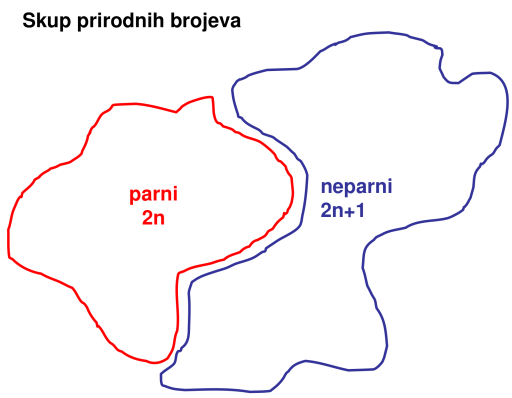
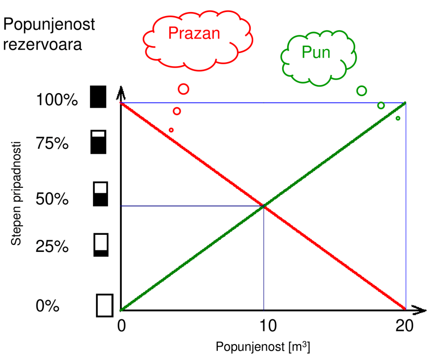
Fuzzy logika

- *Fuzzy (rasplinuta) logika* je način modelovanja logičkog razmišljanja gdje istinitost iskaza nije jasno binarna (0 ili 1, TRUE ili FALSE), već **stepen istinitosti** koji se kreće od 0 (apsolutno FALSE) do 1 (apsolutno TRUE).
- Odluke uz neodređenosti, nepreciznosti - podrške u odlučivanju.
- U kontekstu upravljanja, *fuzzy* sistemima (regulaciji) nije neophodan matematički model sistema te se kao takav može upotrebljavati za kompleksne sisteme čija mehanika nije u potpunosti poznata.

Fuzzy skupovi

- Osnovni element za predstavljanje i obradu nepreciznosti u *fuzzy* tehnologijama je ***fuzzy skup***.
- ***Fuzzy skup*** predstavlja skup elemenata sa sličnim svojstvima, za razliku od klasičnih (diskretnih, *crisp set*) skupova gdje jednom skupu moraju pripadati elementi sa istim svojstvima.
- U diskretnom skupu, svaki element tom skupu pripada **stepenom pripadnosti** koji iznosi 1. U *fuzzy* logici, element ne mora nužno imati stepen pripadnosti 1 kako bi pripadao nekom datom skupu.





- Slike iznad pokazuju razliku između klasičnih skupova (prirodan broj je ili paran ili neparan, ništa između) i *fuzzy* skupova (popunjenost rezervoara je u nekoj mjeri puna, a u nekoj mjeri prazna).

Funkcija pripadnosti

- Neka je dat neprazan skup X. *Fuzzy* skup A u X opisuje se funkcijom pripadnosti:

$$\mu_A(x) : X \rightarrow [0,1]$$

  gdje je $\mu_A(x)$ stepen pripadnosti elementa x u *fuzzy* skupu A za svako $x \in X$. Skup X takođe se naziva nadskup ili univerzalan skup. Gornji primjer - X je skup popunjenosti rezervoara dok je $A = \{pun\}$ ili $A = \{prazan\}$
- *Fuzzy* skup A može se predstaviti skupom parova:

$$A = \{(x, \mu_A(x)), x \in X\}$$

  Najčešće je X konačan skup $X = \{x_1, x_2, \ldots, x_n\}$, tako da se skup A može predstaviti na sledeći način (diskretna reprezentacija *fuzzy* skupa):

$$A = \frac{\mu_1}{x_1} + \frac{\mu_2}{x_2} + \ldots + \frac{\mu_n}{x_n} = \sum_{i=1}^{n} \frac{\mu_i}{x_i}$$

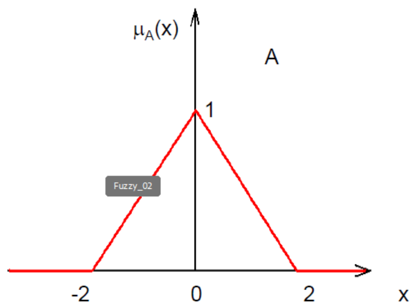
  Ako je X beskonačan i neprekidan skup sa elementima $x \in X$, tada se *fuzzy* skup može predstaviti u obliku (kontinualna reprezentacija *fuzzy* skupa):

$$A = \int_{x \in X} \frac{\mu_A(x)}{x} dx$$

- Neki od oblika funkcije pripadnosti:
  1. Trougaona funkcija pripadnosti (kontinualan i diskretan slučaj) - jedna maksimalna vrijednost:

1.a. Kontinualan slučaj

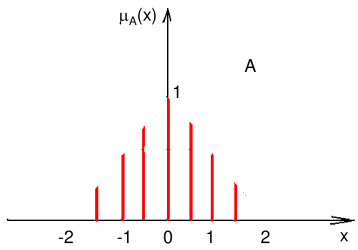
$$A = \int_{-2}^{0} \frac{\left(\frac{2+x}{2}\right)}{x} + \int_{0}^{2} \frac{\left(\frac{2-x}{2}\right)}{x}$$



1.b. Diskretan slučaj

X={-2,-1.5,-1,-0.5,0,0.5,1,1.5,2}

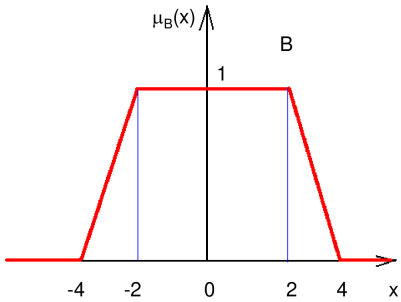
$$A = \frac{0.25}{-1.5} + \frac{0.5}{-1} + \frac{0.75}{-0.5} + \frac{1}{0} + \frac{0.75}{0.5} + \frac{0.5}{1} + \frac{0.25}{1.5}$$



2. Trapezoidna funkcija pripadnosti - više maksimalnih vrijednosti:

2. Trapezoidna funkcija pripadnosti

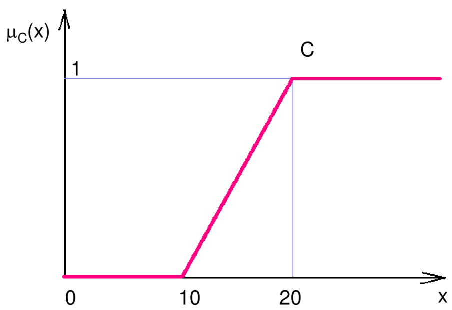
$$B = \int_{-4}^{-2} \frac{\left(\frac{4+x}{2}\right)}{x} + \int_{-2}^{2} \frac{1}{x} + \int_{2}^{4} \frac{\left(\frac{4-x}{2}\right)}{x}$$



3. Dio po dio pravolinijska funkcija prpadnosti - krajnji slučajevi:

3. Deo po deo pravolinijska funkcija pripadnosti

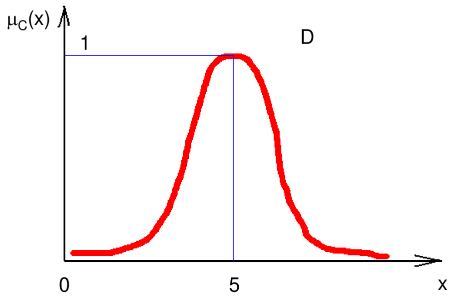
$$C = \int_{10}^{20} \frac{0.1x - 1}{x} + \int_{20}^{\infty} \frac{1}{x}$$



4. Zvonasta funkcija pripadnosti:

4. Zvonasta funkcija pripadnosti – radial basis

$$D = \int_X \frac{e^{-0.5(x-5)^2}}{x}$$



- Postoje i ostali oblici funkcija pripadnosti (Gausova, ...).

Osobine *fuzzy* skupova

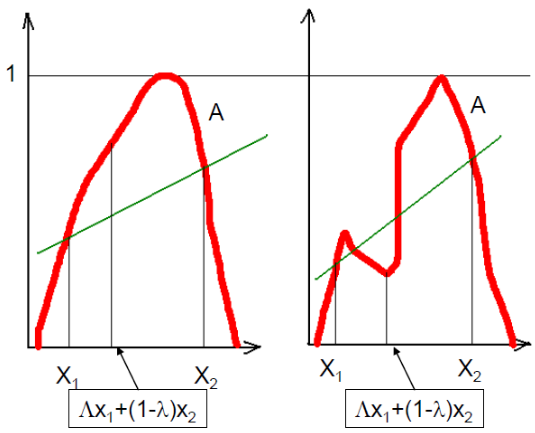
- **Normalnost *fuzzy* skupova**
    - *Fuzzy* skup je ***normalan*** ako i samo ako važi:

    $$\max_{x \in X} \mu_A(x) = 1$$

    Fuzzy skup koji nije normalan naziva se ***sub-normalan*** *fuzzy* skup. On se može transformisati u normalan tako što sve vrijednosti stepena pripadnosti podijele najvećim stepenom pripadnosti u datom skupu. Ovakva operacija se naziva ***normalizacija***.

- **Konveksnost (ispupčenost) *fuzzy* skupa**
    - *Fuzzy* skup je konveksan ako i samo ako važi da $\forall x_1, x_2 \in X, \lambda \in [0,1]$,

    $$\mu_A(\lambda x_1 + (1-\lambda)x_2) \geq \min(\mu_A(x_1), \mu_A(x_2))$$



    - Dok funkcija sa lijeve strane zadovoljava uslov konveksnosti, funkcija sa lijeve strane ne zadovoljava.

- **Kardinalnost (broj elemenata) *fuzzy* skupa**
  - Ako je X diskretan i konačan skup, onda se kardinalnost *fuzzy* skupa izražava zbirom stepena pripadnosti pojedinih elemenata *fuzzy* skupa:

$$|A| = \sum_{x \in X} \mu_A(x)$$

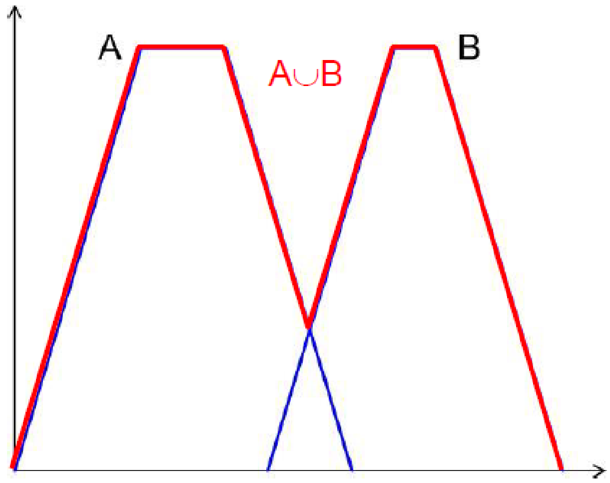
    Ova definicija broja elemenata *fuzzy* skupa odgovara definiciji broja elemenata diskretnog skupa.
  - Relativna kardinalnost se dobija kao:

