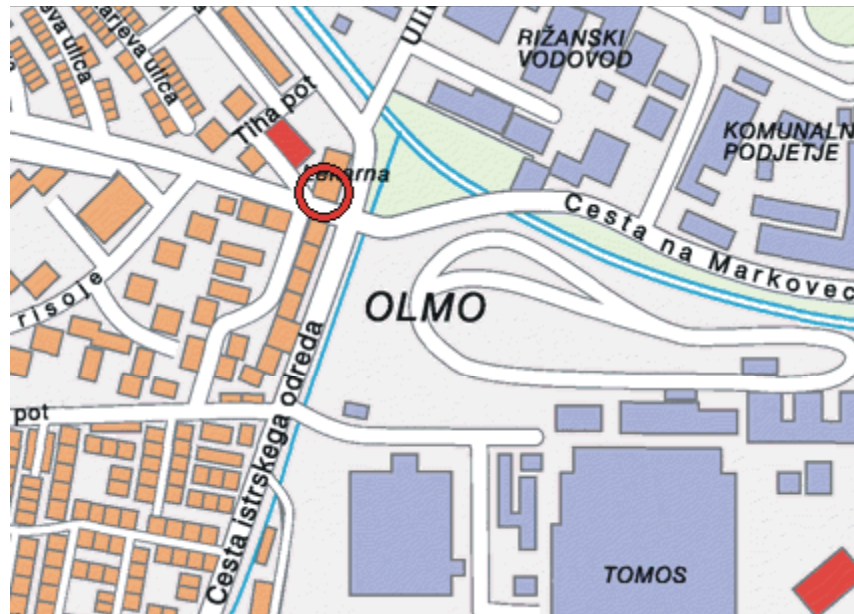
6.1 Analiza problema

Kot obravnavani primer pri upravljanju signalizacije z mehko logiko smo izbrali križišče v Kopru, ki se nahaja v predmestju Olmo blizu nakupovalnega središča. Gre za izolirano križišče dveh dvosmernih cest z enostavno dvofazno signalizacijo, ki deluje s fiksnimi časovnimi intervali.

Slika 6.1: Slika izbranega križišča v Kopru



Slika 6.2: Lokacija izbranega križišča



Vir: Elektronski Telefonski imenik Slovenije 2002.

To križišče smo izbrali iz naslednjih razlogov:

1. Gre za manjše križišče, kar enemu samemu opazovalcu omogoča lažji posnetek stanja, odvijanje prometa lahko posnamemo z eno kamero in tudi analiza sistema je lažja.
2. Križišče je izolirano.
3. Gre za križišče dveh cest s približno enakim prometnim pretokom: približno 590 in 680 vozil/h (vsaj ob opazovanem času).
4. Lokacija je v predmestju, prisotni so tudi prehodi za pešce.
5. Signalizacija križišča je dvofazna, dolžina faz pa fiksna.

Za analizo križišča smo uporabili metodo meritev na terenu s pomočjo videokamere. Posnetki so bili opravljeni 2. 2. 2004 od 12:00 do 12:30. Opazovali smo naslednja parametra: dolžina kolone in število vozil, ki se približujejo zelenemu signalu. Opazovali smo vsa vozišča na območju približno 30 m pred križiščem.

V križišču se sekata Cesta na Markovec in Cesta Istrskega odreda. Opazovanje prometa skozi križišče je pripeljalo do naslednjih ugotovitev: čeprav je Cesta na Markovec nekoliko pomembnejša (in zato tudi širša) od druge ceste, je bil prometni pretok ob opazovanem času

približno enak (8:7). K temu najbrž prispeva bližina nakupovalnega središča, proti kateremu vodi Cesta Istrskega odreda. Dolžina cikla je 60 s, dolžina zelene faze (faza 1) za Cesto na Markovec 30 s, medtem ko je dolžina zelene faze za Cesto Istrskega odreda 20 s (faza 2). Vmes nastopa rumeni signal po 5 s. Dolžina druge faze je bila v opazovanem času kar optimalna, dolžina prve faze pa v vsakem primeru predolga. To je povzročilo, da so vozila na Cesti Istrskega odreda čakala na zeleno luč več sekund, medtem ko iz druge smeri ni bilo nobenega vozila. Vozila so čakala tudi do 15 sekund, to pa je celo polovica časa, kolikor traja signal. S sistemom mehke logike za upravljanje signalizacije bi lahko pripomogli, da do takih situacij ne bi prišlo.

Slika 6.3: Slika situacije, ko kolona vozil po nepotrebnem čaka



Pri mehkem sistemu upravljanja signalizacije korakoma podaljšujemo interval zelenega signala. Vprašanje je, za koliko podaljškov se odločimo. Glede na to, da promet v opazovanem obdobju ni bil zelo gost, se zaradi enostavnosti in preglednosti odločimo le za dva podaljška.

Za natančnejšo analizo križišča bi potrebovali veliko več časa in pripomočkov, vendar bomo glede na naše možnosti problem poenostavili in obravnavali le pol ure opazovanega prometa skozi to križišče.

6.2 Modeliranje sistema mehkega upravljanja signalizacije

Za konstrukcijo sistema mehke logike smo uporabili program *fuzzyTECH 5.52*. Orodje sestavlja grafična lupina, ki omogoča pregled definicij vseh komponent sistema in obenem ponuja tudi vse urejevalne možnosti. Urejevalni del je bogat in razdeljen po značilnostih sistema mehke logike. Tako strukturirani urejevalnik postavlja, prikazuje in dokumentira strukturo zgradbo sistema. Urejevalnik spremenljivk postavlja in prikazuje lingvistične spremenljivke s pripadajočimi pripadnostnimi funkcijami. Urejevalnik pravil je na razpolago v seznamski ali matrični obliki, kar omogoča tudi dober vizualni pregled kompleksnih pravil. Na razpolago so še urejevalnik z izbiro operatorjev in razni grafični prikazovalniki.

Grafična lupina omogoča interaktivno delo in vizualizacijo informacijskih dogajanj v sistemu mehke logike. Vizualno prikazana informacijska obdelava pa teče v okviru odstranjevanja napak.

Potek gradnje sistema lahko razdelimo na 5 korakov:

- postavitve cilja,
- izbira lingvističnih spremenljivk,
- definicija strukture sistema,
- definicija mehkih množic in
- izpeljava pravil.

Kot cilj upravljanja signalizacije smo postavili zahtevo po tekočem prometu s čimmanj ustavljanji in čimmanj čakanja.

Sistem vsebuje dve glavni vhodni spremenljivki:

A – število približujočih se vozil,

Q – število vozil v koloni.

Da bi dobili sistem, ki je dokaj enostaven in pregleden, smo se odločili za samo dva podaljška, saj je tudi dolžina trenutnih faz razmeroma kratka in bi bilo recimo 5 podaljškov veliko.

Zaradi potreb algoritma uvedemo še tretjo vhodno spremenljivko N , s katero ugotavljamo, pri katerem podaljšku smo.

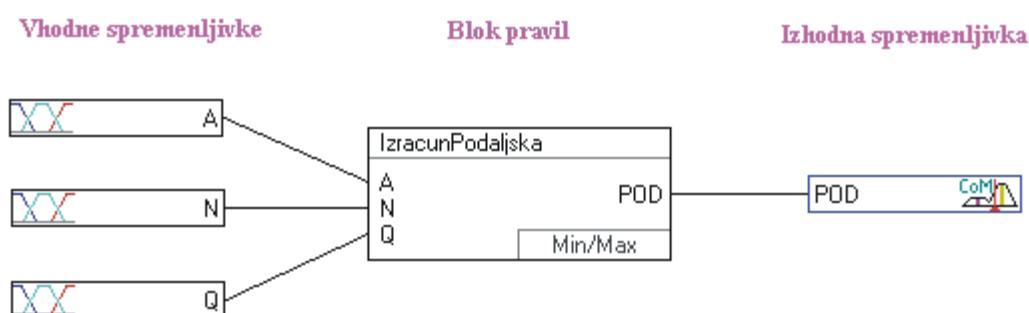
Struktura sistema predstavlja potek sklepanja od vhodnih spremenljivk proti izhodni spremenljivki. Mehčanje vhodnih spremenljivk prevede vhodne numerične podatke v mehke lingvistične vrednosti. Sistem sklepanja se odvija v bloku pravil, ki vsebuje lingvistična

navodila. Izhodne vrednosti bloka pravil so lingvistične. Nato ostrenje v izhodni spremenljivki prevede rezultat v numerično vrednost izhodne spremenljivke.

Naslednja slika prikazuje strukturo sistema. Povezave med njegovimi elementi predstavljajo pretok podatkov skozi sistem.

Slika 6.4: Struktura sistema mehke logike za upravljanje prometne signalizacije

Upravljalnik prometne signalizacije



6.2.1 Definicija lingvističnih spremenljivk

V tem podpoglavju so navedene definicije vseh lingvističnih spremenljivk in pripadnostnih funkcij sistema. Lingvistične spremenljivke moramo definirati za vse vhodne in izhodne spremenljivke. Pripadnostne funkcije so standardnih trikotnih oblik in so definirane z definicijskimi točkami.

