Primenjeni algoritmi – pitanja i odgovori

1. Evolutivni algoritmi.
2. Genetski algoritam. Osnovni pojmovi. Koraci.
3. Genetski algoritam. Kodiranje. Tipovi kodiranja.
4. Genetski algoritam. Početna populacija.
5. Genetski algoritam. Karakteristike i vrste selekcije.
6. Genetski algoritam. Ukrštanje. Tipovi i izbor.
7. Genetski algoritam. Mutacija. Vrste mutacije.
8. Genetski algoritam. Mutacija. Značaj. Varijacije operatora.
9. Genetski algoritam. Rekombinacija. Vrste.
10. Genetski algoritam. Elitizam. Kriterijumi zaustavljanja.
11. Izbor modela. Unakrsna validacija. Vrste.
12. Izbor svojstava. Algoritmi.
13. PSO algoritam. Osnovni principi.
14. PSO algoritam. Modifikacije PSO algoritma.
15. Diskretni PSO algoritam.
16. Distribuirani PSO algoritam.
17. Evolutivni PSO algoritam.
18. ACO algoritam. Osnovni principi.
19. ACO algoritam. Promena količine feromona – strategije.
20. Mašinsko učenje. Definicija i osnovna podela.
21. Merenje rastojanja između podataka.
22. Skaliranje podataka.
23. K-means klasterizacija. Izbor optimalnog k.
24. Linearna regresija.
25. Vrste gradijentnog algoritma.
26. Lokalno ponderisana linearna regresija.
27. Logistička regresija. Postupak određivanja parametara.
28. K najbližih suseda. Izbor optimalnog k.
29. Kriptografija (primena, procesi, šifrovanje i bezbednosne pretnje).
30. Osnovni algoritmi kriptografije (podela, primitivni i napredni algoritmi).
31. RSA algoritam (principi, uključeni algoritmi).
32. Lanac blokova (*Blockchain*).
33. Dokazi bez otkrivanja informacija (*Zero Knowledge proof*).
34. Problem vizantijskih generala.

1. Evolutivni algoritmi.

Evolutivni algoritmi su algoritmi u čiju proceduru je ugrađena zakonitost ponašanja nekog sistema u prirodi.

Genetski algoritmi(**GA**) su preuzeli zakonitosti koje počivaju na genetskom ukrštanju hromozoma čoveka.

Ponašanja grupa životinja su ugrađena u veliki broj uspešnih optimizacionih algoritama.

- ponašanje jata ptica – **PSO** (engl. *Particle Swarm Optimization*)
- ponašanje kolonije mrava – **ACO** (engl. *Ant Colony Optimization*)
- ponašanje roja pčela – **BCO** (engl. *Bee Colony Optimization*) ili **ABC** (engl. *Artificial Bee Colony*)

Predstavljanje rešenja:

- Jedno rešenje se predstavlja preko jedinke
- Inicijalno –definiše se skup jedinki, najčešće generisanih na slučajan način
- Vrši se pretraga prostora dopustivih rešenja u cilju pronalaženje globalnog optimuma

2. Genetski algoritam. Osnovni pojmovi. Koraci.

Osnovni pojmovi:

- ***Jedinka (hromozom u GA)*** – potencijalno rešenje problema, implementirana kao struktura podataka; niz promenljivih
- ***Gen*** – deo jedinke koji se odnosi na jedan brojčani podatak; promenljiva
- ***Populacija*** – skup više jedinki
- ***Generacija*** – skup svih jedinki u jednom trenutku; iteracija
- ***Prilagođenost (Fitness)*** – vrednost kriterijuma optimalnosti
- ***Operatori*** – akcije nad jedinkama:
  - Selekcija
  - Ukrštanje
  - Mutacija
  - Elitizam

Koraci u GA:

- Inicijalizacija–kreiranje inicijalne populacije
- Izračunavanje fitnes funkcije za svaku jedinku u populaciji
- Ponavljati dok se uslovi ne zadovolje
  - Selekcija
  - Ukrštanje
  - Mutacija
  - Izračunavanje fitnes funkcije za svaku jedinku u populaciji
- Izbor jedinke sa najvećom fitnes funkcijom

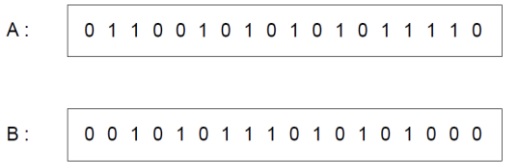
3. Genetski algoritam. Kodiranje. Tipovi kodiranja.