$$||A|| = \frac{|A|}{|X|}$$

Osnovne operacije sa *fuzzy* skupovima

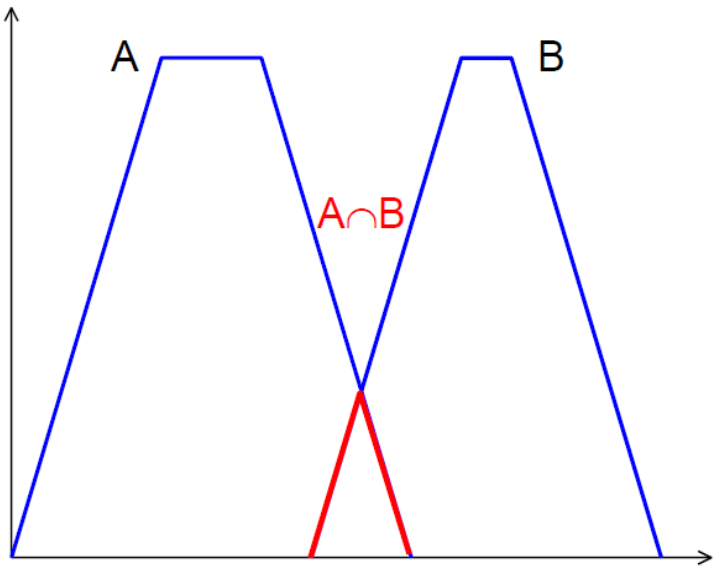
- **Zbir dva *fuzzy* skupa - unija *fuzzy* skupova**
  - Zbir (unija) *fuzzy* skupova A i B je *fuzzy* skup $A \cup B$ predstavljen pomoću funkcije pripadnosti:

$$\mu_{A \cup B}(X) = \mu_A(x) \vee \mu_B(x) = \max\{\mu_A(x), \mu_B(x)\}$$



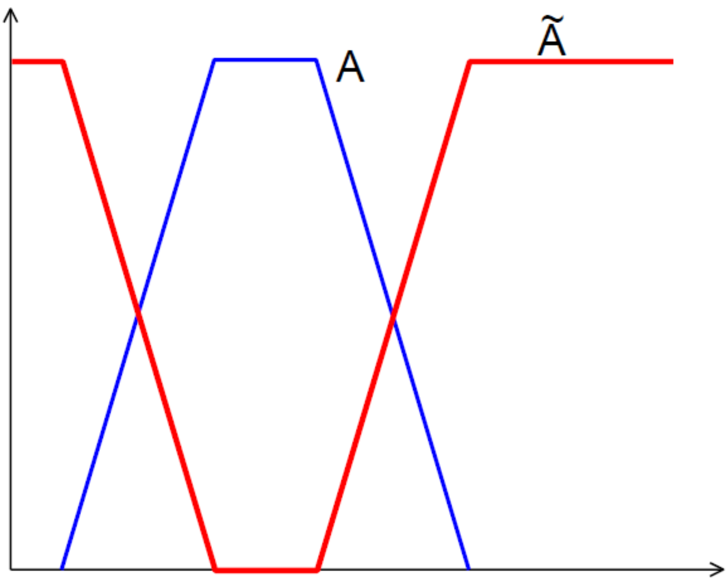
- **Zajednički skup dva *fuzzy* skupa - presjek *fuzzy* skupova**
  - Zajednički skup (presjek) dva *fuzzy* skupa A i B je *fuzzy* skup $A \cap B$ predstavljen pomoću funkcije pripadnosti:

$$\mu_{A \cap B}(x) = \mu_A(x) \wedge \mu_B(x) = \min\{\mu_A(x), \mu_B(x)\}$$



- **Komplement *fuzzy* skupa**
  - Računa se pomoću funkcije pripadnosti kao:

$$\mu_{\tilde{A}}(x) = 1 - \mu_A(x)$$



Fuzzy broj

- *Fuzzy* broj A je *fuzzy* skup predstavljen funkcijom pripadnosti $\mu_A(x)$ sa sledećim osobinama:
  1. $\mu_A(x)$ je definisana nad skupom $\mathbb{R}$
  2. $\mu_A(x)$ je konveksna
  3. $\mu_A(x)$ je normalna
  4. $\mu_A(x)$ je dio po dio neprekidna funkcija
- **Princip proširenja (ekstenzije)**:
  - Data je funkcija $f : X \to Y$. Ako je elemenat x iz skupa X *fuzzy* broj, pretpostavlja se da će y iz jednakosti $y = f(x)$ takođe biti *fuzzy* broj. Takav *fuzzy* broj se određuje primjenom pravila proširenja:

$$A = \frac{\mu_1}{x_1} + \frac{\mu_2}{x_2} + \ldots + \frac{\mu_n}{x_n}$$

$$f(A) = f(\frac{\mu_1}{x_1} + \frac{\mu_2}{x_2} + \ldots + \frac{\mu_n}{x_n}) = \frac{\mu_1}{f(x_1)} + \frac{\mu_2}{f(x_2)} + \ldots + \frac{\mu_n}{f(x_n)}$$

- Ako je više od jednog elementa skupa X preslikano funkcijom f u isti element y iz Y, uzima se maksimum među njihovim stepenima pripadnosti:

$$\mu_{f(A)}(y) = \max_{x_i \in X, f(x_i)=y} (\mu_A(x_i))$$

- U opštem slučaju, kada je f funkcija koja preslikava n - dimenzionalni prostor $X_1 \times X_2 \times \ldots \times X_n$ u jednodimenzionalni prostor Y, tako da je $y = f(x_1, x_2, \ldots, x_n)$ i skupovi $A_1, A_2, \ldots, A_n$ *fuzzy* skupovi u $X_1, X_2, \ldots, X_n$ - definiše se *fuzzy* skup B principom proširenja kao:

$$\mu_B(y) = \begin{cases} \max_{(x_1,x_2,\ldots,x_n)=f^{-1}(y)} \lfloor \min_i \mu_{A_i}(x_i) \rfloor & \text{, ako je } f^{-1}(y) \neq 0 \\ 0 & \text{, ako je } f^{-1}(y) = 0 \end{cases}$$

Fuzzy relacije

- Neka su X i Y dva univerzalna skupa. Tada je relacija:

$$R = \{(x,y), \mu_R(x,y) | (x,y) \in X \times Y)\}$$

  binarna *fuzzy* relacija.
- Primjer:

Primer

Neka su:

R_1 = „x je relevantno za y“,

R_2 = „y je relevantno za z“

dve fuzzy binarne relacije definisane na X×Y i Y×Z, gde je

X={1, 2, 3}, Y={α, β, γ, δ} Z={a,b},

i neka su R1 i R2 predstavljeni sledećim matricama

$$R_1 = \begin{array}{c c} \begin{array}{cccc} \alpha & \beta & \gamma & \delta \end{array} & \\ \begin{bmatrix} 0.1 & 0.3 & 0.5 & 0.7 \\ 0.4 & 0.2 & 0.8 & 0.9 \\ 0.6 & 0.8 & 0.3 & 0.2 \end{bmatrix} & \begin{array}{c} 1 \\ 2 \\ 3 \end{array} \end{array} \qquad R_2 = \begin{array}{c c} \begin{array}{cc} a & b \end{array} & \\ \begin{bmatrix} 0.9 & 0.1 \\ 0.2 & 0.3 \\ 0.5 & 0.6 \\ 0.7 & 0.2 \end{bmatrix} & \begin{array}{c} \alpha \\ \beta \\ \gamma \\ \delta \end{array} \end{array}$$

Potrebno je odrediti $R_1 \circ R_2$ = „x je relevantno za z“ i bazira se na R_1 i R_2.

Radi jednostavnosti odrediće se samo stepen zavisnosti između elemenata $2 \in X$ i $a \in Z$.

Max-min kompozicijom se dobija

$\mu_{R1 \circ R2}(2,a) = \max(0.4 \wedge 0.9,\ 0.2 \wedge 0.2,\ 0.8 \wedge 0.5,\ 0.9 \wedge 0.7) =$

$= \max(0.4,\ 0.2,\ 0.5,\ 0.7) = 0.7$

Max-proizvod kompozicija daje

$\mu_{R1 \circ R2}(2,a) = \max(0.4 \cdot 0.9,\ 0.2 \cdot 0.2,\ 0.8 \cdot 0.5,\ 0.9 \cdot 0.7) =$

$= \max(0.36,\ 0.04,\ 0.40,\ 0.63) = 0.63$

- Uobičajene *fuzzy* relacije su:
  - x je blizu y
  - x zavisi od y
  - x liči na y
  - ako je x veliko, y je malo (posebno interesantno za upravljanje)

Max **-** ***min*** **kompozicija**

- Neka su R_1 i R_2 *fuzzy* relacije definisane u $X \times Y$ i $Y \times Z$, respektivno. *Max-min* kompozicija R_1 i R_2 je *fuzzy* skup definisan sa:

$$\mu_{R_1 \circ R_2}(x,z) = \max_y \min(\mu_{R_1}(x,y), \mu_{R_2}(y,z)) = \vee_y(\mu_{R_1}(x,y) \wedge \mu_{R_2}(y,z))$$

  gdje važi da su operatori $\vee$ i $\wedge$ jednaki $\max$ i $\min$, respektivno.
- Naziva se još i *max-min* proizvod.
- Osobine *max-min* proizvoda
  - *Asocijativnost*

$$R \circ (S \circ T) = (R \circ S) \circ T$$

  - *Distributivnost u odnosu na uniju*

$$R \circ (S \cup T) = (R \circ S) \cup (R \circ T)$$

  - *Slaba distributivnost u odnosu na presjek*

$$R \circ (S \cap T) \subseteq (R \circ S) \cap (R \circ T)$$

  - *Monotonost*

$$S \subseteq T \implies (R \circ S) \subseteq (R \circ T)$$

- Postoji i ***max*-proizvod kompozicija**:

$$\mu_{R_1 \circ R_2}(x,z) = \max_y \{\mu_{R_1}(x,y)\mu_{R_2}(y,z)\}$$

Fuzzy AKO-ONDA pravila

Lingvistička promjenljiva

- Promjenljiva čije su dozvoljene vrijednosti riječi prirodnog jezika.
- Primjer:
  - Ako promjenljiva *STAROST* može imati vrijednosti *star*, *mlad*, *veoma mlad*,..., tada je ona lingvistička promjenljiva.
  - U tom slučaju, vrijednosti *star*, *mlad*, *veoma mlad*,... nazivaju se *lingvističke vrijednosti*.
- Ovakva promjenljiva se često naziva i *fuzzy promjenljiva*, a njene vrijednosti *lingvističkim modifikatorima*.
- Dodavanjem modifikatora i kombinovanjem sa veznicima dobijamo složene lingvističke izraze.
- Primjer:
  - *ni veoma mlad, ni veoma star*
- Operatori modifikacije:
  - *Množenje skalarom*

$$\mu_{\alpha A}(x) = \alpha\mu_A(x);\ \alpha \in \mathbb{R}^+ \text{ i } \forall x \in X, \alpha\mu_A(x) \leq 1$$

  - *Stepenovanje*

$$\mu_A^\alpha(x) = (\mu_A(x))^\alpha;\ \alpha \in \mathbb{R}^+$$

  - *Normalizacija*

$$NORM(A) = \frac{A}{\max_x \mu_A(x)}$$

  - *Koncentrisanje*

$$CON(A) = A^2;\ \mu_{CON(A)} = (\mu_A(x))^2$$

  - *Proširenje*

$$DIL(A) = A^{\frac{1}{2}};\ \mu_{DIL(A)} = (\mu_A(x))^{\frac{1}{2}}$$

  - *Pojačavanje kontrasta*

$$\mu_{INT(A)} = \begin{cases} 2(\mu_A(X))^2 & \text{, za } 0 \leq \mu_A(x) \leq 0.5 \\ 1 - 2(1-\mu_A(x))^2 & \text{, za } 0.5 \leq \mu_A(x) \leq 1 \end{cases}$$

- Ispred osnovnih lingvističkih vrijednosti (*mlad*, *star*,...) dodaju se modifikatori (*veoma*, *ponešto*,...).
- Primjer:

    - *veoma* $A = CON(A)$
    - *manje-više* $A = DIL(A)$
    - *ponešto* $A = NORM(A \wedge \neg CON(A))$
    - *prilično* $A = NORM(INT(A) \wedge \neg INT(CON(A)))$
    - *izuzetno* $A = NORM(INT(A))$
- Lingvistički modifikatori i veznici se mogu definisati zavisno od upotrebe i namjene.