Naslednje slike in tabele vsebujejo vse spremenljivke sistema, njihove pripadnostne funkcije in lastnosti. Po zgledu sistema sklepanja iz projekta FUSICO so poleg lingvističnih vrednosti *malo*, *srednje* in *veliko* vpeljane še lingvistične vrednosti *več kot malo*, *manj kot malo* in *manj kot srednje* (glej podpoglavje 5.4.2).

Tabela 6.1: Vhodne spremenljivke

#	Ime spremenljivke	Enota	Min	Max	Privzeta vrednost	Lingvistične vrednosti
1	A	vozila	0	12	0	nič malo večkotmalo srednje veliko
2	N	podaljšek	1	2	1	p1 p2
3	Q	vozila	0	12	0	manjkotmalo malo manjkotsrednje srednje preveliko

Tabela 6.2: Izhodna spremenljivka

#	Ime spremenljivke	Tip ostrenja	Enota	Min	Max	Privzeta vrednost	Lingvistične vrednosti
4	POD	CoM	sekunda	-1	11	0	nič kratek srednji dolg

Privzeta vrednost izhodne spremenljivke se uporabi v primeru, ko nobeno pravilo ne da rezultata za to spremenljivko.

Število približujočih se vozil *A* predstavlja število vseh vozil, ki se približujejo križišču v smeri zelenega signala, šteto na vseh voziščih skupaj.

Slika 6.5: Pripadnostne funkcije vhodne spremenljivke *A*



Tabela 6.3: Definijske točke pripadnostnih funkcij spremenljivke A

Lingvistične vrednosti	Oblika	Definijske točke (x, y)		
nič	linearna	(0, 1)	(1, 0)	(12, 0)
malo	linearna	(0, 0)	(3, 1)	(6, 0)
srednje	linearna	(0, 0)	(3, 0)	(6, 1)
večotmalo	linearna	(0, 0)	(3, 0)	(6, 1)
veliko	linearna	(0, 0)	(6, 0)	(9, 1)

S spremenljivko N sledimo, pri katerem podaljšku smo.

Slika 6.6: Pripadnostne funkcije vhodne spremenljivke N

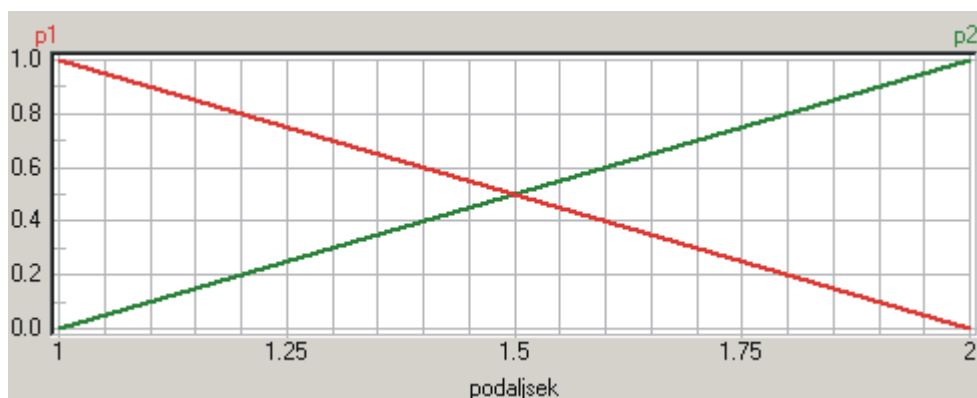


Tabela 6.4: Definijske točke pripadnostnih funkcij spremenljivke N

Lingvistične vrednosti	Oblika	Definijske točke (x, y)	
p1	linearna	(1, 1)	(2, 0)
p2	linearna	(1, 0)	(2, 1)

Število vozil v koloni Q predstavlja število vseh vozil, ki čakajo na zelen signal, šteto na vseh voziščih skupaj.

Slika 6.7: Pripadnostne funkcije vhodne spremenljivke Q

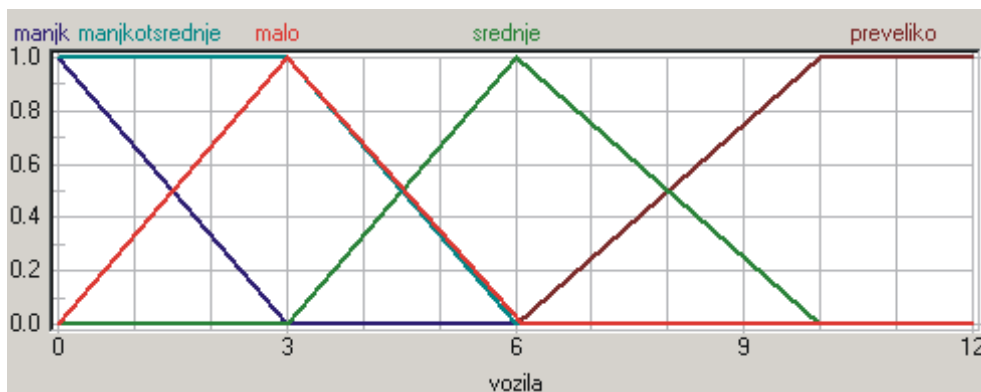


Tabela 6.5: Definijske točke pripadnostnih funkcij spremenljivke Q

Lingvistične vrednosti	Oblika	Definijske točke (x, y)
malo	linearna	(0, 0) (3, 1) (6, 1) (12, 0)
manjkotsrednje	linearna	(0, 1) (3, 1) (6, 0) (12, 0)
srednje	linearna	(0, 0) (3, 0) (6, 1) (10, 0) (12, 0)
manjkotmalo	linearna	(0, 1) (3, 0) (12, 0)
preveliko	linearna	(0, 0) (6, 0) (10, 1) (12, 1)

Podaljšek POD pove, za koliko sekund naj podaljšamo zeleno fazo.

Slika 6.8: Pripadnostne funkcije izhodne spremenljivke POD

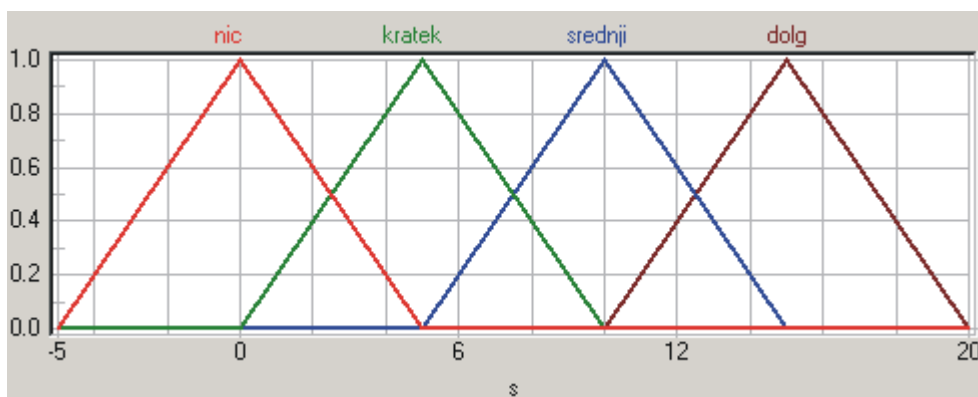


Tabela 6.6: Definijske točke pripadnostnih funkcij spremenljivke *POD*

Lingvistične vrednosti	Oblika	Definijske točke (x, y)		
nič	linearna	(-5, 0) (20, 0)	(0, 1)	(5, 0)
kratek	linearna	(-5, 0) (10, 0)	(0, 0) (20, 0)	(5, 1)
srednji	linearna	(-5, 0) (15, 0)	(5, 0) (20, 0)	(10, 1)
dolg	linearna	(-5, 0) (20, 0)	(10, 0)	(15, 1)

6.2.2 Konstrukcija sistema sklepanja

Sistem sklepanja je zajet v enem bloku pravil. Blok pravil vsebuje logiko upravljalne strategije sistema. Prvi, vhodni del pravil, opisuje situacijo, za katero so pravila načrtovana, drugi, izhodni del pravil, pa opisuje odziv sistema mehke logike na situacijo. Utež pravila (DoS) daje večji ali manjši pomen pravilu. V našem primeru so vsa pravila enako obtežena z utežjo 1, torej so enakovredna.

Obdelovanje pravil se začne z agregacijo. Pri tem je bil uporabljen operator minimum kot posplošitev lingvističnega veznika IN. Implikacija uporablja običajno množenje, pri akumulaciji pa je uporabljen operator maksimum.

Tabela 6.7: Statistika bloka pravil za izračun podaljška

Parametri

Agregacija	minimum
Akumulacija	maksimum
Število vhodov	3
Število izhodov	1
Število pravil	11

Kot cilj si postavimo, da je promet tekoč, kar pomeni čim krajše čakanje in čim redkeje ustavljanje. Če se zelenemu signalu približuje malo vozil, dodelimo kratek podaljšek, če se približuje srednje veliko vozil, dodelimo srednje dolg podaljšek, če pa je vozil veliko, dodelimo dolg podaljšek. Tako logiko lahko praktično dobesedno prenesemo v sistem sklepanja.

Pri dodeljevanju prvega podaljška število vozil v koloni ne vpliva na podaljšek. Nanj vpliva le število vozil, ki se približuje. Pri drugem podaljšku pa upoštevamo tudi, koliko vozil čaka v koloni pri rdečem signalu. Recimo, če je teh preveč, se faza takoj prekine in ni drugega

podaljška. S spremenljivko N sledimo, pri katerem podaljšku smo, oziroma, za kateri podaljšek se odločamo.