Kodiranje:

- Ovom transformacijom se tačke parametarskog prostora predstavljaju nizovima (stringovima) ili matricama.
- Šema kodiranja – način transformacije specifičnog znanja vezanog za određeni problem u okvire genetskog algoritma.
- Direktno (ključno) utiče na uspešnost genetskog algoritma.
    - **Binarno** I **sivo** (Gray) kodiranje;
    - **Realni brojevi**, diskretne vrednosti (najčešće nizovi ali mogu biti N-dimenzionalne matrice u opštem slučaju, npr. Kod kodiranja koordinata putanje)
    - **Redosledno kodiranje** –**Permutacioni** problem –svaku permutaciju treba kodirati na jedinstven način (bin-packing problem, problem trgovačkog putnika, …)
    - **Stablo**

Binarno kodiranje:

- Jedinka (hromozom) je predstavljena stringom binarnih vrednosti.



- *Prednosti binarnog kodiranja:*
    - Brzina binarnih operacija – brži GA.
    - Svaki optimizacioni problem može se binarno predstaviti.
- *Nedostaci binarnog kodiranja:*
    - *Napor pretvaranja u binarnu formu.*
    - *Tačnost zavisi od binarne predstave.*

Kodiranje realnim brojevima:

- Najpogodnija za optimizaciju u neprekidom prostoru pretraživanja.
- Direktna predstava vrednosti parametara.
- Izbegavaju se koraci kodiranja i dekodiranja.

- *Korak 1 – izabrati preciznost.*
  - Za kontinualnu promenljivu $x \in [X_D, X_G]$ ako je ε potrebnu preciznost tada je string n jednak:

$$n = \log_2\left(\frac{X_G - X_D}{\varepsilon}\right)$$

tj.

$$\varepsilon = \left(\frac{X_G - X_D}{2^n}\right)$$

$\varepsilon \in [0, 1]$ – preciznost

- *Korak 2 – dobijanje binarne reprezentacije.*
  - Za određenu dužinu binarnog niza *n* i za postignutu tačnost, može se odrediti realne vrednosti X do njene ekvivalentne binarno dekodirane vrednosti $\mathbf{X_B}$, koja je data sa:

$$X = X_D + \frac{X_G - X_D}{2^n - 1} X_B$$

gde je $\mathbf{X_B}$ - dekodirana vrednost binarnog niza,

$X_D \rightarrow 000\ldots0$

$X_G \rightarrow 111\ldots1$

su dekodirane vrednosti binarnog prikaza donje i gornje vrednosti.

Redosledno kodiranje:

- *Primer: Problem trgovačkog putnika*
  - Sve gradove samo jednom posetiti.
  - Definisani su troškovi posete gradu iz drugog grada.
  - Početni I krajnji grad su isti.

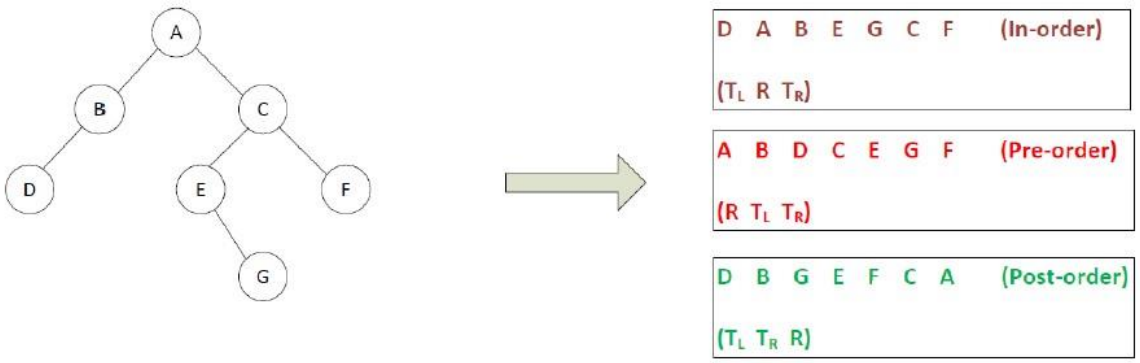
Problem trgovačkog putnika – kodiranje:

- **Funkcija cilja:** Pronaći putanju posete uz minimalne troškovima.
- **Ograničenja:** Svi gradovi moraju biti posećeni samo jednom (osim početnog grada).
- **Parametri:** Euklidova udaljenost se može uzeti kao mera troškova, u suprotnom, ako je izričito navedeno.