Fuzzy proporcija

- Koristi se za predstavljanje tvrđenja koja sadrže lingvističke vrijednosti.
- Primjer:
    - Ako u tvrđenju "P : x je A", A predstavlja *fuzzy* skup, onda je P *fuzzy proporcija*
- Kada je A *fuzzy* skup, tada su moguće vrijednosti promjenljive x *fuzzy* skupovi. Tada se x naziva *fuzzy promjenljiva*.
- Kada je A *lingvistička* vrijednost, tada su moguće vrijednosti promjenljive x *lingvističke* vrijednosti. Tada se x naziva *lingvistička promjenljiva*.
- Primjer:
    - P je proporcija koja opisuje starost osobe. Lingvistička promjenljiva x je starost.
    - Rečenica "*Osoba je mlada*" piše se kao *STAROST(osoba) = mlad*.

Fuzzy AKO-ONDA pravila

- Naziva se i *fuzzy* implikacija.
- Obično se piše u obliku *AKO* x *je* A, *ONDA* y *je* B, gdje su A i B lingvističke vrijednosti definisane nad univerzalim skupovima X i Y.
    - x *je* A naziva se *antecedent* (prethodnik, pretpostavka, premisa, činjenica).
    - y *je* B naziva se *konsekvenca* (konkluzija, zaključak, posljedica).
    - *Fuzzy* implikacija se u kraćem obliku zapisuje kao $A \to B$ (*AKO* A *ONDA* B)

Fuzzy zaključivanje

- U binarnoj logici, uobičajen način zaključivanja naziva se *modus ponens*.
    - Na osnovu poznate činjenice A i pravila $A \to B$, pokazuje se da važi zaključak B.
- Kod *fuzzy* zaključivanja, koristi se *stepen saglasnosti* između činjenice i preduslova pravila.
    - *Premisa 1* (*antecedent*): x *je* A^*
    - *Premisa 2* (*pravilo*) : *AKO* x *je* A, *ONDA* y *je* B
    - *Zaključak*: y *je* B^*

- A^* je blisko sa A i B^* je blisko sa B
- Ovakvo zaključivanje naziva se *generalizovan modus ponens*.

- *Fuzzy* zaključivanje definiše se formalno kao:

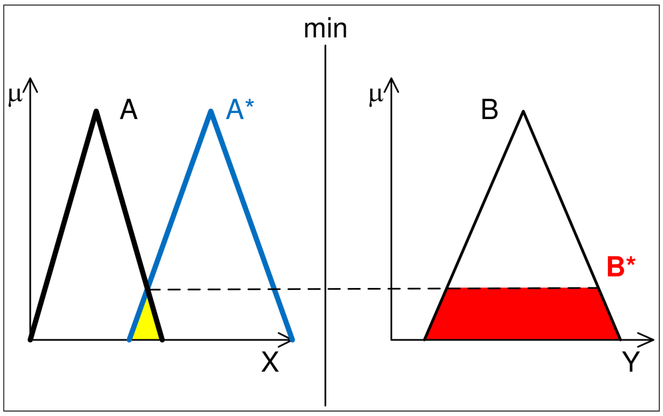
$$\mu_{B^*}(y) = \max_x \min\{\mu_{A^*}(x), \mu_R(x,y)\} = \vee_x(\mu_{A^*}(x) \wedge \mu_R(x,y))$$

- Slučajevi zaključivanja:
  - *Jedno pravilo sa jednom pretpostavkom*
    - Zapisuje se kao

$$\mu_{B^*}(y) = \vee_y(\mu_{A^*}(x) \wedge \mu_A(x)) \wedge \mu_B(y) = \omega \wedge \mu_B(y)$$

    - Dakle, prvo se određuje ω kao maksimum minimuma od $\mu_{A^*}(x)$ i $\mu_A(x)$. Zatim se određuje rezultujuća funkcija pripadnosti skupu B^*, koja je jednaka stepenu pripadnosti skupu B ograničenoj sa ω.

$$\mu_{B^*}(y) = \vee_x[\mu_{A^*}(x) \wedge \mu_A(x)] \wedge \mu_B(y) = \omega \wedge \mu_B(y)$$



Intuitivno, ω predstavlja meru stepena poverenja premisi (činjenici – prethodnom delu pravila). Ova mera „prolazi“ kroz AKO-ONDA pravilo i daje stepen poverenja (stepen pripadnosti) fuzzy skupu B (B* na slici), koji nije veći od ω.

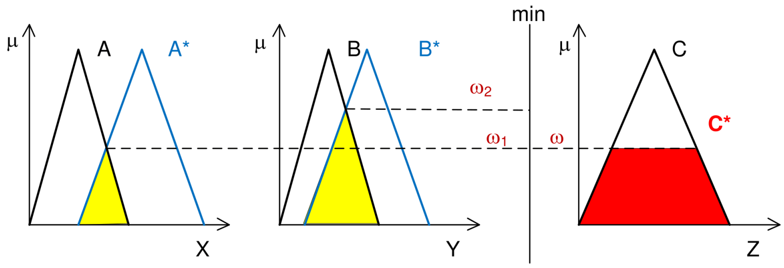
  - *Jedno pravilo sa više pretpostavki*
    - Zapisuje se kao:
      - *Premisa 1* (*antecedent*): x *je* A^* i y *je* B^*
      - *Premisa 2* (*pravilo*): *AKO* x *je* A i y *je* B, *ONDA* z *je* C
      - *Zaključak*: z *je* C^*
    - Konsekvenca C^* može se izraziti kao:

$$\mu_{C^*}(z) = \{\vee_x(\mu_{A^*}(x) \wedge \mu_A(x))\} \wedge \{\vee_y(\mu_{B^*}(x) \wedge \mu_B(x))\} \wedge \mu_C(z)$$

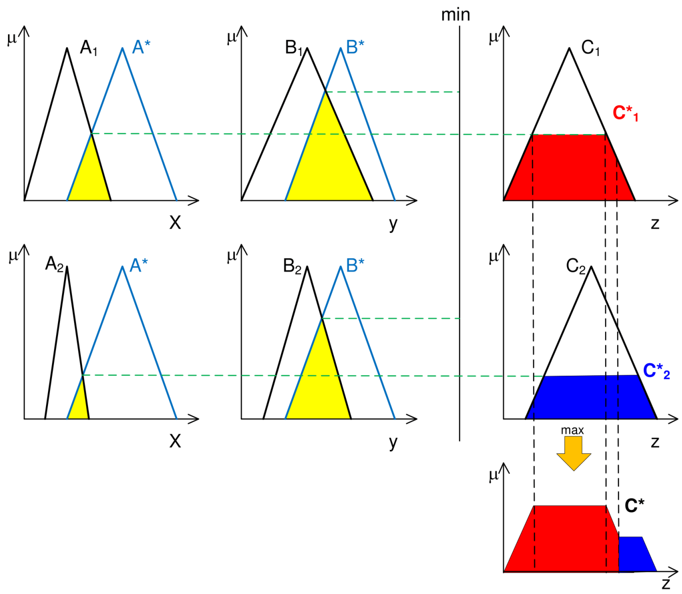
$$\mu_{C^*}(z) = (\omega_1 \wedge \omega_2) \wedge \mu_C(z)$$

jedno pravilo sa više pretpostavki

$$\mu_{C^*}(z)=\{\vee_x[\mu_{A^*}(x)\wedge\mu_A(x)]\}\wedge\{\vee_y[\mu_{B^*}(y)\wedge\mu_B(y)]\}\wedge\mu_C(z)$$



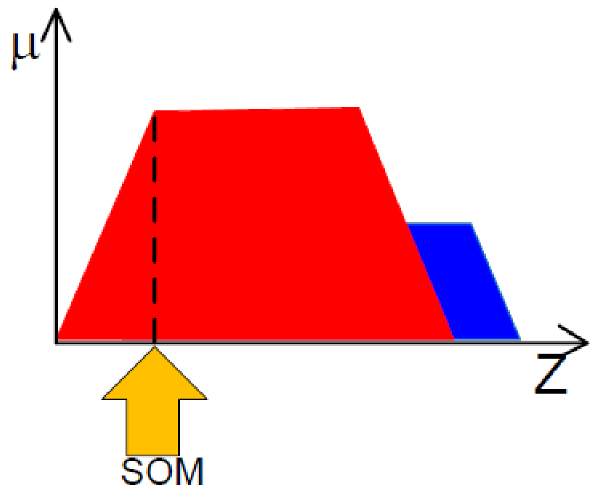
- *Više pravila sa više pretpostavki*
  - Predstavlja se kao unija *fuzzy* relacija odgovarajućih *fuzzy* pravila
  - Zapisuje se kao:
    - *Premisa 1* (*antecedent*): x *je* A^* i y *je* B^*
    - *Premisa 2* (*pravilo*): *AKO* x *je* A_1 i y *je* B_1, *ONDA* z *je* C_1
    - *Premisa 3* (*pravilo*): *AKO* x *je* A_2 i y *je* B_2, *ONDA* z *je* C_2
    - *Zaključak*: z *je* C^*



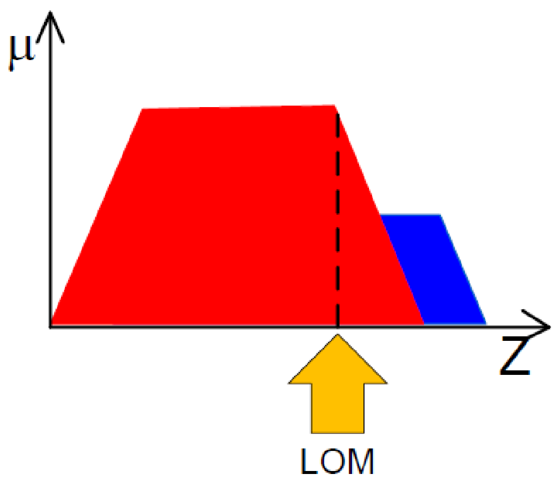
- Generalno, proces *fuzzy* zaključivanja može biti podijeljen u četiri osnovna koraka:
  1. *Stepen kompatabilnosti* - upoređivanje poznatih činjenica sa pretpostavkom *fuzzy* pravila i utvrđivanje stepena kompatabilnosti sa svakom pojedinačnom premisom.
  2. *Stepen aktivacije* - indikacija koliko je zadovoljena premisa *fuzzy* pravila.
  3. *Indukovana funkcija pripadnosti* - primjena stepena aktivacije na svaki pojedinačni posljedični dio *fuzzy* pravila.
  4. *Ukupni izlaz funkcija pripadnosti* - kombinacija svih konsekvenci *fuzzy* pravila.