Sistem sklepanja je konstruiran po modelu splošnega sistema sklepanja za upravljanje signalizacije iz projekta FUSICO (glej podpoglavje 5.4.3.1). Odločili smo se le za dva podaljška, zato so nekatera pravila prirejena temu, obenem pa tudi potrebam obravnavanega primera.

Tabela 6.8: Pravila za izračun podaljška

IF			THEN	
A	N	Q	DoS	POD
malo	p1		1.00	kratek
srednje	p1		1.00	srednji
nič	p1		1.00	nič
veliko	p1		1.00	dolg
nič	p2		1.00	nič
	p2	preveliko	1.00	nič
srednje	p2	manjkotsrednje	1.00	srednji
veliko	p2	manjkotsrednje	1.00	srednji
večkotmalo	p2	malo	1.00	kratek
malo	p2	manjkotsrednje	1.00	kratek
veliko	p2	manjkotmalo	1.00	dolg

Tabela 6.9: Statistika celotnega sistema

Vhodne spremenljivke	3
Izhodne spremenljivke	1
Bloki pravil	1
Pravila	11
Pripadnostne funkcije	16

Primer:

Oglejmo si, kako deluje sistem v primeru vhodnih podatkov $A = 5$ in $N = 1$. Torej se odločamo o dolžini prvega podaljška, v smeri zelenega signala pa se približuje 5 vozil. Po pripadnostnih funkcijah spremenljivke A (slika 6.5, str. 63) je 5 *malo* vozil s stopnjo pripadnosti $1/3$ in *srednje* veliko vozil s stopnjo pripadnosti $2/3$. Za spremenljivko N pa 1 pripada množici $p1$ s stopnjo 1 (slika 6.6, str. 64). Tem podatkom ustrezata pravili št. 1 in 2. Rezultat mehkega sklepanja je: podaljšek je *kratek* s stopnjo $1/3$ in *srednji* s stopnjo $2/3$. Po postopku ostrenja z metodo CoM dobimo, glede na pripadnostne funkcije spremenljivke POD , numerični rezultat

$$\frac{\frac{1}{3} \cdot 5 + \frac{2}{3} \cdot 10}{\frac{1}{3} + \frac{2}{3}} = 8,3 \text{ oziroma } 8 \text{ celih sekund.}$$

6.3 Optimizacija

Z analizo posnetkov ugotovimo, koliko naj bi bil dolg *kratek*, *srednji* ali *dolg* podaljšek.

Podaljšek *nič* je dolg 0 sekund. To pomeni takojšnjo prekinitév.

Kratek podaljšek je dolg toliko, da gre *malo* vozil še čez križišče.

Srednji je dolg toliko, da gre *srednje* veliko vozil še čez križišče.

Dolg podaljšek pa omogoča, da gre *veliko* vozil čez križišče.

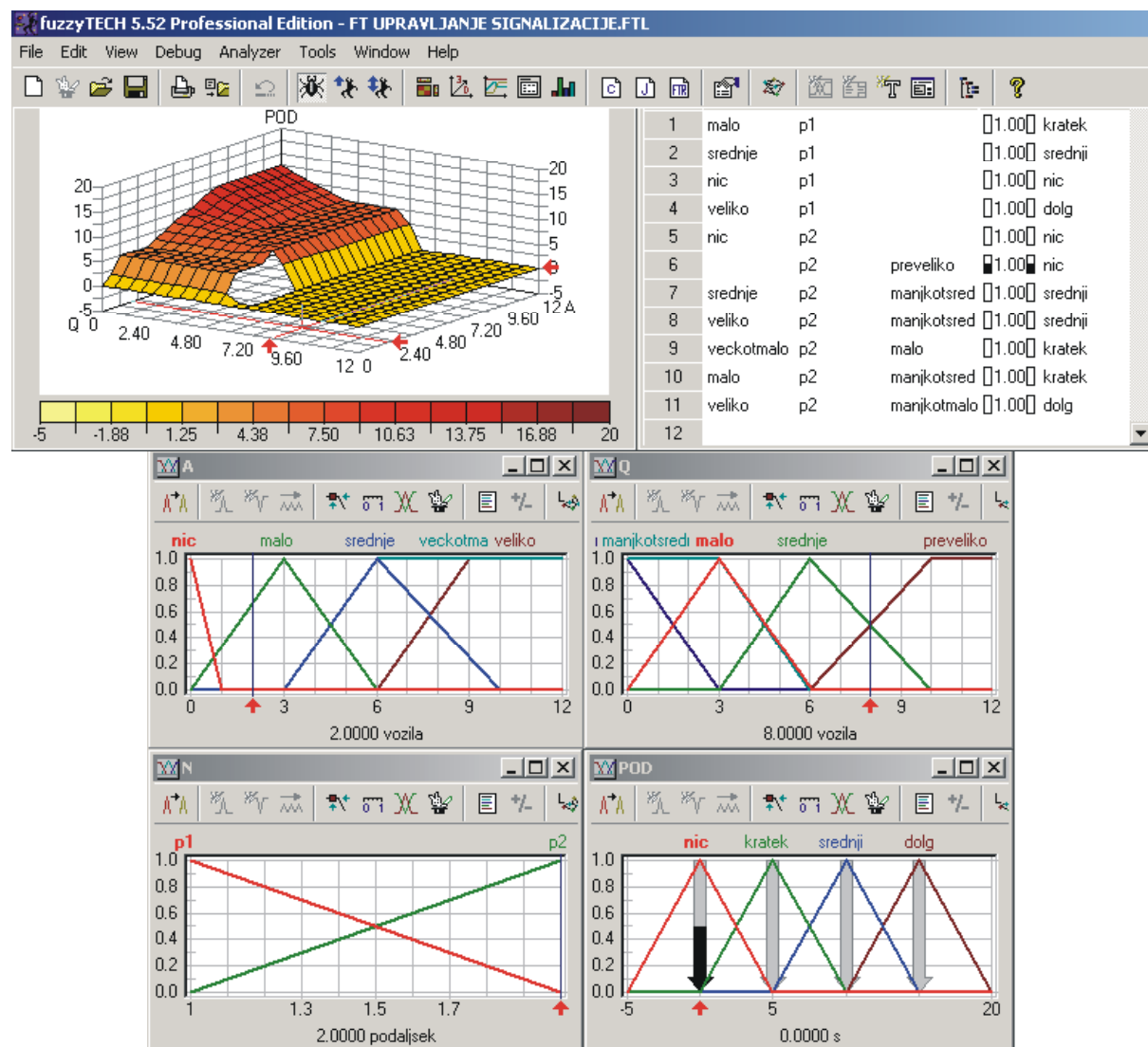
Pomemben podatek je tudi razdalja, na kateri naj bi bili detektorji, ki vpliva na relativni pomen lingvističnih vrednosti *malo*, *srednje*, *veliko* in *preveliko* ter na dolžino temu ustreznega podaljška. Namreč, dlje kot je prvi detektor, več časa potrebujejo zaznana vozila, da še uspejo priti čez križišče. Vzemimo, da je v našem primeru na vseh voziščih postavljen detektor vozil na razdalji 30 m pred križiščem. Detektorjev trenutno ni, zato lahko za naše potrebe poljubno določimo razdaljo, v praksi pa bi to preverili ali analizirali še s pomočjo simulacije.

Ko je postavljena struktura sistema in so definirani tudi vsi njegovi elementi, je potrebno tako imenovani prototip testirati in preveriti, če ustreza podatkom in če daje željene rezultate. Optimizacija je lahko zelo dolg postopek, še posebej, če si želimo, da bi se sistem dobro obnašal v vseh pogojih. Toda glede na to, da smo naš problem omejili na precej konstantne razmere zaradi zelo kratkega opazovalnega obdobja, in glede na to, da nimamo na razpolago dovolj dobrega simulacijskega orodja, s katerim bi lahko temeljito testirali sistem, je optimizacija enostavnejša. Sistem smo optimizirali le toliko, da je dobro definiran in da nam daje željene rezultate.