- Potrebno je pronaći najbolju putanju od n! mogućih putenja.

Stablo:

- Binarno stablo
- Tri načina predstavljanja



4. Genetski algoritam. Početna populacija.

Veličina populacije:

- Veća populacija.
  - više genetskog materijala,
  - veće šanse za dolaženje do optimalnog rešenja,
  - sporiji algoritam algoritma.
- Manja populacija.
  - Prednost manje populacije je brzina.
- Promenljiva.
- Promena veličina kroz generacije.

Načini generisanja početne populacije:

- Slučajno (najčešće).
- Uniformno raspoređene jedinke u prostoru rešenja.
  - Pokriva se prostor pretrage.
  - Mana – veštački napravljena.
- U početnu populaciju treba uključiti rešenja dobijena drugim optimizacionim algoritmima, ako su dostupna.

5. Genetski algoritam. Karakteristike i vrste selekcije.

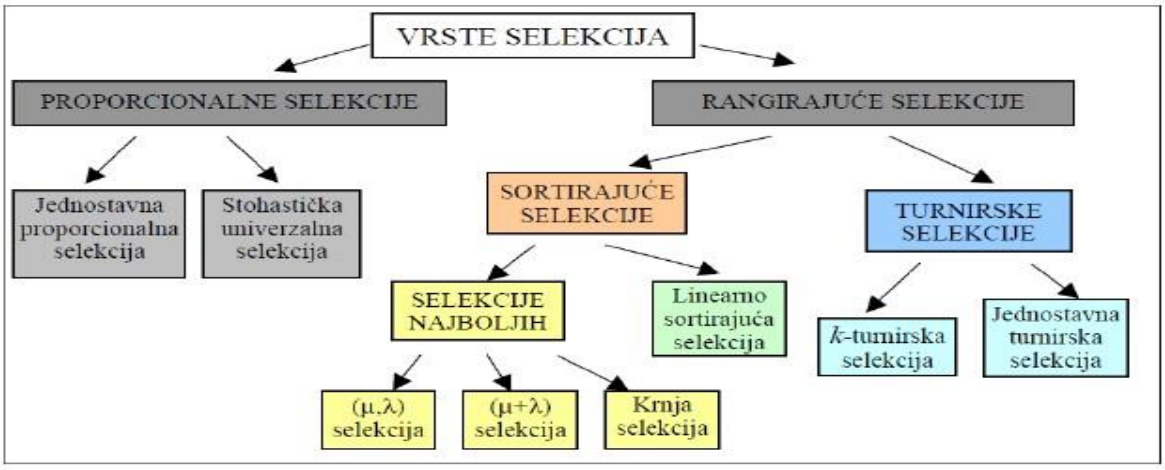
Selekcija:

- Veća verovatnoća boljih jedinki za reprodukciju.
- Podela selekcija prema načinu prenošenja boljih jedinki u sledeću generaciju:
  - generativna;
  - eliminaciona.
- Svaka jedinka (sem najboljih) postoji samo u jednoj generaciji.

Karakteristike selekcije:

- Brzina.
- Pritisak.
  - odnos verovatnoća preživljavanja dobrih I loših jedinki.
  - veći – kada se velikom verovatnoćom prenose "bolje" jedinke – prerana konvergencija.
- Raznolikost – omogućavanje "lošijim" jedinkama da budu izabrane.