Strategije defazifikacije

- *Defazifikacija* predstavlja mapiranje iz prostora *fuzzy* upravljačkih akcija, definisanog nad izlaznim univerzumom posmatranja, u prostor *nefuzzy* (diskretnih) upravljačkih akcija.
- Primjenjuje se zbog potrebe da se u mnogim praktičnim aplikacijama koristi diskretna vrijednost upravljanja.
- Teži se proizvesti *nefuzzy* upravljanje koje najbolje predstavlja moguću raspodjelu zaključene *fuzzy* upravljačke akcije.
- Najčešće korišćene straterije:
    - *Kriterijum najmanje vrijednosti maksimuma* (*SoM - Smallest of Maximum*)
        - Daje prvu najmanju tačku u kojoj mogućnost raspodjele upravljačke akcije dostiže maksimalnu vrijednost.



    - *Kriterijum najveće vrijednosti maksimuma* (*LoM - Largest of Maximum*)
        - Daje poslednju najveću tačku u kojoj mogućnost raspodjele upravljačke akcije dostiže maksimalnu vrijednost.

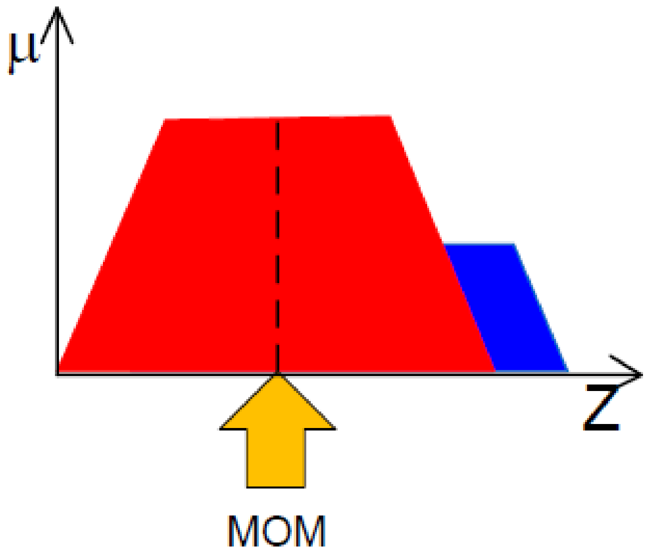


    - *Kriterijum sredine maksimuma* (*MoM - Mean of Maximum*)

- Stvara upravljanje koje predstavlja sredinu vrijednosti svih lokalnih upravljačkih akcija čine funkcije članice dostižu maksimum.
- Izražava se kao:

$$z_o = \frac{\sum_{j=1}^{l} z_j}{l}$$

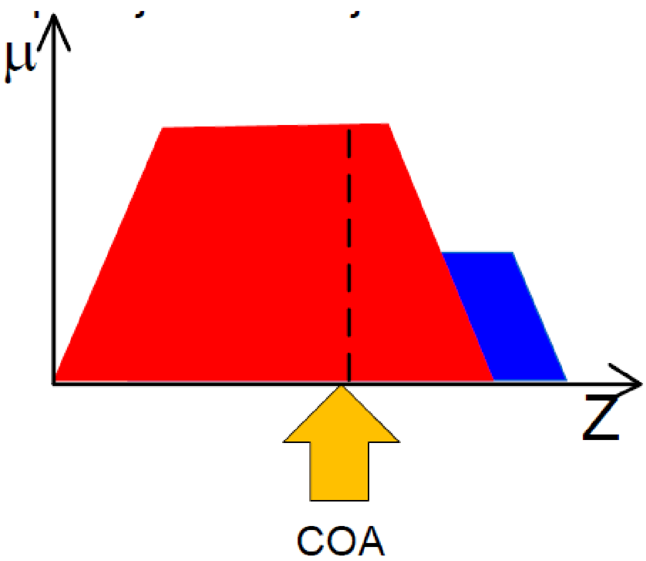
gdje su z_j one vrijednosti gdje se dostiže maksimum funkcije pripadnosti $\mu_Z(z_j)$



- *Kriterijum centra mase* (*CoA - Center of Area*, *CoG - Center of Gravity*)
  - Određuje diskretnu upravljačku akciju tako da se lijevo i desno od nje nalaze dijelovi upravljačke površine "jednakih težina".
  - Izražava se kao:

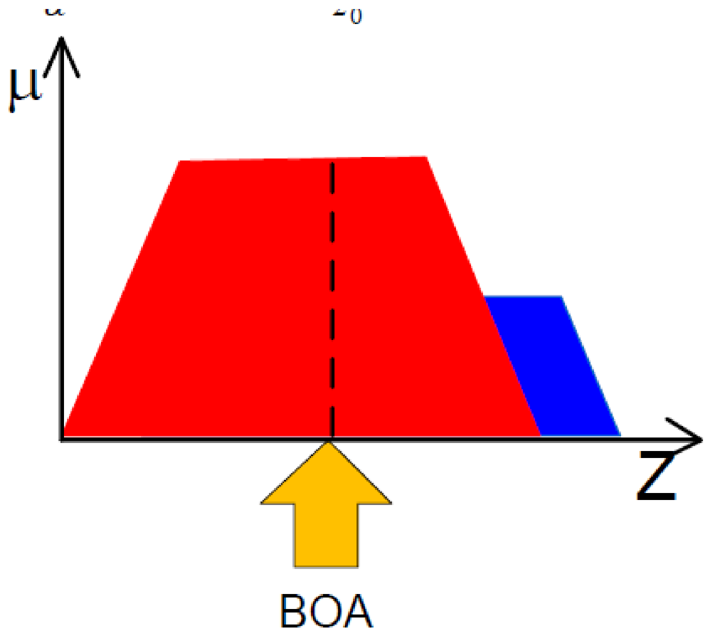
$$z_o = \frac{\sum_{j=1}^{l} z_j \mu_Z(z_j)}{\sum_{j=1}^{l} \mu_Z(z_j)}$$

$$z_o = \frac{\int z\mu_z \, dz}{\int \mu_z \, dz}$$



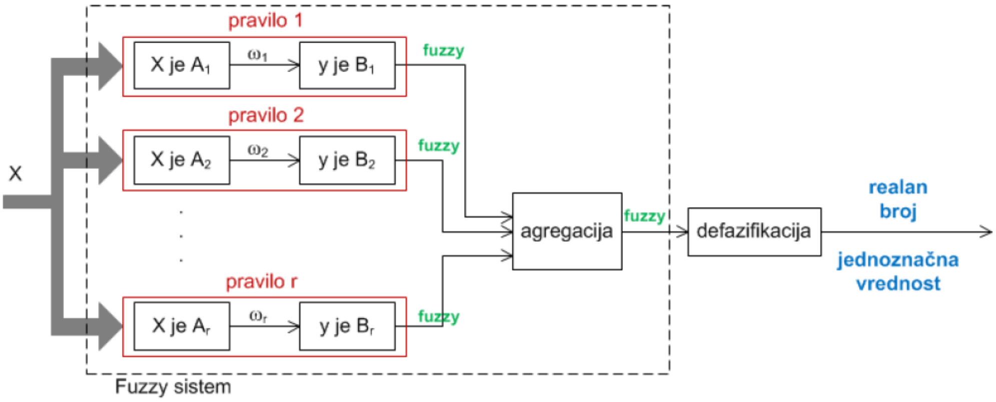
- *Kriterijum bisekcije oblasti* (*BoA - Bisection of Area*)
  - Daje tačku koja dijeli izlaznu oblast na dva dijela jednakih površina.
  - Izražava se kao:

$$\int_{\alpha}^{z_o} \mu_Z(z)\, dz = \int_{z_o}^{\beta} \mu_Z(z)\, dz$$



Fuzzy sistemi odlučivanja

- Baziraju se na teoriji *fuzzy* skupova, *fuzzy AKO-ONDA* pravilima i *fuzzy* zaključivanju.
- Upotrebljavaju se u automatskom upravljanju, klasifikaciji podataka, ekspertskim sistemima, robotika,...
- Osnovna struktura:
  - *Baza pravila* (*rule base*) - sadrži definisana *fuzzy* pravila.
  - *Baza podataka* (*rječnik*) - definiše funkcije pripadnosti korištene u okviru *fuzzy* pravila (*fuzzy* skupovi).
  - *Mehanizam zaključivanja* - izvršava procedure izvođenja zaključaka, odnosno formiranja posljedica na osnovu datih premisa.
- Ulazi u *fuzzy* sistem zaključivanja mogu biti ili *fuzzy* ili diskretni (*crisp*) ulazi, dok je izlaz uvijek *fuzzy* skup.
- Kod regulacije (Pretpostavimo da imamo dva *fuzzy* pravila):
  - *FSO* mora imati jasno definisane izlaze i u tom slučaju koristi se neka od metoda *defazifikacije*.



- Ulazne vrijednosti su diskretne (*crisp*) i obično se mjere senzorima.
- U nekim slučajevima, ulazne (diskretne) vrijednosti se moraju pretvoriti u *fuzzy* skupove. U opštem slučaju, mogu se interpretirati kao *fuzzy* ton.
- Stepeni aktivacije (težinski faktori) prvog i drugog pravila mogu se izraziti kao:

$$\omega_1 = \mu_{A_1}(x_0) \wedge \mu_{B_1}(y_0)$$

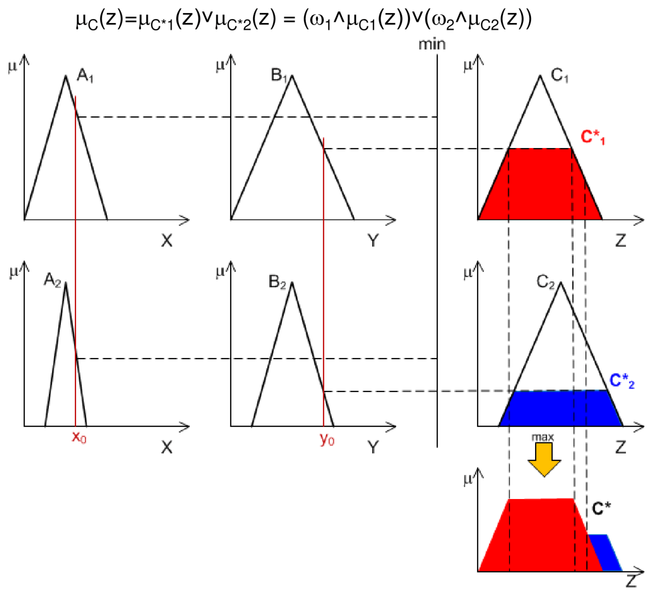
$$\omega_2 = \mu_{A_2}(x_0) \wedge \mu_{B_2}(y_0)$$

- Najčešće korišteni sistemi zaključivanja su:
  - *Mamdani fuzzy model*
    - Koristi *max-min* kompoziciju, tako da i-to pravilo dovodi do upravljačke akcije:

$$\mu_{C_i^*}(z) = \omega_i \wedge \mu_{C_i}(z)$$

    - Zaključna konsekvenca se dobija kao *max* kompozicija konsekvenci svih pravila:

$$\mu_C(z) = \mu_{C_1^*}(z) \vee \mu_{C_2^*}(z)$$

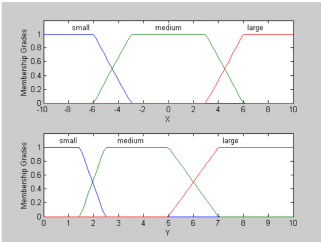


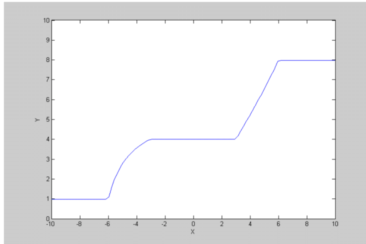
Mamdani Fuzzy model - primer

Max-min composition and centroid defuzzification were used.