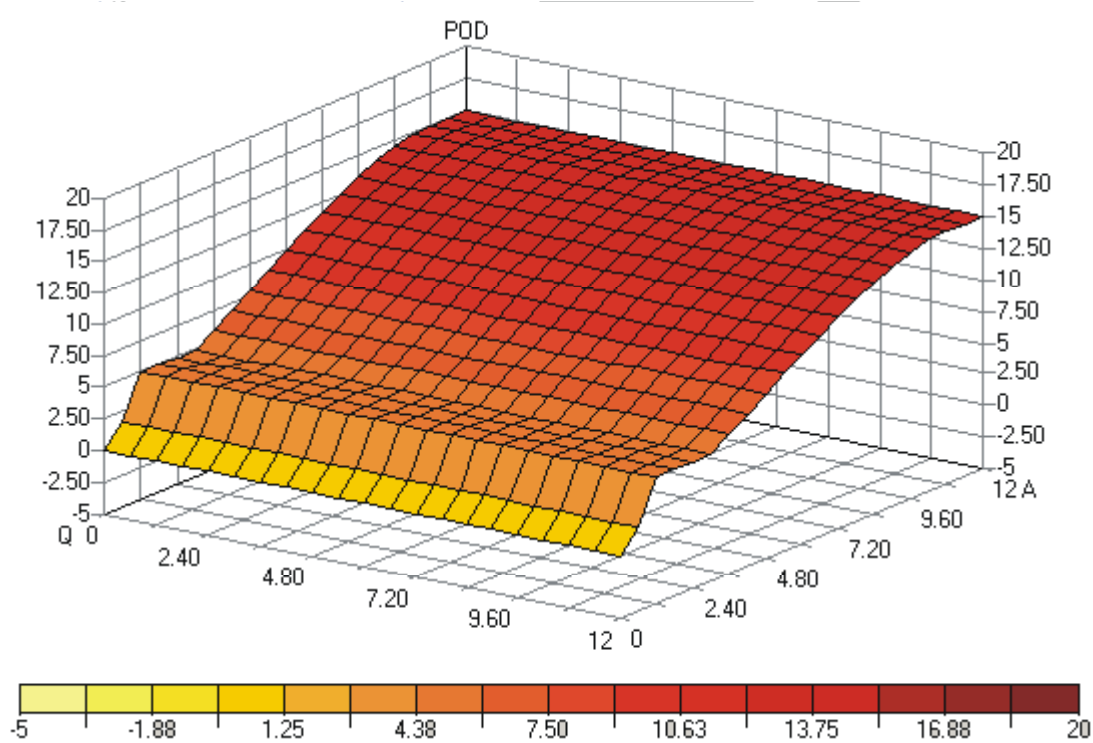
Pri optimizaciji gremo skozi celotno definicijsko območje vhodnih podatkov. To je tridimenzionalni prostor $D_A \times D_N \times D_Q$, kjer so D_A , D_N , D_Q po vrsti definicijska območja spremenljivk A , N , Q . Za vsako točko iz definicijskega območja preverimo, če sistem daje željeni rezultat ali če je ta rezultat smiseln. Če z rezultatom nismo zadovoljni, spremenimo kako pripadnostno funkcijo ali pravilo. Preverjanje delovanja sistema nam je zelo olajšal program *fuzzyTECH*, ki omogoča interaktivno delo in vizualizacijo dogajanja v sistemu mehke logike (slika 6.9, str. 69).

Zelo pregleden prikaz delovanja omogočajo tudi preslikave sistema (glej sliki 6.10 in 6.11, str. 70). Pri interaktivnem delu nam graf preslikave takoj prikaže odziv na spremembo v podatku ali spremembo v definiciji katerega od elementov sistema. Sistem je bil tako optimiziran po metodi poizkusov in napak ter z odstranjevanjem napak.

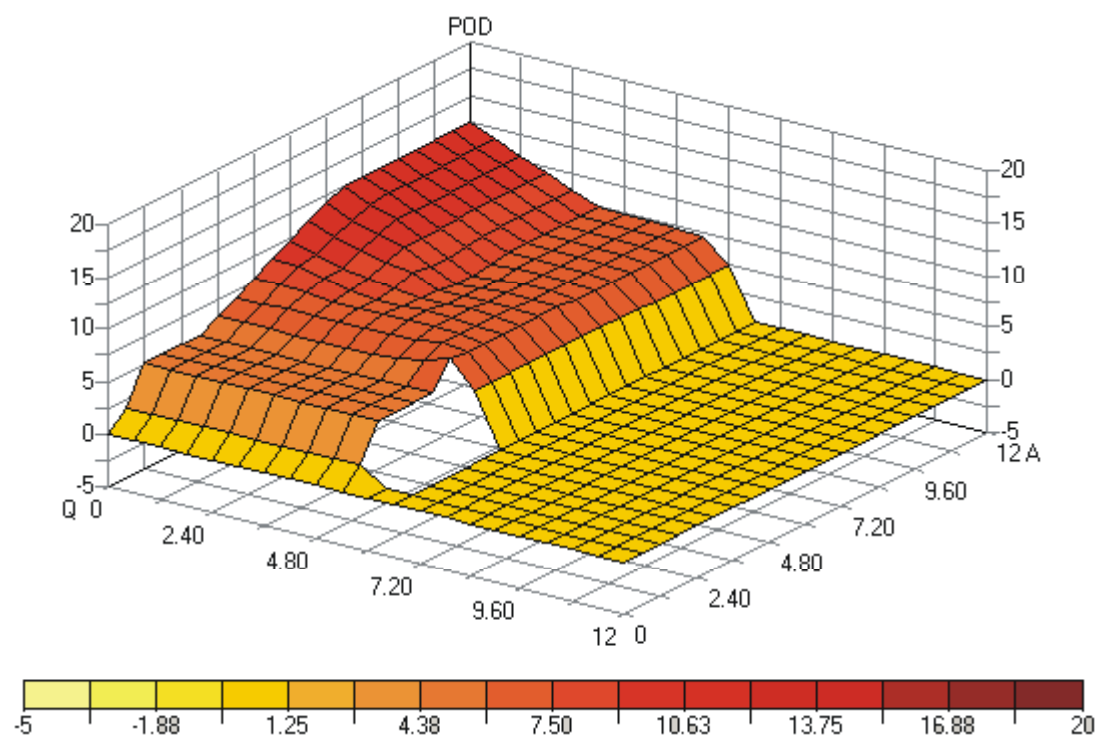
Slika 6.9: Slika interaktivnega optimiziranja sistema



Slika 6.10: Graf preslikave sistema pri prvem podaljšku



Slika 6.11: Graf preslikave sistema pri drugem podaljšku



7 Rezultati

Sistem je bil testiran tudi s pomočjo simulacije. Ker nimamo na razpolago pravega simulacijskega sistema prometne signalizacije v križišču, je bila narejena preprosta simulacija glede na podatke, zbrane z videoposnetkov, s pomočjo programa Microsoft Excel.

Tabela 7.1: Simulacija delovanja izpeljanega sistema mehke logike

#	Cesta 2	stanje (št.vozil)	dodeli (sekund)	Cesta 1	stanje (št.vozil)	dodeli (sekund)
cikel 1	začetek zelenega signala		5 minimalna zelena	rdeč signal		
	konec minimalne zelene	5 se približuje	8 prvega podaljška	rdeč signal		
	konec prvega podaljška	0 se približuje	0 drugega podaljška	rdeč signal	2 čaka v koloni	
	konec zelenega signala	0 se približuje		rdeč signal		
	rdeč signal			začetek zelenega signala		5 minimalna zelena
	rdeč signal			konec minimalne zelene	5 se približuje	8 prvega podaljška
	rdeč signal	6 čaka v koloni		konec prvega podaljška	5 se približuje	5 drugega podaljška
	rdeč signal			konec zelenega signala	2 se približuje	
cikel 2	začetek zelenega signala		5 minimalna zelena	rdeč signal		
	konec minimalne zelene	5 se približuje	8 prvega podaljška	rdeč signal		
	konec prvega podaljška	6 se približuje	8 drugega podaljška	rdeč signal	3 čaka v koloni	
	konec zelenega signala	1 se približuje		rdeč signal		
	rdeč signal			začetek zelenega signala		5 minimalna zelena
	rdeč signal			konec minimalne zelene	7 se približuje	12 prvega podaljška
	rdeč signal	5 čaka v koloni		konec prvega podaljška	2 se približuje	5 drugega podaljška
	rdeč signal	9 čaka v koloni		konec zelenega signala	0 se približuje	
cikel 3	začetek zelenega signala		5 minimalna zelena	rdeč signal		
	konec minimalne zelene	4 se približuje	7 prvega podaljška	rdeč signal		
	konec prvega podaljška	3 se približuje	5 drugega podaljška	rdeč signal	5 čaka v koloni	
	konec zelenega signala	3 se približuje		rdeč signal		
	rdeč signal			začetek zelenega signala		5 minimalna zelena
	rdeč signal			konec minimalne zelene	5 se približuje	8 prvega podaljška
	rdeč signal	3 čaka v koloni		konec prvega podaljška	3 se približuje	5 drugega podaljška
	rdeč signal			konec zelenega signala	0 se približuje	
cikel 4	začetek zelenega signala		5 minimalna zelena	rdeč signal		
	konec minimalne zelene	7 se približuje	12 prvega podaljška	rdeč signal		
	konec prvega podaljška	0 se približuje	0 drugega podaljška	rdeč signal	2 čaka v koloni	
	konec zelenega signala	0 se približuje		rdeč signal		
	rdeč signal			začetek zelenega signala		5 minimalna zelena
	rdeč signal			konec minimalne zelene	4 se približuje	7 prvega podaljška
	rdeč signal	5 čaka v koloni		konec prvega podaljška	2 se približuje	5 drugega podaljška
	rdeč signal			konec zelenega signala	0 se približuje	

Vir: rezultati simulacije glede na podatke z videoposnetkov.

Dogajanje poteka v navpični smeri. Na začetku zelene faze vedno nastopi minimalna dolžina zelenega signala, ki znaša 5 s. Nato glede na število vozil, ki se približujejo zelenemu signalu, izračunamo dolžino prvega podaljška. Ob koncu prvega podaljška pa glede na število vozil, ki se približujejo, in obenem glede na število vozil, ki stojijo v koloni pri rdečem signalu, izračunamo še dolžino drugega podaljška.