Vrste selekcija:



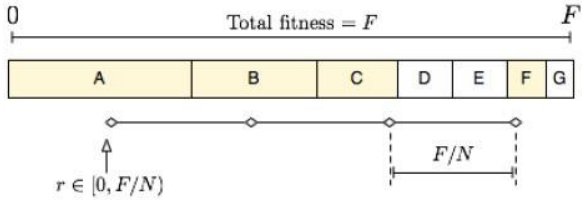
- ***Rulet selekcija:***
  - Jednostavna proporcionalna selekcija ili selekcija ruletnog točka.
  - Verovatnoća da će jedinka biti selektovana je proporcionalna njenom stepenu prilagođenosti

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

gde je N broj jedinki u populaciji

$$F = \sum_{j=1}^{N} f_j$$

- ***Stohastička univerzalna selekcija:***
  - Za izbor N roditelja vrše se sledeći koraci:
    - Generiše jedan slučajan broj r u intervalu [0, F/N].
    - 1. Roditelj se bira na poziciji r.
    - Svi ostali roditelji su na pozicijama r + i * F/N, i = 1,….,N –1
  - Uniformno raspoređeni roditelji.



  - Slabijim jedinkama se daje šansa da budu izabrane (smanjuje se nefer priroda ruleta).

- ***K – turnirska selekcija:***
  - Sprovodi se više “turnira” među nekoliko (K) jedinki izabranih slučajno iz populacije.
  - Na turniru se od K slučajno odabranih jedinki pronalazi najbolja (pobednik turnira – jedinka sa najboljom pogodnošću).
  - N turnira daje N roditelja.
  - Menjanjem veličine turnira se lako podešava pritisak selekcije.
    - Što je K veće, slabiji pojedinci imaju manje šanse da budu izabrani.
  - K – 1 najslabijih jedinki nemaju nikakve šanse (jedan od načina da najlošije jedinke ne budu nikad odabrane)
  - Ako je K = 1, selekcija je ekvivalentna slučajnom izboru.
  - Prednosti:
    - Brzina – pogodnost se računa samo za odabrane jedinke I nema sortiranja cele populacije.
    - Radi na paralelnim arhitekturama.
    - Pritisak selekcije se lako podešava.
- ***Jednostavna turnirska selekcija:***
  - Specifičan oblik binarne turnirske selekcije (K = 2)
  - Postupak:
    - Slučajnim postupkom se bira dva para jedinki.
    - U svakom paru se lošije jedinke eliminišu.
    - Ukrštanjem boljih jedinki se generiše dvoje potomaka, koji potom mutiraju i evaluiraju.
    - Taj par novostvorenih jedinki nadomešćuje eliminisane jedinke.
  - Na taj način GA se jednostavnom turnirskom selekcijom u istom koraku obalja i selekciju i reprodukciju.

- ***Linearno sortirajuća selekcija:***
  - Verovatnoća izbora zavisi od položaja jedinke u poretku jedinki sortiranih po fitnesu.
  - Kod linearno sortirajuće selekcije verovatnoća je proporcionalna rangu, odnosno poziciji jedinke u poretku sortiranom po pogodnosti.

$$p(i) = \frac{i}{\sum_{i=1}^{N} i} = \frac{2i}{N(N+1)}, \qquad i = 1, \dots, N$$

    gde najbolja jedinka ima indeks N, a nojgora 1.

- ***Selekcija najboljih:***
  - Bira se unapred zadati broj najboljih jedinki.
  - Tri vrste:
    - (μ + λ) selekcija
    - (μ, λ) selekcija
    - Krnja selekcija

    μ - veličina populacije roditelja

    λ – veličina populacije potomaka

  - Razlikuju se prema skupu jedinki iz kojeg se biraju najbolje.
    Mogu se birati iz skupa samo roditelja, roditelja i potomaka ili samo potomaka.
- ***Selekcija (μ + λ):***
  - Postupak:
    - Slučajno se bira μ roditelja.
    - Njihovim ukrštanjem se stvara λ potomaka.
    - Iz skupa roditelja i potomaka bira najboljih μ jedinki za sledeću generaciju.
    - Postupak se ponavlja sve dok se ne popuni nova generacija sa N novih jedinki, odnosno N/ μ puta.

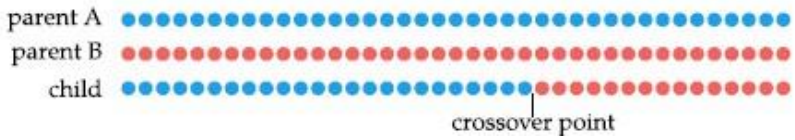
- ***Selekcija (μ, λ):***
    - Postupak:
        - Slučajno se bira μ roditelja.
        - Njihovim ukrštanjem se stvara λ potomaka.
        - Potomaka je više od roditelja (λ ≥ μ) i μ najboljih potomaka se bira za narednu generaciju.
    - Uključena je rekombinacija (selekcija + ukrštanje)

Krnja selekcija bira n najboljih jedinki I kopira ih N/n puta u bazen za ukrštanje.