R1 : If X is small then Y is small
R2 : If X is medium then Y is medium
R3 : If X is large then Y is large

X = input $\in [-10, 10]$
Y = output $\in [0, 10]$





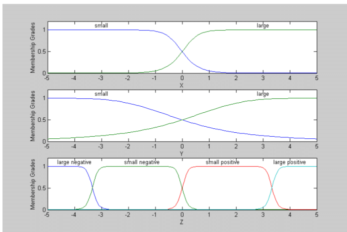
zaključak konsekvenca

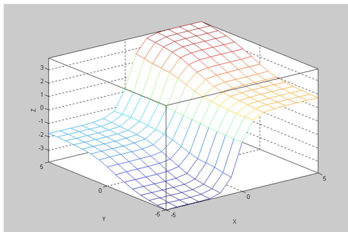
Mamdani Fuzzy models - Example

Max-min composition and centroid defuzzification were used.

R1: If X is small & Y is small then Z is negative large
R2: If X is small & Y is large then Z is negative small
R3: If X is large & Y is small then Z is positive small
R4: If X is large & Y is large then Z is positive large

$X, Y, Z \in [-5, 5]$





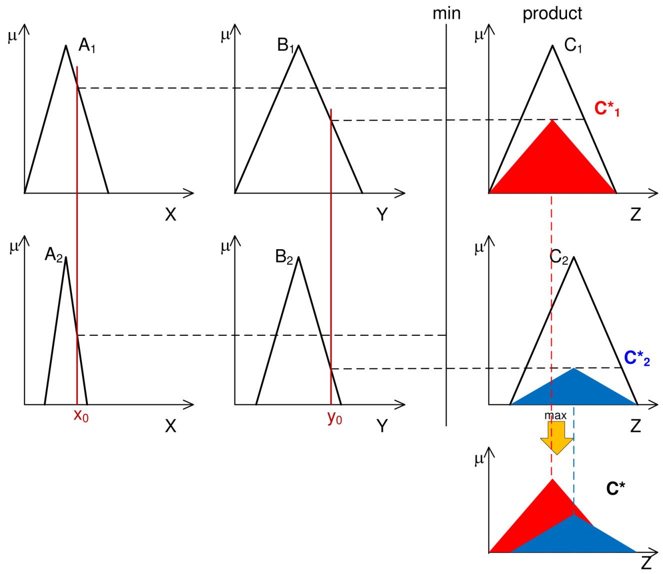
Zaključna konsekvenca

- *Larsenovo pravilo*
  - Koristi *max*-proizvod kompoziciju, tako da i-to pravilo dovodi do upravljačke akcije:

$$\mu_{C_i^*}(z) = \omega_i \times \mu_{C_i}(z)$$

  - Zaključna konsekvenca se dobija kao *max* kompozicija konsekvenci svih pravila:

$$\mu_C(z) = \mu_{C_1^*}(z) \vee \mu_{C_2^*}(z)$$



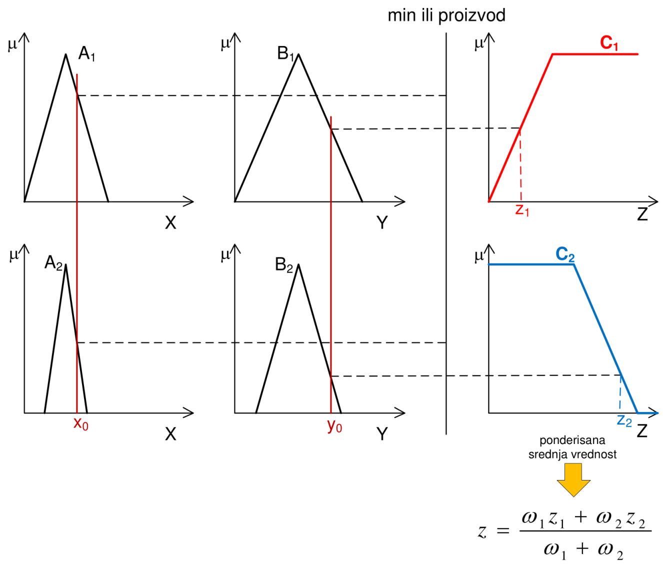
- *Tsukamoto metod zaključivanja*
  - Metod sa lingvističkim terminima tipa monotonih funkcija članica
  - i-to pravilo vodi do upravljačke akcije:

$$\mu_{C_i^*}(z_i) = \omega_i$$

  - Zaključna konsekvenca se dobija kao funkcija konsekvenci svih pravila:

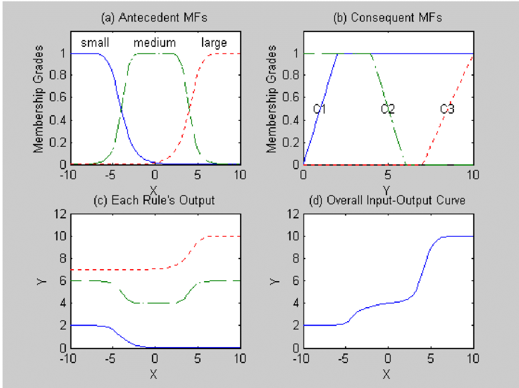
$$z = \frac{\sum_{i=1}^{n} \omega_i z_i}{\sum_{i=1}^{n} \omega_i}$$

  - Zaključna konsekvenca je diskretna upravljačka akcija.



Fuzzy Inference Systems – Tsukamoto Fuzzy models - Example

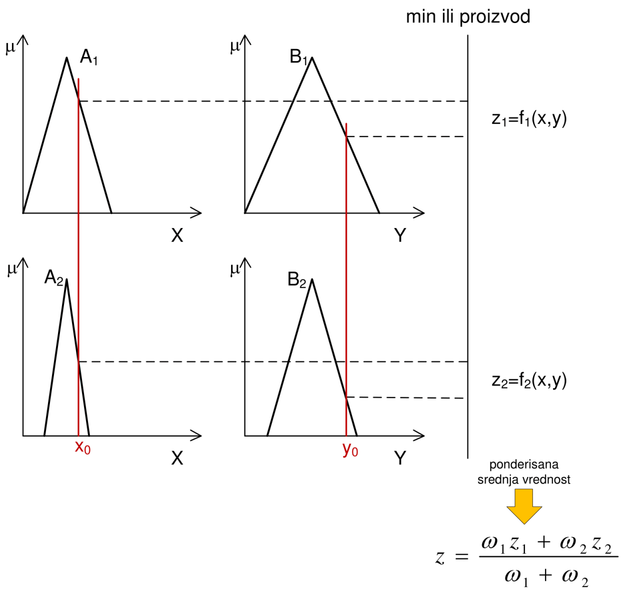
R1: If *X* is small then *Y* is C_1
R2: If *X* is medium then *Y* is C_2
R3: if *X* is large then *Y* is C_3



- *Takagi-Sugeno-Kang način odlučivanja*
  - Konsekvenca pravila je funkcija od ulaznih lingvističkih promjenljivih:
    - R_1: *AKO* x *je* A_i *i* y je B_i, *ONDA* $z_i = f_i(x, y)$
  - Zaključna konsekvenca se dobija kao funkcija konsekvenci svih pravila:

$$z = \frac{\sum_{i=1}^{n} \omega_i f_i(x, y)}{\sum_{i=1}^{n} \omega_i}$$

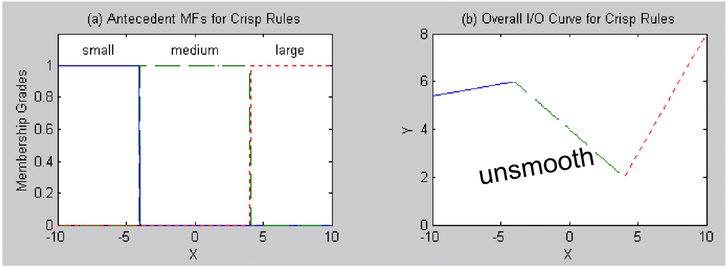
  - Zaključna konsekvenca je diskretna upravljačka akcija.



Fuzzy Inference Systems – Sugeno Fuzzy Models - Example

R1: If X is small then $Y = 0.1X + 6.4$
R2: If X is medium then $Y = -0.5X + 4$
R3: If X is large then $Y = X - 2$

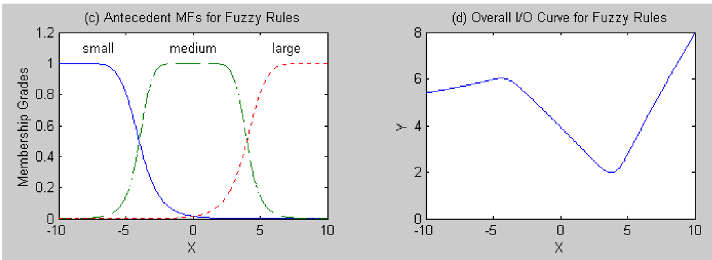
X = input $\in [-10, 10]$



Fuzzy Inference Systems – Sugeno Fuzzy Models - Example

R1: If X is small then $Y = 0.1X + 6.4$
R2: If X is medium then $Y = -0.5X + 4$
R3: If X is large then $Y = X - 2$

X = input $\in [-10, 10]$

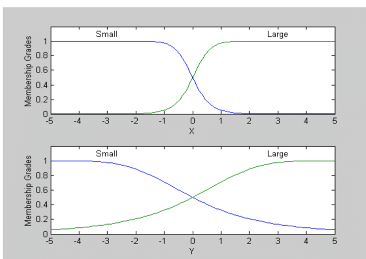


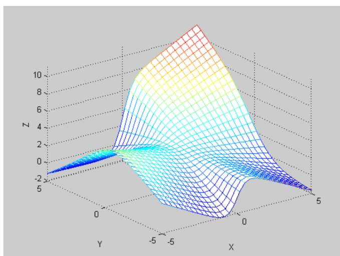
If we have smooth membership functions (fuzzy rules) the overall input-output curve becomes a smoother one.

Fuzzy Inference Systems – Sugeno Fuzzy Models - Example

R1: if X is small and Y is small then $z = -x + y + 1$
R2: if X is small and Y is large then $z = -y + 3$
R3: if X is large and Y is small then $z = -x + 3$
R4: if X is large and Y is large then $z = x + y + 2$

$X, Y \in [-5, 5]$





Fuzzy upravljanje

- Konvencionalno upravljanje - oslanjanje na matematički model sistema (*PID* kontroleri, *Lead-Lag* kontrola,...).
- *Fuzzy* upravljanje - intuitivno razumijevanje sistema.
  - *Real-time* ekspertski sistem koji "implementira ljudsko iskustvo o procesu", koje nije lako ispoljivo konvencionalnim kontrolerima ili diferencijalnim jednačinama već "situacionim, ljudski naklonjenim pravilima".
  - Prednosti *fuzzy* upravljanja nad konvencionalnim:
    - Implementiranje ekspertskog znanja radi većeg stepena automatizacije
      - Primjer: upravljanje hemijskim procesima zahtjeva ljudskog operatera radi podešavanja parametara kontrolera. Ljudsko prisustvo je, u ovom slučaju, moguće zamijeniti *fuzzy* kontrolerom.
    - Robusno nelinearno upravljanje
      - *PID* kontroleri imaju problem pri ogromnim vanjskim poremećajima i stabilizaciji sistema, dok *fuzzy* kontroleri mogu da se "adaptiraju" problemima kontrole i stabilizacije.
    - Redukcija vremena razvijanja kontrolera
  - Četiri komponente *fuzzy* kontrolera:
    - *Rule base* - baza pravila, sadrži znanje u formi pravila koje govore kako najbolje kontrolisati sistem.
    - *Inference mechanism* - sistem odlučivanja, koji odlučuje koja pravila su koliko relevantna za aktivaciju
    - *Fuzzification* - fazifikacija, modifikuje *input crisp* vrijednosti u *fuzzy* vrijednosti.
    - *Defuzzification* - defazifikacija, radi obratno od fazifikacije - transformiše *fuzzy* odluku u *crisp* vrijednost.
  - Fuzija *fuzzy* i *PID* kontrole (*High-Level* kontroleri)
    - *Fuzzy* supervizor koji se koristi da namjesti parametre *PID* kontrolera nakon svakog *sample*-a kako bi *PID* bio dobar za sve opsege kontrolisane varijable.
    - Djelovanje *fuzzy* i *PID* kontrolera zajedno - *PID* kontroler je dobar kada je proces radi u normalnim uslovima, ali ako se dese jaki poremećaji, tada se iskorištava *fuzzy* koji dovodi proces nazad u normalno stanje što je brže moguće.
    - Indirektna koordinacija *fuzzy* kontrolera koja modifikuje strateriju kontrole generalno, npr. namješta novi *set point* za *PID* kontroler(e).