Povprečna dolžina zelenega signala je po simulaciji 17 sekund, torej gre za približno enak rezultat kot pri fiksnih intervalih. Kratek podaljšek znaša 5 sekund. V kolikor bi se izkazal za prekratkega, to zlahka spremenimo: le definicijsko točko vrednosti *kratek* popravimo na recimo (7,1) in dobimo 7 sekund za kratek podaljšek. Kot smo že omenili v prejšnjih poglavjih, je sisteme mehke logike enostavno optimizirati, saj so precej transparentni in nudijo vizualno predstavo logike sistema.

Zadovoljni smo z rezultatom, da je bil v petih od osmih primerov podaljšek ravno idealno dolg: vsa vozila bi uspela prečkati križišče in ob prekinitvi ne bi bilo nobenega vozila, ki bi se še približevalo. Do nezaželenih primerov, ko vozila čakajo na zeleni signal več kot 15 sekund, ne da bi v tem času kako vozilo iz druge smeri prečkalo križišče, ne more priti. Po pravilih sistema se tudi ne more zgoditi, da bi bil po prvem podaljšku dodeljen še en podaljšek, če čaka v koloni več kot 7 vozil. V primeru, da je to prenizka meja, jo enostavno povišamo tako, da spremenimo pripadnostno funkcijo mehke množice *preveliko* za spremenljivko Q .

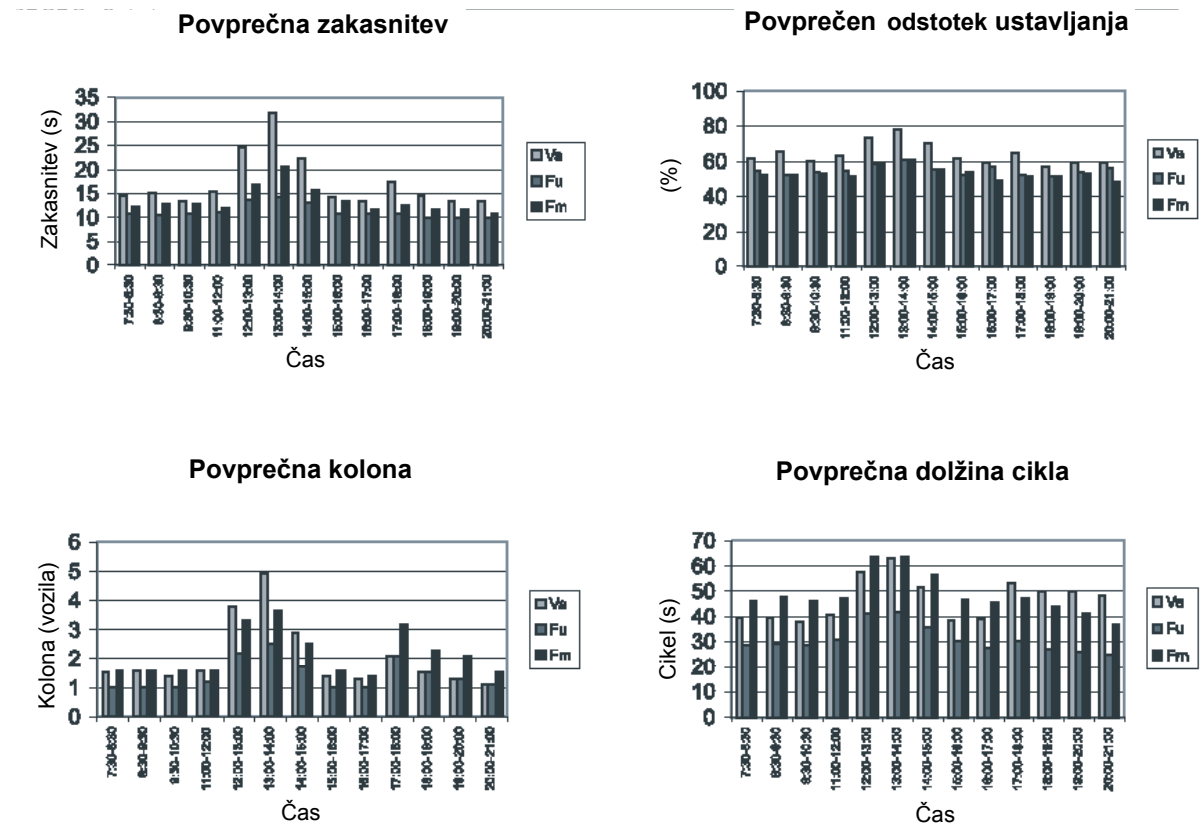
Ker so pri upravljanju signalizacije z uporabo mehke logike cikli krajši, je to prednost tudi za pešce, saj se jim pogosteje ponuja možnost prečkanja ceste. Z ozirom na lokacijo križišča je to tudi zelo pomembno dejstvo.

Omenimo še rezultate, ki so jih pri upravljanju prometne signalizacije z mehko logiko dobili pri projektu FUSICO. Zaradi dobrega simulacijskega sistema so lahko primerjali sistem upravljanja z mehko logiko z najsodobnejšim sistemom na zaznavo vozil (slika 7.1, strani 73).

Pri primerjavi strategij signalizacije sta bila geometrija križišča in pretok prometa enaka. Pri primerjavi mehkega upravljalnega sistema in sistema na zaznavo vozil so prišli do naslednjih ugotovitev: v uporabi je več algoritmov na zaznavo vozil glede na del dneva, mehki upravljalni algoritem pa je običajno eden in ta zadošča. Sistem mehke logike je konkurenčen tradicionalnemu sistemu na zaznavo vozil, v razmerah večjega prometnega pretoka pa je po raziskavah projekta FUSICO sistem mehke logike boljši. Maksimalni prihranki pri zakasnitvi so okoli 20%. Pomemben rezultat je, da so cikli pri sistemu mehke logike krajši. Razlog za to je v osnovi, da pri mehkem upravljalnem sistemu ni podaljškov zaradi enega samega približajočega se vozila, če je število čakajočih vozil veliko.

Rezultati meritev s terena kažejo, da mehki upravljalni algoritem v večini primerov deluje bolje. Povprečne zakasnitve so krajše, število vozil, ki se morajo ustaviti, je manjše, kar prispeva tudi k varnosti v prometu. Izkazalo se je, da je tudi poraba goriva manjša.

Slika 7.1: Primerjava rezultatov sistema na zaznavo vozil in sistema z mehko logiko



Vir: Rezultati meritev glede na simulacije algoritma sistema na zaznavo vozil (VA) in algoritmov mehkega upravljanja (FU s ciljem tekočega prometa in FM s ciljem tekočega prometa, hkrati pa tudi varnosti in prijaznosti okolju) po raziskavi projekta FUSICO (Niittymäki, 2001, str. 279).

Torej je mehki upravljalni algoritem boljši z vidika vseh treh glavnih ciljev: minimalnih zakasnitev, maksimalne varnosti in minimalnih negativnih posledic za okolje.

Iz navedenega lahko sklepamo, da je uporaba mehke logike pri upravljanju prometne signalizacije smiselna in ekonomsko upravičena.

8 Sklep

Mehka logika postaja vse bolj razširjena metoda pri upravljanju prometne signalizacije. Izkazala se je kot primerna, saj z uporabo lingvističnih spremenljivk in pravil omogoča enostaven prenos aproksimativne narave človeškega sklepanja. Mehka logika razume lingvistična navodila in omogoča generiranje upravljalne strategije na osnovi verbalne komunikacije.

V magistrskem delu je predstavljena uporaba mehke logike pri upravljanju prometne signalizacije v križišču. V ta namen so v delu zajeta poglavja iz osnov mehke logike, sistemov prometne signalizacije in metod preverjanja njihove učinkovitosti. V drugem delu magistrskega dela smo našteje vsebine povezali; predstavljen je bil splošen pristop upravljanja prometne signalizacije z uporabo mehke logike s poudarkom na predstavitvi sistematičnega pristopa mehkega upravljanja signalizacije glede na tip križišča in na tvorbi pravil. Logika mehkega upravljalnega sistema je zajeta ravno v bazi pravil, ki so lahko različna glede na prometne situacije. Osnovni princip gradnje mehkega sistema je uporaba ekspertnega znanja.

Namen raziskave je bil tudi predstaviti uporabo izpeljanega algoritma na konkretnem primeru, in sicer za izbrano križišče v Kopru. Sistem smo konstruirali in optimizirali z uporabo računalniškega programa za delo s sistemi mehke logike, nato pa smo ga testirali z enostavno simulacijo, izpeljano glede na podatke, pridobljene z videoposnetki.

Rezultati, ki jih daje izpeljani sistem, so kljub njegovi enostavni strukturi povsem zadovoljivi. Sistem upravljanja signalizacije z uporabo mehke logike deluje po principu podaljševanja faze zelenega signala, pri tem pa je pomembno poudariti, da sistem mehke logike upošteva tako število vozil, ki se približujejo zelenemu signalu, kot tudi število vozil, ki čakajo pri rdečem signalu. Zahtevnejši sistemi pa hkrati lahko upoštevajo tudi druge dejavnike, kot so pešci in kolesarji ter prednost vozil javnega prevoza.