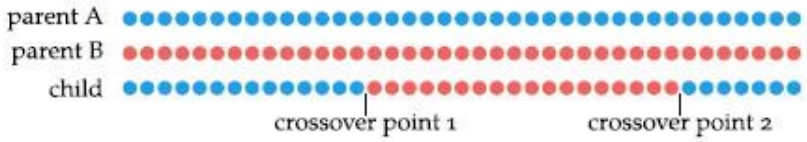
6. Genetski algoritam. Ukrštanje. Tipovi i izbor.

Ukrštanje:

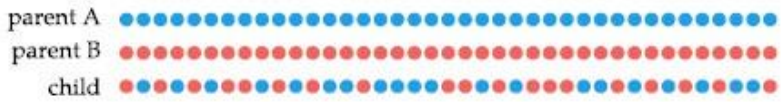
- Potomak se dobija kombinovanjem hromozoma roditelja.
- **Ukrštanje u jednoj tački**: kombinacija roditelja od jedne tačke.



- **Ukrštanje u dve tačke**(analogno u N - tačaka).



- **Uniformno ukrštanje:** svaki bit ima 50% šanse za ukrštanje.



Izbor tipa ukrštanja:

- Veće populacije (veća raznolikost) – ukrštanje u jednoj ili dve tačke.
- Manje populacije – uniformno ukrštanje.
- Ograničeno ukrštanje u N tačaka, npr. kod rutiranja putanja (samo neke jedinke mogu da se ukrštaju i to samo na nekim mestima).

Karakteristike ukrštanja realnih vrednosti:

Prednosti:

- Jednostavni i brzi proračuni.
- Moguće je generisati veliki broj dece od dva roditelja.
- Lako se upravlja širokim spektrom varijacija.

Ograničenja:

- Potrebe za generisanjem konstanti (α_i *i* β_i)
- Ako α_i *i* β_i nisu dobri, rešenje se zaglavi u lokalnom minimumu.

7. Genetski algoritam. Mutacija. Vrste mutacije.

Mutacija:

- Negacija kod binarnog kodiranja (0 u 1 i obrnuto).
- Realne vrednosti (sabiranje sa malim slučajno generisanim brojem).
- Mutacija pregrupisavanja –zamena gena (npr. kod permutacionih problema gde se ne smeju pojavljivati novi geni i svaki gen može da se pojavi tačno jednom).
- Kod realno-kodiranih jedinki se može koristiti veći stepen mutacije jer se kod njih, za razliku od binarno-kodiranih, povećava nivo moguće pretrage prostora rešenja, a da se ne utiče negativno na karakteristike konvergencije.

Slučajna mutacija – mutacija kod kodiranja realnim vrednostima:

Koristi se pravilo: $P_{mutirano} = P_{original} + (r - 0.5) * \Delta$

gde je $r \in [0, 1]$ slučajan broj, a Δ je maksimalna vrednost promene definisana od strane korisnika.

Polinomska mutacija – mutacija kod kodiranja realnim vrednostima:

Zasnovana na polinomskoj raspodeli, sastoji se iz sledećih koraka:

1. Generisati slučajan broj r izmedju 0 i 1
2. Izračunati faktor promene δ kao

$$\delta = \begin{cases} (2r)^{\frac{1}{q+1}} - 1, & r < 0.5 \\ 1 - [2(1-r)]^{\frac{1}{q+1}}, & r \geq 0.5 \end{cases}$$

gde je q korisnički definisan eksponent (pozitivan ili negativan).

3. Mutirano rešenje se dobija kao

$$P_{mutirano} = P_{original} + \delta * \Delta$$

gde je maksimalna vrednost promene između originalne i mutirane vrednosti.

8. Genetski algoritam. Mutacija. Značaj. Varijacije operatora.

Zašto je mutacija važna?