Genetski algoritam

- Stohastički metod koji ne koristi izvode funkcija.
- Bazirani na prirodnim pojavama *selekcije* i *evolucije*.
- Karakteristike:
    - Predstavlja proceduru paralelnog pretraživanja prostora - pogodni za paralelizaciju.
    - Podjednako primjenljivi u rješavanju kontinualnih i diskretnih optimizacionih problema.
    - Manje podložni upadanju u lokalni minimum (čime su opterećeni mnogi optimizacioni problemi).
    - Veliki stepen prilagodljivosti čime se olakšava primjena u okviru problema strukturne i parametarske identifikacije složenih modela
- Svaku tačku parametarskog prostora kojeg pretražujemo predstavlja preko binarnog niza (stringa) koji se zove **hromozom**.
    - U modernijim primjenama se posmatra kodirano kao niz drugačijih elemenata, npr. realnih brojeva, stringova,...
    - Svakom hromozomu je pridružena određena **vrijednost podobnosti** (**stepen prilagođenosti**, ***fitness***, **kriterijum optimalnosti**).
- Umjesto jedne izolovane tačke, GA često koristi skup tačaka koji se naziva **populacija** i koja evoluira u pravcu što boljeg "zadovoljenja" kriterijuma optimalnosti.
    - U svakoj **generaciji** (novoj iteraciji) dobijamo novu populaciju pomoću **genetskih operatora**.
- Osnovne komponente genetskih algoritama su **kodiranje**, **prilagođenost**, **selekcija**, **ukrštanje**, **mutacija** i **elitizam**.
- **Jedinke** koje u većoj mjeri zadovoljavaju kriterijumsku funkciju imaju veću šansu da prežive i učestvuju u budućim ukrštanjima.

Kodiranje

- Tačke parametarskog prostora predstavljaju se nizovima (stringovima) binarnih oznaka.
- Šema kodiranja predstavlja način transformacije specifičnog znanja vezanog za određeni problem u okvire genetskog algoritma, što ujedno predstavlja i korak koji će direktno uticati na uspješnost genetskog algoritma.
- Moguće je kodirati realne brojeve, diskretne vrijednosti,...
- Generalna preporuka jeste da se genetski operatori ukrštanja i mutacije projektuju zajedno sa šemom kodiranja kako bi što bolje odgovorili zahtjevima specifičnih problema.
- Uobičajeni postupci kodiranja su *binarno* i *Grej* (*Gray*) kodiranje.

- Klasičan način predstavljanja hromozoma (jedinke) u okviru GA jeste binarni vektor fiksne dužine. U slučaju N-dimenzionalnih problema, svaka jedinka se sastoji od N vrijednosti (koordinata) od kojih je svaka kodirana binarnim stringom.

Binarno kodiranje

- Svaka koordinata parametarskog prostora se predstavlja **binarnim stringom** dužine *D* (*D-dimenzionalni* binarni vektor).
- Ako se promenljiva *z* sa kontinualnog intervala $[z_{min}, z_{max}]$ želi **konvertovati** u D-bitni string može se primeniti sledeći obrazac:

$$\left(2^D - 1\right)\frac{z - z_{min}}{z_{max} - z_{min}}$$

Primer: tačka (2,3,7) trodimenzionalnog parametarskog prostora može biti predstavljena sledećim binarnim stringom:

$$\underbrace{010}_{2}\underbrace{011}_{3}\underbrace{111}_{7}$$

Svaka koordinata je kodirana kao **gen** sastavljen od tri binarna bita.

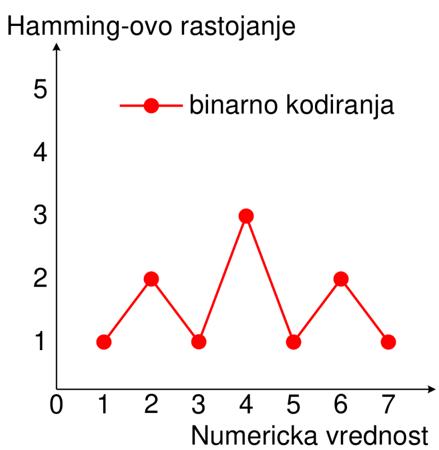
Posmatra se interval *[0, 7]* i neka promenljiva *z* može poprimati celobrojne vrednosti sa tog intervala. Binarno kodiranje datih vrednosti promenljive *z* se može predstaviti sledećom tabelom:

Koordinata	0	1	2	3	4	5	6	7
Binarno kodiranje	000	001	010	011	100	101	110	111

Iako se binarno kodiranje veoma često koristi ono sa sobom nosi i značajan problem – **Hamming-ove litice**.

Hamming-ova litica se formira u slučaju kada dve numerički **bliske vrednosti** imaju veoma **udaljene kodne oznake**, kao što su npr. 3 i 4 iz gornje tabele. Neka je 3 optimalno rešenje problema, a 4 tekuće rešenje. Da bi se kodna oznaka promenila i postiglo optimalno rešenje potrebno je promeniti vrednost svih bitova kodne oznake (u ovom slučaju tri bita). Problem Hamming-ovih litica se može ilustrovati sledećim grafikom:

- Binarno kodiranje jeste često ali sa sobom nosi problem *Hamingonovih litica*.



Problem Hamming-ovih litica se može rešiti uvođenjem **sivog (Grey) kodiranja**.

- *Grejev* kod koristi osobinu da *Hammingovo* rastojanje između numerički susjednih podataka bude 1.

Grejevo (Sivo) kodiranje (Frank Gray, 1947)

- Nakon primene sivog kodiranja **Hamming-ovo rastojanje** između numerički susednih podataka poprima **vrednost 1**.
- **Primena** sivog kodiranja se može ilustrovati na prethodnom primeru sledećom tabelom:

Koordinata	0	1	2	3	4	5	6	7
Binarno kodiranje	000	001	010	011	100	101	110	111
Sivo kodiranje	000	001	011	010	110	111	101	100

- Binarno kodirane oznake se mogu jednostavno transformisati u sivo kodirane primenom sledeće transformacije

$$g_1=b_1$$
$$g_k=b_{k-1}b_k^*+b_{k-1}^*b_k$$

gde je: b_k k-ti bit binarnog broja $b_1b_2...b_K$; b_1 najstariji bit; b_k^* je neb_k, $+$ znači logičko "ili"; ***množenje*** logičko "i".

Generisanje početne populacije

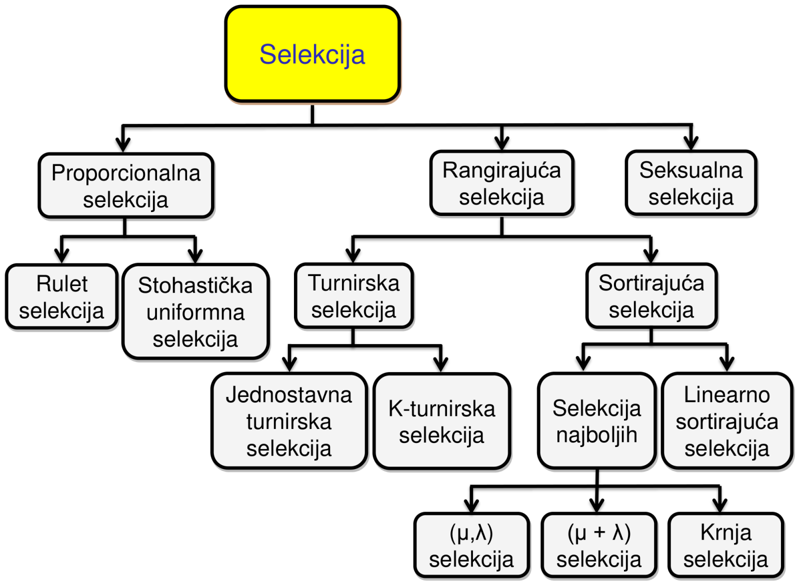
- Bitni faktori pri generisanju početne populacije su
  - *Veličina populacije*
    - Veća populacija daje više genetskog materijala, pa samim time i veću šansu za nalaženje optimalne vrijednosti, ali glomazno usporava izvršavanje algoritma.
    - Prednost manje populacije je brzina, dok postoji mogućnost za promjenom veličine populacije kako algoritam napreduje.
  - *Princip generisanja*
    - Najčešće slučajno, dok nekad uniformno (ali problem - vještačka tvorevina gena).
    - Savjetuje se da se u početnu populaciju uključe rješenja dobijena drugim optimizacionim algoritmima, ako je to moguće.

Prilagođenost (*fitness*)

- Prvi korak nakon generisanja populacije jeste izračunavanje stepena prilagođenosti svakog člana populacije.
- Uobičajeno je da ishod računanja bude pozitivna vrijednost, te je korisno nekad primjeniti *skaliranje* i/ili *translaciju*.
- Alternativno, može se upotrijebiti i **rang**, gdje se jedinke redaju od najgore ka najboljoj (tako davajući šansu za ukrštanjem i onim jedinkama koje nemaju "najbolji" *fitness* zbog potrebe različitosti populacije, koja je ključna za efikasno pretraživanje parametarskog prostora).