Sistemi upravljanja signalizacije z mehko logiko omogočajo manjše zakasnitve v prometu, redkejše ustavljanje vozil v križiščih in tudi manjšo porabo goriva. Torej je mehki upravljalni algoritem boljši od ostalih algoritmov z vidika vseh treh glavnih ciljev upravljanja signalizacije: minimalnih zakasnitev, maksimalne varnosti in minimalnih negativnih posledic za okolje. Dobro načrtovanje signalizacije pripomore k povečanju učinkovitosti celotnega cestnega omrežja. Pomembno pa je tudi vzdrževanje. Sistemi mehke logike so zaradi svoje strukture bolj transparentni od klasičnih sistemov, to pa poenostavi tudi kasnejše optimiziranje.

Po teh ugotovitvah lahko sklepamo, da je uporaba mehke logike pri upravljanju prometne signalizacije smiselna in ekonomsko upravičena.

Slovarček tujih izrazov

Advanced Traveler Information System	napredni potniški informacijski sistem
And	in
Bounded Sum	omejena vsota
Center of Area	center ploščine
Center of Gravity	center gravitacije
Center of Maximum	center maksimumov
Degree of Support	utež
Delay	zakasnitev
Fuzzy Logic	mehka logika
If	če
Mean of Maximum	srednji maksimum
On-line simulation	simulacija v realnem času
Softmin	mehki minimum
Then	potem

Kazalo slik

Slika 2.1: Pripadnostne funkcije in njihovi parametri.....	9
Slika 2.2: Operatorji preseka.....	12
Slika 2.3: Operatorji unije.....	14
Slika 2.4: Struktura sistema mehke logike.....	15
Slika 2.5: Pripadnostne funkcije za primer spremenljivke »število vozil iz smeri A«	17
Slika 2.6: Operator minimum.....	19
Slika 2.7: Operator maksimum	19
Slika 2.8: Operator min-max.....	20
Slika 2.9: Operator gama.....	20
Slika 2.10: Izbira tipičnih vrednosti mehkih množic	22
Slika 2.11: Primer ostrenja z metodo CoM.....	23
Slika 2.12: Primer ostrenja z metodo CoA.....	23
Slika 2.13: Primer ostrenja z metodo MoM	24
Slika 3.1: Glavni cilji upravljanja prometne signalizacije in njihovi medsebojni vplivi	27
Slika 3.2: Princip podaljševanja pri upravljalnem sistemu signalizacije na zaznavo vozil	29
Slika 4.1: Upravljanje prometne signalizacije na osnovi on-line simulacije	37
Slika 4.2: Mehki upravljalnik signalizacije in povezave s simulacijskim sistemom	38
Slika 5.1: Konfiguracija križišča pri poizkusu Pappisa in Mamdanija	40
Slika 5.2: Struktura sistema za upravljanje prometne signalizacije.....	42
Slika 5.3: Primer pripadnostnih funkcij dvofaznega upravljanja signalizacije.....	48
Slika 5.4: Preslikave za sistem signaliziranega prehoda za pešce	52
Slika 5.5: Preslikave sistema sklepanja za podaljševanje ali prekinitve faze	53
Slika 5.6: Primerjava metod ostrenja	54
Slika 5.7: Primerjava principov podaljševanja pri metodi na zaznavo vozil in metodi mehke logike.....	57
Slika 6.1: Slika izbranega križišča v Kopru	58
Slika 6.2: Lokacija izbranega križišča	59
Slika 6.3: Slika situacije, ko kolona vozil po nepotrebnem čaka.....	60
Slika 6.4: Struktura sistema mehke logike za upravljanje prometne signalizacije	62
Slika 6.5: Pripadnostne funkcije vhodne spremenljivke A	63
Slika 6.6: Pripadnostne funkcije vhodne spremenljivke N	64
Slika 6.7: Pripadnostne funkcije vhodne spremenljivke Q	65
Slika 6.8: Pripadnostne funkcije izhodne spremenljivke POD	65
Slika 6.9: Slika interaktivnega optimiziranja sistema	69
Slika 6.10: Graf preslikave sistema pri prvem podaljškju	70
Slika 6.11: Graf preslikave sistema pri drugem podaljškju	70
Slika 7.1: Primerjava rezultatov sistema na zaznavo vozil in sistema z mehko logiko	73

Kazalo tabel

Tabela 2.1: Primer lingvističnih spremenljivk in njihovih lingvističnih vrednosti.....	16
Tabela 2.2: Primerjava metod ostrenja.....	25
Tabela 6.1: Vhodne spremenljivke.....	63
Tabela 6.2: Izhodna spremenljivka	63
Tabela 6.3: Definijske točke pripadnostnih funkcij spremenljivke A	64
Tabela 6.4: Definijske točke pripadnostnih funkcij spremenljivke N	64
Tabela 6.5: Definijske točke pripadnostnih funkcij spremenljivke Q	65
Tabela 6.6: Definijske točke pripadnostnih funkcij spremenljivke POD	66
Tabela 6.7: Statistika bloka pravil za izračun podaljška	66
Tabela 6.8: Pravila za izračun podaljška	67
Tabela 6.9: Statistika celotnega sistema.....	67
Tabela 7.1: Simulacija delovanja izpeljanega sistema mehke logike	71

Literatura in viri

Literatura

1. Adler Jeffrey L., Blue Victor J.: Toward the Design of Intelligent Traveler Information Systems. Transportation Research Part C, Elsevier Science, 1998, str. 157-172.
2. Bingham Ella: Neurofuzzy Traffic Signal Control. Master's Thesis. Helsinki: Helsinki University of Technology, 1998. 107 str.
3. Bingham Ella: Reinforcement Learning in Neurofuzzy Traffic Signal Control. European Journal of Operational Research, Elsevier Science, Vol. 131, 2001, str. 232–241.
4. Chiu S.: Adaptive Traffic Signal Control Using Fuzzy Logic. Proceedings of the Intelligent Vehicles Symposium, Detroit, 1992, str. 98-107.
5. Chou Chin-Hsun, Teng Jen-Chao: A Fuzzy Logic Controller for Traffic Junction Signals. Information Sciences, Elsevier, 143, 2002, str. 73-97.
6. David Matej, Malej Alenka: Bio-Pollution Risk Assessment with Fuzzy Logic. Zbornik 6. mednarodnega znanstveno-strokovnega posvetovanja o prometni znanosti ICTS 2002. Portorož: Fakulteta za pomorstvo in promet, 2002, str. 116-123.
7. Dolenc Jure: Že spet rdeča!? ali kako deluje promet v mestih. Kvarkadabra, časopis za tolmačenje znanosti, april 2002.
[URL: <http://www.kvarkadabra.net/article.php?story=20030210162920972>]
8. Favilla J., Machion A., Gomide F.: Fuzzy Traffic Control: Adaptive Strategies. Proceedings of the 2nd IEEE International Conference on Fuzzy Systems, Vol. 1, San Francisco, USA, 1993, str. 505-511.
9. Golob Marjan, Muškinja Nenad: Osnove snovanja mehkih (fuzzy) regulacij. Maribor: Tehniška fakulteta, Elektrotehnika, računalništvo in informatika, 1994. 82 str.
10. Harrill Rob: When it Comes to Heavy Seattle Traffic, 'Fuzzy Logic' Smooths the Flow. University of Washington, 24.11.1999.
[URL: <http://www.washington.edu/newsroom/1999archive/11-99archive/k112499.html>]
11. Hellendoorn Hans: After the Fuzzy Wave Reached Europe. European Journal of Operational Research, Elsevier Science, 99, 1997, str. 58-71.

12. Khiang Tan Kok, Khalid Marzuki, Yusof Rubiyah: Intelligent Traffic Lights Control by Fuzzy Logic. Malaysian Journal of Computer Science, November 1995. Centre for Artificial Intelligence and Robotics, University of Teknologi Malaysia.
[URL: http://www.cairo.utm.my/publications/kktan_traffic.pdf]
13. Kim S.: Application of Petri Networks and Fuzzy Logic to Advanced Traffic Management Systems. Ph. D. Thesis. Polytechnic University, New York, USA, 1994. 139 str.
14. Klir George J., St. Clair Ute, Yuan Bo: Fuzzy Set Theory: Foundations and Applications. Upper Saddle River (New Jersey): Prentice Hall PTR, 1997. 245 str.
15. Klir George J., Yuan Bo: Fuzzy Sets and Fuzzy Logic: Theory and Applications. Upper Saddle River (New Jersey): Prentice Hall PTR, 1995. 574 str.
16. Kosko Bart: Fuzzy Engineering. Upper Saddle River: Prentice Hall, 1997. 549 str.
17. Kosonen Iisakki: HUTSIM – Urban Traffic Simulation and Control Model: Principles and Applications. Helsinki University of Technology, Transportation Engineering, Publication 100. Espoo, Finland, 1999. 249 str.
18. Kosonen Iisakki, Bargiela Andrzej: A Distributed Traffic Monitoring and Information System. Journal of Geographic Information and Decision Analysis, vol. 3, no. 1, 1999, str. 31-40.
19. Kosonen Iisakki, Kokkinen Matti: A New Simulation System for Traffic Signal Control Evaluation. ITE 62nd Annual Meeting, Compendium of Technical Papers. Washington D.C., 1992, str. 250-254.
20. Murat Yetis Sazi, Gedizlioglu Ergun: A New Approach for Fuzzy Traffic Signal Control. Proceedings of the 13th Mini-Euro Conference: Handling Uncertainty in the Analysis of Traffic and Transportation Systems. Roma: Istituto di Analisi dei Sistemi ed Informatica "Antonio Ruberti", 2002, str. 174-179.
21. Nakatsuyama M., Nakahashi H., Nishizuka N.: Fuzzy Logic Phase Controller for Traffic Junctions in the One-way Arterial Road. Proceedings of the IFCA 9th Triennial World Congress. Pergamon Press. Oxford, 1984, str. 2865-2870.
22. Nam Do H., Drew Donald R.: Automatic Measurement of Traffic Variables for Intelligent Transportation Systems Applications. Transportation Research 33B, 1999, str. 437-457.
23. Niittymäki Jarkko: Using Fuzzy Logic to Control Traffic Signals at Multi-Phase Intersections. Reusch B. (Ed.). Computational Intelligence – Theory and Applications.