- Izbegavanje lokalnih optimuma.
- Dovoljno je da jedna jedinka (nastala mutacijom) bzde bolja od ostalih, pa da se u nekoliko narednih generacija, sve jedinke "presele" u prostor gde se nalazi bolje rešenje.

Varijacije operatora mutacije

Postoje mnoge varijacije operatora mutacije. Npr:

- **Naklonjenost mutacije jedinkama sa manjim fitnesom** da bi se proširilo polje pretrage, a istovremeno očuvale informacije kod onih sa većim fitnesom.
- **Parametrizacija mutacije** tako da frekvencija mutacije opada sa konvergencijom populacije.

Dodavanje novih jedinki u populaciju

- U toku smanjenja stepena mutacije kroz generacije može da bude zamena najlošijih (ili najstarijih) jedinki novim jedinkama generisanim na slučajan način.
    - Tada broj jedinki koj se zamenjunje opada kroz generacije i time se postepeno prelazi iz slučajne pretrage u optimizaciju genetskim algoritmom.

9. Genetski algoritam. Rekombinacija. Vrste.

Rekombinacija(ukrštanje + mutacija):

Npr. kod jedinki kodiranih nizom realnih brojeva

- **Intermedijarna rekombinacija** – dobijaju se nove jedinke okolo i između roditelja. α se bira za svaki par roditeljskih gena, najčeće:
  $\alpha \in [-0.25, 1.25]$
  $Q_1 = \alpha_1 * P_1 + (1 - \alpha_2) * P_2$
- **Linearna rekombinacija** – jedna vrednost α u toku rekombinacije.
  $Q_1 = \alpha * P_1 + (1 - \alpha) * P_2$

10. Genetski algoritam. Elitizam. Kriterijumi zaustavljanja.

Elitizam:

Obezbeđuje monotono nerastuću funkciju prilagođenosti najbolje jedinke kroz generacije.

Prednost elitizma:

Jedini način da se najbolja jedinka iz populacije sačuva.

Mana elitizma:

Pretraživanje ili sortiranje zahteva procesorsko vreme zbog čega se može znatno usporiti genetski algoritam.

Kriterijumi zaustavljanja:

- Zadati broj iteracija.
- Dostignuta željena vrednost kriterijumske funkcije.
- Vreme (dužina trajanja optimizacije)
- "Zaglavljivanje" algoritma.
    - ako se prilagođenost najbolje jedinke nije značajno promenila(function tolerance) u zadatom broju generacija.
    - može se pokušati sa promenom parametara, npr. povećati stepen mutacije.

11. Izbor modela. Unakrsna validacija. Vrste.

Unakrsna validacija (Cross validation):

- Trenirati svaki model M_i na obučavajućem skupu S – dobiju se hipoteze h_i
- Izabrati hipotezu sa najmanjom greškom.
    - Ovo ne radi… Ako se izabere polinom velikog reda on će bolje fitovati podatke iz obučavajućeg skupa S i dati manju obučavajuću grešku. Ali nije dobra zato što daje veliku varijansu.
- Algoritmi validacije:
    - Jednostavna unakrsna validacija
    - K-tostruka validacija
    - Validacija jednostruke eliminacije

Jednostavna unakrsna validacija

1. Na slučajan način se podeli skup S na S_{train} (npr. 70% podataka) – obučavajući skup i S_{cv} (preostalih 30%) - validacioni skup.
2. Trenira se svaki model na skupu S_{train} i dobijaju hipoteze h_i
3. Bira se hipoteza h_i sa najmanjom greškom $\varepsilon_{Scv}(h_i)$ na validacionom skupu $S_{cv.}$

Nedostatak: “Gubitak” oko 30% podataka

Unakrsna validacija izdvajanjem jednog

- Ako je broj primera jako mali uzima k=m
    - Svaki model se obučava na svakom podskupu m-1
    - Testira se na jednom (izostavljenom) primeru
    - Uzima se prosek

12. Izbor svojstava. Algoritmi.

- Ako je broj svojstava d veoma velik d >> n samo je mali skup svojstava relevantan.
- Zadatak: izabrati podskup "značajnih" svojstava.
- Postoji ukupno 2^n podskupova – obimna pretraga.
- **Algoritmi*:***
    - ***Algoritmi omotača:***
        - ***Pretraga unapred –*** dodavanje svojstava
        - ***Pretraga unazad*** – uklanjanje svojstava
    - ***Rangirajući algoritmi:***
        - Uzajamna informacija svojstva i izlaza - korelacija