Selekcija

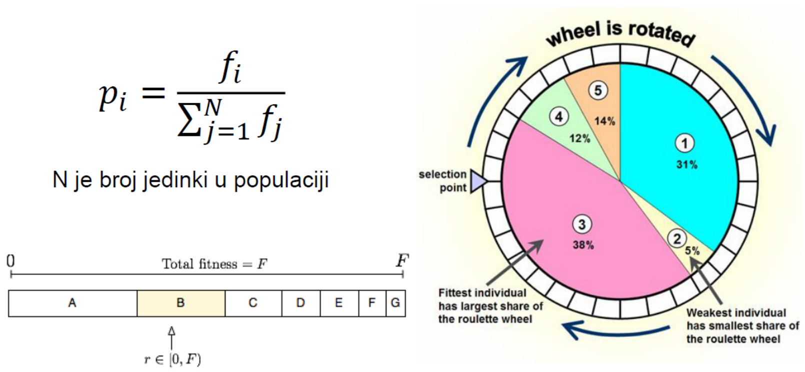
- Nakon procjene stepena prilagođenosti, potrebno je formirati novu populaciju na osnovu postojeće.
- Selekcijom se biraju "roditelji" koji će učestvovati u produkciji naredne generacije i taj postupak je analogan "opstanku najboljih, najprilagođenijih".
- Uobičajeno je da se biraju članovi onom vjerovatnoćom proporcionalnom njenoj stepenu prilagođenosti.
- Vrste selekcija:



Rulet selekcija

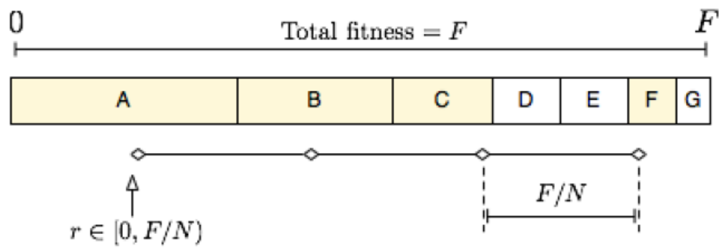
- Vjerovatnoća da će jedinka biti selektovana za ukrštanje proporcionalna je njenom stepenu prilagođenosti

$$p_i = \frac{f_i}{\Sigma_{j=1}^{N} f_j}$$



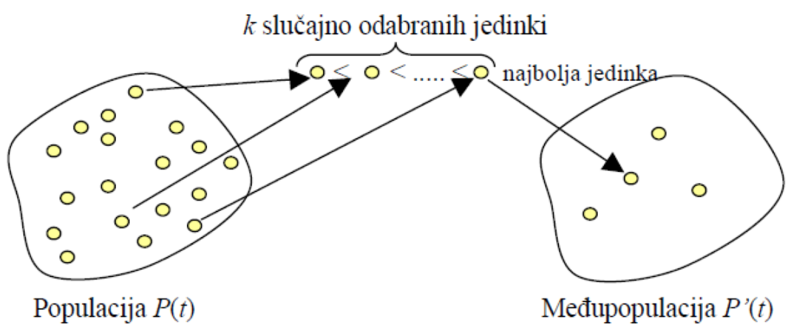
Stohastička uniformna selekcija

- Razvijena od rulet selekcije.
- Kod ruleta se N roditelja bira tako što se generiše N slučajnih brojeva na intervalu $[0, F)$, dok kod stohastičke uniformne selekcije generiše se samo jedan slučajan broj na intervalu $[0, \frac{F}{N})$, pa se N roditelja bira tako što se ova vrijednost raspodjeli N puta po čitavom opsegu sa jednakim međusobnim rastojanjem (*uniformno raspoređeni*).
- Stohastičkom uniformnom selekcijom postiže se da i slabije jedinke budu izabrane kao roditelji.



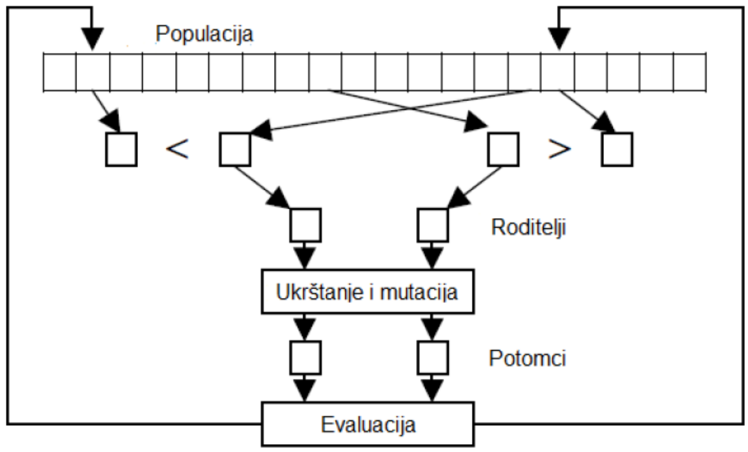
K-turnirska selekcija

- Sprovodi se više "turnira" među K jedinki izabranih na slučajan način iz populacije.
- Na turniru, između slučajno odabranih K jedinki pronalazi se najbolja. N turnira daje N roditelja.
- Mijenjanjem veličine turnira povećava se pritisak selekcije, što znači da slabije jedinke imaju manju šansu da budu izabrani.
- $K - 1$ najslabijih jedinki nemaju nikakve šanse (jedan od načina da najlošije jedinke ne budu nikad odabrane).
- Odlikuje je brzina izvršavanja.



Jednostavna turnirska selekcija

- Poseban slučaj K-turnirske selekcije kada je $K = 2$.
  - U jednom koraku, slučajnim postpukom se bira dva para jedinki.
  - U svakom paru, lošije jedinke se eliminišu.
  - Ukrštanjem boljih jedinki se generišu dvoje potomaka, koji se potom mutiraju i evaluiranju, te se vraćaju u originalnu populaciju na mjestu eliminsanih jedinki.

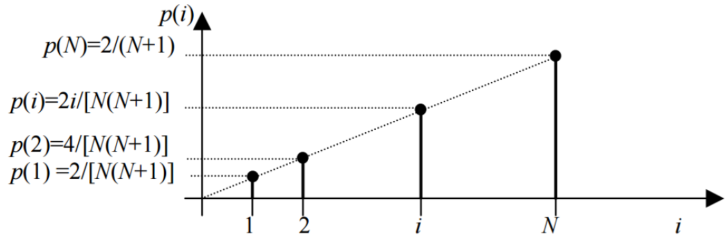


Linearno sortirajuća selekcija

- Rangirajuća selekcija kod koje vjerovatnoća izbora ne zavisi direktno od *fitnessa*-a, već od položaja jedinke u poretku jedinki sortiranih po *fitness*-u. Vjerovatnoća je proporcionalna rangu, odnosno položaju jedinke u tom poretku.

$$p(i) = \frac{i}{\Sigma_{i=1}^{N} i} = \frac{2i}{N(N+1)}$$

- Najbolja jedinka ima indeks N, dok najgora 1.



Selekcija najboljih

- Bira se unaprijed zadati broj najboljih jedinki.
- Tri vrste:
  - $(\mu + \lambda)$ selekcija
    - Slučajno se bira μ roditelja i ukrštanjem se stvara λ potomaka.
    - Iz skupa roditelja i potomaka se potom bira najboljih μ jedinki za sledeću generaciju.
    - Postupak se ponavlja sve dok se ne popuni nova generacija sa N jedinki, odnosno $\frac{N}{\mu}$ puta.
  - (μ, λ) selekcija
    - Slučajno se bira μ roditelja i ukrštanjem se stvara λ potomaka.
    - Potomaka je više od roditelja ($\lambda \geq \mu$) i najboljih μ potomaka se bira za narednu generaciju.
  - *Krnja selekcija*
    - Bira se n najboljih jedinki i kopira ih $\frac{N}{n}$ puta u bazen za ukrštanje.

Seksualna selekcija

- Koristi se u slučaju kada je populacija podijeljena na dvije populacije - mužjake i ženke. Svrha ovakve podjele jeste definicija pravila ko sa kime može da se ukršta i pod kojim uslovima.
- Jedan od načina da se sprovede multikriterijumska optimizacija jer može da se računa jedna pogodnost za pravo na ukrštanje a druga za vjerovatnoću da li će doći do ukrštanja (oponaša prirodan proces).
- Postoji i *ukrštanje sa maticom* koje oponaša razmnožavanje pčela kod kojih jedna jedinka sa najboljim *fitness*-om (*matica*) učestvuje u svim ukrštanjima sa ostalim jedinkama.

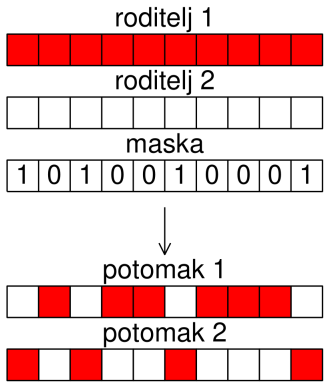
Ukrštanje

- Da bi se iskoristio genetski potencijal (*fitness*), koristi se ukštanje u cilju generisanja novih hromozoma, odnosno nove generacije.
- Nova generacija bi trebalo da očuva dobre osobine prethodne generacije.
- Svaka grupa roditelja neće obavezno proizvesti potomstvo. Ukrštanje se primjenjuje na odabrane parove roditelja čija je vjerovatnoća jednaka *stepenu ukrštanja* $p_c \in [0, 1]$.
- Pseudokod ukrštanja

```
1 Odredi broj k koji pripada intervalu [0, 1]
2 Ako je k > p_c, tada nema ukrštanja; u suprotnom, ima
3 a = C1 (Prva jedinka); b = C2 (Druga jedinka)
4 Izračunati masku m (određuje koji biti roditelja će biti zamijenjeni)
5 za i = 1 do N, ako je m_i = 1, izvrši zamjenu genetskog materijala
      a_i = C2_i
      b_i = C1_i
6 Vrati potomke a i b
```

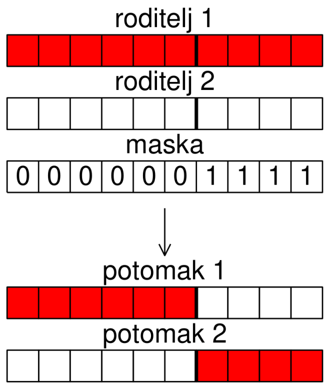
- Nekoliko vrsta:
  - *Uniformno ukrštanje*
    - Na slučajan način se kreira maska dužine N za svaki par jedinki izabranih za reprodukciju.
    - Određuje se slučajan broj k koji je u intervalu $[0, 1]$. Ako je on veći od p_x (što je vjerovatnoća ukrštanja na svakoj poziciji u hromozomu), tada je $m_i = 1$.

Na slici je grafički prikazan princip **uniformnog ukrštanja**.



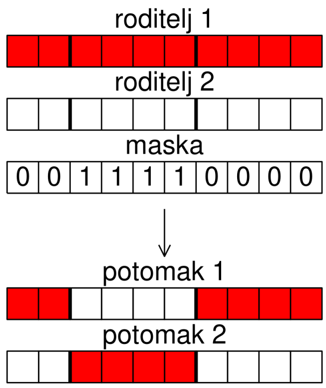
- *Ukrštanje u jednoj tački*
  - Na slučajan način se bira pozicija jednog bita.
  - Na sve bitove prije izabranog stavljaju se geni jednog roditelja, dok na sve bitove poslije izabranog se stavljaju geni drugog roditelja.

Na slici je grafički prikazan princip **ukrštanja u jednoj tački**.



- *Ukrštanje u dvije tačke*
  - Na slučajan način se biraju dvije pozicije.
  - Slično kao i kod ukrštanja u jednoj tački, samo što se još više rekombinuju geni.

Na slici je grafički prikazan princip **ukrštanja u dve tačke**.