- International Conference, 6th Fuzzy Days, Proceedings. Berlin-Heidelberg: Springer, 1999, str. 354-362.
24. Niittymäki Jarkko: Installation and Experiences of Field Testing a Fuzzy Signal Controller. *European Journal of Operational Research*, Elsevier Science, Vol. 131/2, 2001, str. 273–281.
 25. Niittymäki Jarkko: General Fuzzy Rule Base for Isolated Traffic Signal Control – Rule Formulation. *Journal of Transportation Planning and Technology*, Gordon & Breach Science Publishers, Vol. 24, 2001a, 3, str. 227-247.
 26. Niittymäki Jarkko: Fuzzy Traffic Signal Control – Principles and Applications. Doctoral Thesis. Helsinki: Helsinki University of Technology, 2002. 71 str.
 27. Niittymäki Jarkko, Kikuchi S.: Application of Fuzzy Logic to the Control of a Pedestrian Crossing Signal. *Transportation Research*, Washington D.C., 1998, No. 1651, str. 30-38.
 28. Niittymäki Jarkko, Könönen V.: Traffic Signal Controller Based on Fuzzy Logic. *SMC2000 Conference Proceedings, 2000 IEEE International Conference on Systems, Man & Cybernetics*. Nashville, Tennessee: IEEE, 2000, str. 3578-3581.
 29. Niittymäki Jarkko, Mäenpää M.: The Role of Fuzzy Logic Public Transport Priority in Traffic Signal Control. *Traffic Engineering and Control, International Journal of Traffic Management and Transportation Planning*. TEC, Hemming-Group Ltd., Vol. 42., 2001, 1, str. 22-26.
 30. Niittymäki Jarkko, Pursula Matti: Signal Control Using Fuzzy Logic. *Fuzzy Sets and Systems, International Journal of Soft Computing*, Elsevier Science, Vol. 116, 2000, 1, str. 11-22.
 31. Palacharla Prasad V., Nelson Peter C.: Application of Fuzzy Logic and Neural Networks for Dynamic Travel Time Estimation. *International Transactions in Operational Research* 6, 1999, str. 145-160.
 32. Pursula Matti: Simulations of Traffic Systems – An Overview. *Journal of Geographic Information and Decision Analysis*, vol. 3, 1999, no. 1, str. 1-8.
 33. Ross Timothy J.: *Fuzzy Logic with Engineering Applications*. New York [etc.]: McGraw-Hill, Inc., 1995. 600 str.
 34. Sayers T., Anderson J., Bell M.: Traffic Control System Optimisation: A Multiobjective Approach. *Mathematics in Transport Planning and Control. Proceedings of the 3rd IMA International Conference on Mathematics in Transport Planning and Control*. Pergamon, Elsevier Science, UK, 1998, str. 37-46.

35. Teodorovic Dusan: Fuzzy Logic Systems for Transportation Engineering: The State of the Art. Transportation Research Part A, Elsevier Science, Vol. 33, 1999, str. 337-364.
36. Trabia Mohamed B., Kaseko Mohamed S., Ande Murali: A Two-stage Fuzzy Logic Controller for Traffic Signals. Transportation Research Part C, Elsevier Science, Vol. 7, 1999, str. 353-367.
37. Usenik Janez, Malej Alenka: Predvidevanje vremenskih razmer v prometu z uporabo mehke logike. Zbornik 10. mednarodnega simpozija o elektroniki v prometu. Ljubljana: Elektrotehniška zveza Slovenije, 2002, P20.
38. Virant Jernej: Uporaba mehke logike v sodobnih sistemih. Ljubljana [i. e.] Radovljica: Didakta, 1992. 318 str.
39. Virant Jernej: Čas v mehkih sistemih. Radovljica: Didakta, 1998. 415 str.
40. von Altrock Constantin: Fuzzy Logic and Neurofuzzy Applications Explained. Upper Saddle River (New Jersey): Prentice Hall PTR, 1995. 350 str.
41. von Altrock Constantin: Fuzzy Logic and Neurofuzzy Applications in Business And Finance. Upper Saddle River (New Jersey): Prentice Hall PTR, 1997. 375 str.
42. Wahle J. et. al.: A Dynamic Route Guidance System Based on Real Traffic Data. European Journal of Operational Research 131, 2001, str. 302-308.
43. Zadeh Lotfi A.: Fuzzy Sets. Information and Control, Vol. 8, 1965, 3, str. 338-353.
44. Zimmermann Hans – Jürgen: Fuzzy Set Theory – and its Applications. Boston: Kluwer Academic Publishers, 1996. 435 str.

Viri

1. Hutsim – spletna stran.
[URL: <http://www.hut.fi/Units/Transportation/HUTSIM/index.html>],
Helsinki University of Technology, 10. 12. 2003.
2. *fuzzy*TECH 5.3 User's Manual. Inform Software Corporation, 1999. 345 str.
3. *fuzzy*TECH 5.52 Professional Edition. Inform Software Corporation
4. Smartest – Simulation Modeling Applied to Road Transport European Scheme Tests:
Review of Micro-Simulation Models.
[URL: <http://www.its.leeds.ac.uk/projects/smartest/append3d.html>],
University of Leeds, UK, Institute for Transport Studies, 1.2.2004.
5. Elektronski telefonski imenik Slovenije 2002. Telekom Slovenije, 2002.
6. Fuzzy Logic Toolbox User's Guide. The Mathworks Inc., 2000.

Priloga

- Vsebina *fuzzy*TECH datoteke FtUpravljanjeSignalizacije.ftl

Vsebina *fuzzy*TECH datoteke FtUpravljanjeSignaizacije.ftl

```
PROJECT {
  NAME      = FT UPRAVLJANJE SIGNALIZACIJE.FTL;
  TITLE     =FT UPRAVLJANJE SIGNALIZACIJE;
  AUTHOR    = Alenka Malej;
  DATEFORMAT = YYYY.DD.MM;
  LASTCHANGE = 2004.06.02;
  CREATED   = 2004.03.02;
  SHELL     = ON_LINE;
  SHELLOPTIIONS {
    ONLINE_REFRESHTIME = 55;
    ONLINE_TIMEOUTCOUNT = 1100;
    ONLINE_CODE = OFF;
    ONLINE_TRACE_BUFFER = (OFF, PAR(0));
    COMMENTS = ON;
    FTL_BUFFER = (OFF, PAR(1));
    PASSWORD = OFF;
    PUBLIC_IO = ON;
    FAST_CMBF = OFF;
    FAST_COA = ON;
    BTYPE = DOUBLE;
    C_TYPE = ANSI;
  } /* SHELLOPTIIONS */
  MODEL {
    VARIABLE_SECTION {
      LVAR {
        NAME = A;
        BASEVAR = vozila;
        LVRANGE = MIN(0.0), MAX(12.0),
                MINDEF(0), MAXDEF(65535),
                DEFAULT_OUTPUT(0.0);
        RESOLUTION = XGRID(1.0), YGRID(1.0),
                    SHOWGRID (ON), SNAPTOGRID(ON);
        EXPLANATION {
          PAR {A = število približujočih se vozil}
        }
        COMMENTTEXT {
          PAR {Število približujočih se vozil predstavlja število vseh vozil, ki se
približujejo križišču v smeri zelenega signala na vseh voziščih skupaj.}
        }
        COLOR = RED (255), GREEN (0), BLUE (0);
        INPUT = CMBF;
        POS = -282, -79;
        TERM {
          TERMNAME = nic;
          POINTS = (0.0, 1.0),
                  (1.0, 0.0),
                  (12.0, 0.0);
          SHAPE = LINEAR;
          COLOR = RED (255), GREEN (0), BLUE (0);
        }
        TERM {
          TERMNAME = malo;
          POINTS = (0.0, 0.0),
                  (3.0, 1.0),
                  (6.0, 0.0),
                  (12.0, 0.0);
          SHAPE = LINEAR;
          COLOR = RED (0), GREEN (128), BLUE (0);
        }
        TERM {
          TERMNAME = srednje;
          POINTS = (0.0, 0.0),
                  (3.0, 0.0),
                  (6.0, 1.0),
                  (10.0, 0.0),
                  (12.0, 0.0);
          SHAPE = LINEAR;
          COLOR = RED (0), GREEN (0), BLUE (255);
        }
      }
    }
  }
}
```

```

TERM {
  TERMNAME = veckotmalo;
  POINTS   = (0.0, 0.0),
             (3.0, 0.0),
             (6.0, 1.0),
             (12.0, 1.0);
  SHAPE    = LINEAR;
  COLOR    = RED (0), GREEN (128), BLUE (128);
}
TERM {
  TERMNAME = veliko;
  POINTS   = (0.0, 0.0),
             (6.0, 0.0),
             (9.0, 1.0),
             (12.0, 1.0);
  SHAPE    = LINEAR;
  COLOR    = RED (128), GREEN (0), BLUE (0);
}
} /* LVAR */
LVAR {
  NAME      = N;
  BASEVAR   = podaljsek;
  LVRANGE   = MIN(1.0), MAX(2.0),
             MINDEF(0), MAXDEF(65535),
             DEFAULT_OUTPUT(1.0);
  RESOLUTION = XGRID(1.0), YGRID(1.0),
              SHOWGRID (ON), SNAPTOGRID(ON);
  EXPLANATION {
    PAR {N = zaporedna številka podaljska}
  }
  COMMENTTEXT {
    PAR {S tem parametrom sledimo, pri katerem podaljsku smo.}
  }
  COLOR     = RED (0), GREEN (128), BLUE (128);
  INPUT     = CMBF;
  POS      = -281, -34;
  TERM {
    TERMNAME = p1;
    POINTS   = (1.0, 1.0),
               (2.0, 0.0);
    SHAPE    = LINEAR;
    COLOR    = RED (255), GREEN (0), BLUE (0);
  }
  TERM {
    TERMNAME = p2;
    POINTS   = (1.0, 0.0),
               (2.0, 1.0);
    SHAPE    = LINEAR;
    COLOR    = RED (0), GREEN (128), BLUE (0);
  }
} /* LVAR */
LVAR {
  NAME      = Q;
  BASEVAR   = vozila;
  LVRANGE   = MIN(0.0), MAX(12.0),
             MINDEF(0), MAXDEF(65535),
             DEFAULT_OUTPUT(0.0);
  RESOLUTION = XGRID(0.1), YGRID(1.0),
              SHOWGRID (ON), SNAPTOGRID(ON);
  EXPLANATION {
    PAR {Q = število vozil v koloni}
  }
  COMMENTTEXT {
    PAR {Število vozil v koloni predstavlja število vseh vozil, ki čakajo na zelen
signal na vseh voziščih skupaj.}
  }
  COLOR     = RED (0), GREEN (128), BLUE (0);
  INPUT     = CMBF;
  POS      = -281, 12;
  TERM {
    TERMNAME = malo;
    POINTS   = (0.0, 0.0),
               (3.0, 1.0),
               (6.1, 0.0),
               (12.0, 0.0);