- Ukrštanje u jednoj ili dvije tačke je preporučljivo pri većim populacijama jer veća populacija ima i veću raznolikost šema. Sa druge strane, uniformno ukrštanje se koristi pri manjim populacijama kako populacija ne bi postala homogena. Postoje i verzije ukrštanja sa N tačaka.
- Ukrštanje permutacija
  - Problem što se permutacije sastoje od niza indeksa i kod običnog ukrštanja može da se desi da se kod potomka jave dva ista indeksa na različitim mjestima, što nije validna permutacija.
  - *Partially Matched Crossover* (*PMX*)

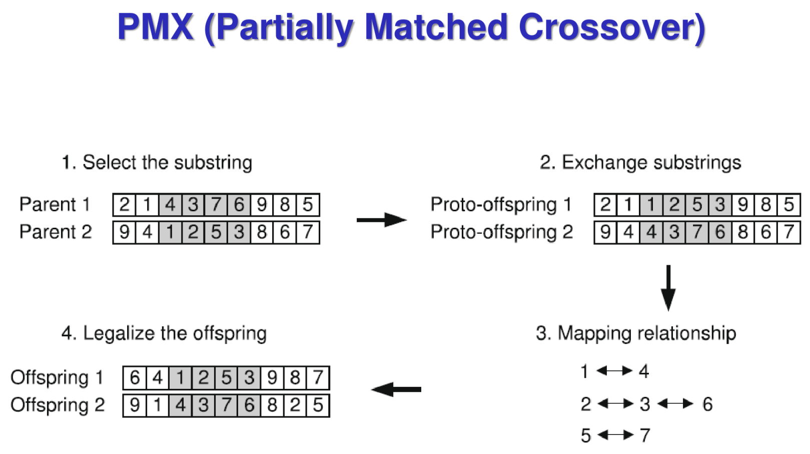
PMX (Partially Matched Crossover)

- Prva varijanta ukrštanja permutacija
- Prvi korak je obično ukrštanje u *N* tačkaka (2 tačke na primeru)

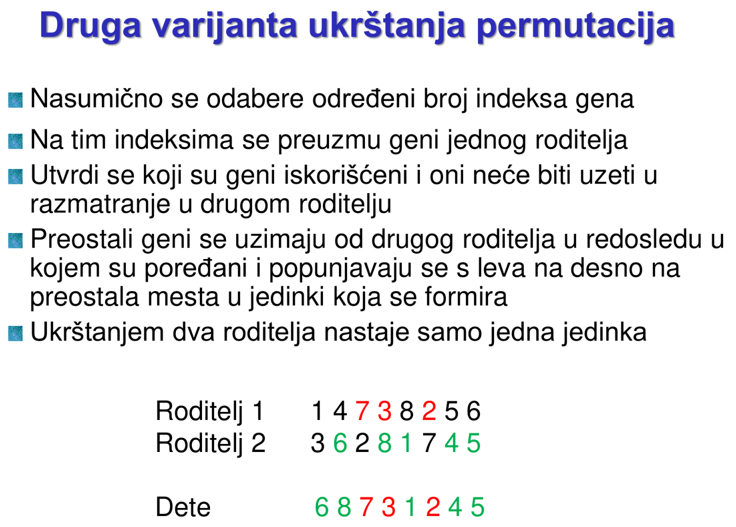
Roditelj 1	1 4 7 | 3 8 2 | 5 6
Roditelj 2	3 6 2 | 8 1 7 | 4 5
Potomak 1	1 4 7 | 8 1 7 | 5 6
Potomak 2	3 6 2 | 3 8 2 | 4 5

- U drugom koraku se proveri da li se neki indeksi ponavljaju kod deteta (dovoljno proveriti kod jednog deteta), pa se na tim mestima izvrši dodatno ukrštanje.

Dete 1	3 4 2 | 8 1 7 | 5 6
Dete 2	1 6 7 | 3 8 2 | 4 5



- *Druga varijanta ukrštanja permutacija*

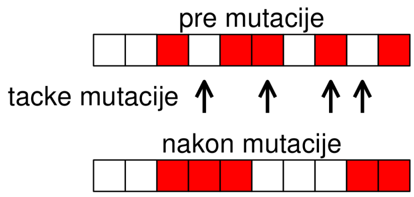


Mutacija

- Vrši se u cilju unošenja **novog** genetskog materijala u već postojeće jedinke (hromozome).
- Primjenjuje se kada populacija ne napreduje ili stagnira, kada ne sadrži kodirane sve informacije neophodne za rješavanje konkretnog problema.
- Mutacija se vrši određenom vjerovatnoćom p_m koji se naziva **stepen mutacije**. On se obično drži na niskom nivou kako se dobri hromozomi ne bi izgubili (ako je visok, GA postaje algoritam slučajnog pretraživanja).
- Operator mutacije može da spriječi populaciju da konvergira ka nekom lokalnom minimumu.
- Kod binarnog kodiranja vrši se negacija, kod realnog sabiranje sa malim slučajno generisanim brojem. Kod realnog kodiranja se može koristiti veći stepen mutacije jer se kod njih tada povećava nivo moguće pretrage prostora rješenja i ne utiče se negativno na karakteristike konvergencije.
- Vrste:

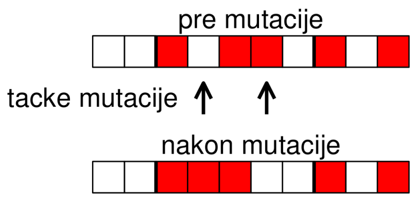
- *Slučajna mutacija*
  - Pozicija bita koji se mutira bira se na slučajan način, nakon čega se vrši operacija negacije nad vrijednošću tog bita.

Na slici je grafički prikazan princip **slučajne mutacije**.



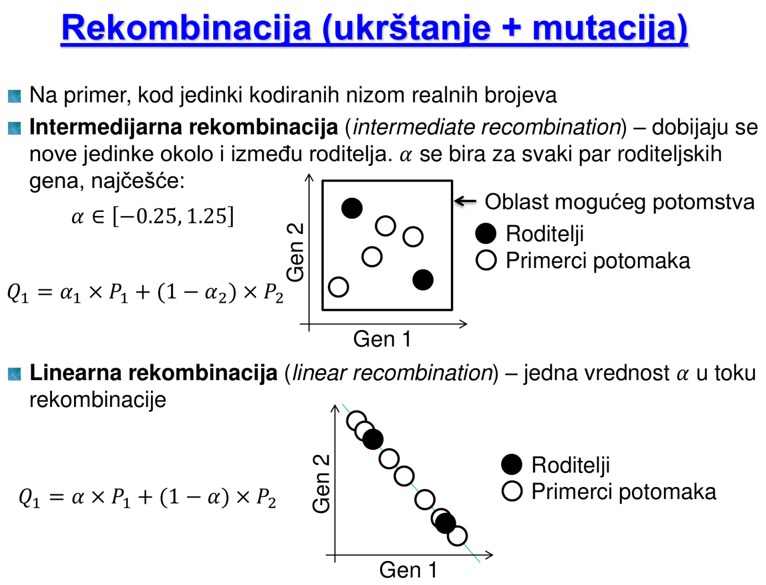
- *Uređena mutacija*
  - Na slučajan način se bira pozicija dva bita.
  - Nakon toga se vrši mutacija samo onih bitova koji se nalaze između prethodno određenih granica.

Na slici je grafički prikazan princip **uređene mutacije**.



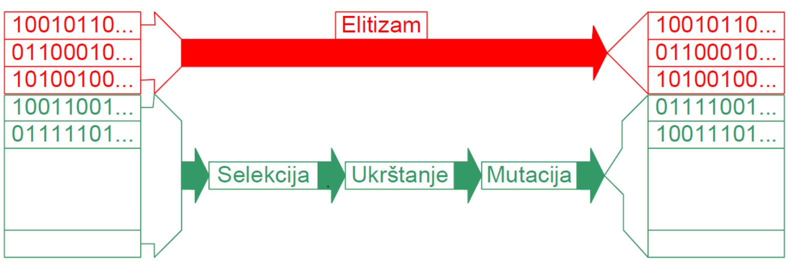
- Postoji i *mutacija pregrupisanja*, gdje se vrši zamjena gena. Ona se obično pojavljuje kod permutacionih problema.
- Mutacijom je moguće pretražiti čitav parametarski prostor i upravo zato je koristan mehanizam za izbjegavanje lokalnih maksimuma.
  - Ako cijela populacija iskonvergira u nekom od lokalnih optimuma, jedino rješenje je mutirati bar jednu kako bi se nastavila pretraga.
- Varijacije operatora mutacije
  - Naklonjenost mutacije jedinkama sa manje boljim *fitness*-om dok se istovremeno čuvaju jedinke sa dobrim.

    - Parametrizacija mutacije tako da frekvencija mutacije opada sa konvergencijom populacije.
    - Dodavanje novih jedinki u populaciju tako što se lošije jedinke mijenjaju onim generisanim na slučajan način, čime se smanjuje stepen mutacije.
- Rekombinacija = ukrštanje + mutacija



Elitizam

- Kako je GA vještački sistem evolucije jedne populacije, u njegovom okviru moguće je izvršiti izbor određenog broja najboljih jedinki i direktno ih prenijeti u narednu generaciju. Takva operacija naziva se **elitizam**.
- Koliko jedinki birati zavisi od veličine populacije ali nema smisla prenijeti previše jedinki.
    - Koristiti elitizam da bi se sačuvala najbolja jedinka iz populacije.
    - Elitizam zahtjeva pronalaženje najbolje jedinke što može biti skupa i vremenski dugačka operacija ako se radi o velikim populacijama.



Kriterijum zaustavljanja

- Zadati broj iteracija
- Dostignuće željene vrijednosti kriterijumske funkcije
- Nenapredovanje algoritma - ako se *fitness* najbolje jedinke nije značajno promijenio kroz nekoliko generacija.

    - U ovakvom slučaju korisno je, recimo, povećati stepen mutacije jer se tako možda može napredovati.
- Vrijeme, odnosno dužina trajanja optimizacije.

U korist i protiv GA

U korist GA	Protiv GA
Kriterijumska funkcija je potpuno proizvoljna, nema posebnih zahteva kao što su neprekidnost, derivabilnost i sl.	Teško je definisati *dobru* funkciju prilagođenosti, potrebno je prilagoditi GA zadatim ograničenjima.
Primenjiv na veliki broj problema	Često je potrebno prilagoditi problem algoritmu.
Mogućnosti nadogradnje i povećanja efikasnisti (puno stepeni slobode)	Teško je postaviti dobre parametre (veliki uticaj parametara na efikasnost)
Jednostavnim ponavljanjem se može povećati pouzdanost. Rezultat je skup rešenja, a ne jedno rešenje.	Konvergencija znatno sporija od ostalih numeričkih metoda.
Rešava sve optimizacione probleme (realne, binarne, diskretne, znakovne, višemodalne, višedimenzione, ...)	Potrebno je prilagoditi genetske operatore kodiranju
Jednostavnost ideje i dostupnost programske podrške	Spor zbog izvođenja velikog broja računskih operacija. Traži se velika procesorska snaga.

Primeri praktičnih primena

- Permutacioni problemi
    - Primena genetskog algoritma u rešavanju bin-paking problema
- Optimizacija parametara FSO
    - Realizacija sistema za podršku u odlučivanju pri upravljanju brodskom prevodnicom zasnovanog na rasplinutoj (fuzzy) logici
- Optimalne putanje na terenu
    - Pronalaženje optimalne putanje za ski stazu upotrebom genetskih algoritama
- Podešavanje parametara regulatora

Podešavanje parametara regulatora

- Podešavanje P, I i D dejstva kod PID regulatora
- Podešavanje parametara Fuzzy regulatora
    - pojačanja regulatora na ulazima i izlazima regulatora
    - broj funkcija pripadnosti
    - tip funkcija pripadnosti
    - parametri funkcija pripadnosti
    - tabela pravila
    - tip regulatora (Mamdani, Takagi-Sugeno, ...), tip implikacije, agregacije, defazifikacije, ...
- Kriterijumska funkcija se obično računa iz odziva sistema
    - vreme uspona
    - vreme smirenja
    - preskok
    - greška u ustaljenom stanju
    - indeks performanse (IAE, ISE, ITAE, ITSE)
    - ltd.
- Često je prilagođenost suma više kriterijuma ponderisana težinama.