```

```

    SHAPE      = LINEAR;
    COLOR      = RED (255), GREEN (0), BLUE (0);
}
TERM {
  TERMNAME = manjkotsrednje;
  POINTS   = (0.0, 1.0),
             (3.0, 1.0),
             (6.0, 0.0),
             (12.0, 0.0);
  SHAPE    = LINEAR;
  COLOR    = RED (0), GREEN (128), BLUE (128);
}
TERM {
  TERMNAME = srednje;
  POINTS   = (0.0, 0.0),
             (3.0, 0.0),
             (6.0, 1.0),
             (10.0, 0.0),
             (12.0, 0.0);
  SHAPE    = LINEAR;
  COLOR    = RED (0), GREEN (128), BLUE (0);
}
TERM {
  TERMNAME = manjkotmalo;
  POINTS   = (0.0, 1.0),
             (3.0, 0.0),
             (12.0, 0.0);
  SHAPE    = LINEAR;
  COLOR    = RED (0), GREEN (0), BLUE (128);
}
TERM {
  TERMNAME = preveliko;
  POINTS   = (0.0, 0.0),
             (6.0, 0.0),
             (10.0, 1.0),
             (12.0, 1.0);
  SHAPE    = LINEAR;
  COLOR    = RED (128), GREEN (0), BLUE (0);
}
} /* LVAR */
LVAR {
  NAME      = POD;
  BASEVAR   = s;
  LVRANGE   = MIN(-5.0), MAX(20.0),
             MINDEF(0), MAXDEF(65535),
             DEFAULT_OUTPUT(0.0);
  RESOLUTION = XGRID(1.0), YGRID(1.0),
              SHOWGRID (ON), SNAPTGRID(ON);
  EXPLANATION {
    PAR {POD = podaljšek}
  }
  COMMENTTEXT {
    PAR {Podaljšek pove, za koliko sekund naj podaljšamo zeleno fazo.}
  }
  COLOR     = RED (0), GREEN (0), BLUE (255);
  OUTPUT    = COM;
  POS       = 117, -41;
  TERM {
    TERMNAME = nic;
    POINTS   = (-5.0, 0.0),
             (0.0, 1.0),
             (5.0, 0.0),
             (20.0, 0.0);
    SHAPE    = LINEAR;
    COLOR    = RED (255), GREEN (0), BLUE (0);
  }
  TERM {
    TERMNAME = kratek;
    POINTS   = (-5.0, 0.0),
             (0.0, 0.0),
             (5.0, 1.0),
             (10.0, 0.0),
             (20.0, 0.0);
    SHAPE    = LINEAR;
    COLOR    = RED (0), GREEN (128), BLUE (0);
  }
}

```

```

}
TERM {
  TERMNAME = srednji;
  POINTS   = (-5.0, 0.0),
            (5.0, 0.0),
            (10.0, 1.0),
            (15.0, 0.0),
            (20.0, 0.0);
  SHAPE    = LINEAR;
  COLOR    = RED (0), GREEN (0), BLUE (255);
}
TERM {
  TERMNAME = dolg;
  POINTS   = (-5.0, 0.0),
            (10.0, 0.0),
            (15.0, 1.0),
            (20.0, 0.0);
  SHAPE    = LINEAR;
  COLOR    = RED (128), GREEN (0), BLUE (0);
}
} /* LVAR */
} /* VARIABLE_SECTION */

OBJECT_SECTION {
  RULEBLOCK {
    NAME      = IzracunPodaljska;
    INPUT     = A, N, Q;
    OUTPUT    = POD;
    AGGREGATION = (MIN_MAX, PAR (0.0));
    RESULT_AGGR = MAX;
    POS       = -103, -65;
    RULES {
      IF A = malo
        AND N = p1
      THEN POD = kratek WITH 1.000;
      IF A = srednje
        AND N = p1
      THEN POD = srednji WITH 1.000;
      IF A = nic
        AND N = p1
      THEN POD = nic WITH 1.000;
      IF A = veliko
        AND N = p1
      THEN POD = dolg WITH 1.000;
      IF A = nic
        AND N = p2
      THEN POD = nic WITH 1.000;
      IF N = p2
        AND Q = preveliko
      THEN POD = nic WITH 1.000;
      IF A = srednje
        AND N = p2
        AND Q = manjkotsrednje
      THEN POD = srednji WITH 1.000;
      IF A = veliko
        AND N = p2
        AND Q = manjkotsrednje
      THEN POD = srednji WITH 1.000;
      IF A = veckotmalo
        AND N = p2
        AND Q = malo
      THEN POD = kratek WITH 1.000;
      IF A = malo
        AND N = p2
        AND Q = manjkotsrednje
      THEN POD = kratek WITH 1.000;
      IF A = veliko
        AND N = p2
        AND Q = manjkotmalo
      THEN POD = dolg WITH 1.000;
    } /* RULES */
  } /* RULEBLOCK */
  REMARK {
    TEXT = Upravljalnik prometne signalizacije;
    POS = -210, -176;
  }
}

```

```

    FONTSPEC = -24, 700, 0, 0, 0, 18, 0;
    FONTNAME =Times New Roman;
    COLOR = RED (0), GREEN (0), BLUE (255);
}
REMARK {
    TEXT = Vhodne spremenljivke;
    POS = -281, -126;
    FONTSPEC = -13, 700, 0, 0, 0, 18, 0;
    FONTNAME =Times New Roman;
    COLOR = RED (255), GREEN (0), BLUE (255);
}
REMARK {
    TEXT = Izhodna spremenljivka;
    POS = 100, -123;
    FONTSPEC = -13, 700, 0, 0, 0, 18, 0;
    FONTNAME =Times New Roman;
    COLOR = RED (255), GREEN (0), BLUE (255);
}
REMARK {
    TEXT = Blok pravil;
    POS = -67, -125;
    FONTSPEC = -13, 700, 0, 0, 0, 18, 0;
    FONTNAME =Times New Roman;
    COLOR = RED (255), GREEN (0), BLUE (255);
}
} /* OBJECT_SECTION */
} /* MODEL */
} /* PROJECT */
/* fuzzyTECH 5.52 Professional Edition */
/* FT SL 00002 03 HS */
ONLINE {
    TIMESTAMP = 20040206141201UT;
} /* ONLINE */

NEUROFUZZY {
    LEARNRULE      =RandomMethod;
    STEPWIDTHDOS   = 0.101563;
    STEPWIDTHTERM  = 1.000000;
    MAXDEVIATION   = (50.000000, 1.000000, 0.750000);
    AVGDEVIATION   = 0.100000;
    MAXSTEPS       = 100;
    NEURONS        = 1;
    DATASEQUENCE  = RANDOM;
    UPDATEDBGWIN   = OFF;
} /* NEUROFUZZY */

```