Algoritmi omotača – pretraga unapred:

1. Polazi se od praznog skupa svojstva $X = \emptyset$
2. Odlika $x_j \notin X$ sa najmanjom greškom se dodaje u skup $X = X \cup x_j$
3. Ako postoji još neizabranih odlika → korak 1
4. U suprotnom vrati podskup X, kraj

Algoritmi omotača – pretraga unazad:

1. Polazi se od skupa svih svojstava X
2. Eliminiše se u svakoj iteraciji jedno svojstvo x_j sa najvećom greškom $X = X/x_j$
3. Ako postoji još neizabranih odlika → korak 1
4. U suprotnom vrati podskup X, kraj

13. PSO algoritam. Osnovni principi.

PSO je zasnovan na imitaciji ponašanja životinjskih skupina, odnosno jedinki u tim skupinama.

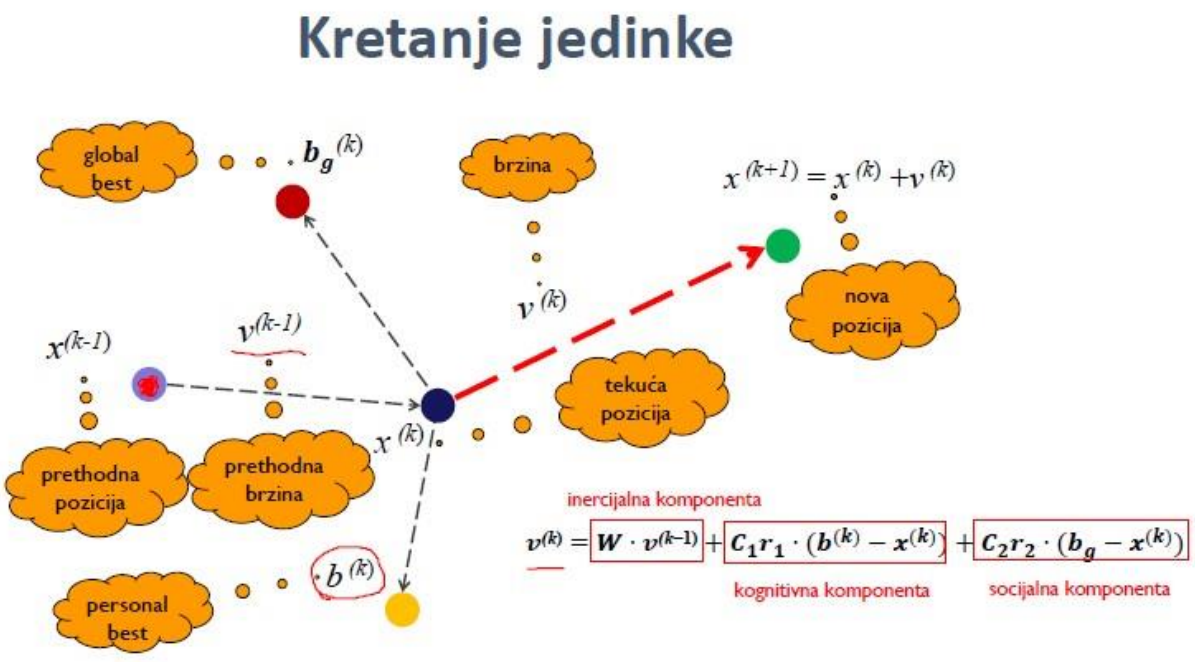
- Jata ptica i riba, rojevi insekata, itd.

PSO je evolutivna, populaciona tehnika.

Posmatra se skup tačaka (potencijalnih rešenja) – populacija ili roj. Tačke se nazivaju jedinkama ili česticama. Jedinka se pomera unutar prostora pretrage. Stoga se za PSO ponekad kaže da je swarm-based optimizaciona tehnika.

Svakoj jedinki se sem tekuće pozicije(koja predstavlja potencijalno rešenje), x, pridružuje i tekuća brzina, v. Svaka jedinka je u stanju da pamti svoju najbolju ikada dostignutu poziciju, b.

Roj, kao celina, pamti najbolju poziciju ikada dostignutu od strane ma koje čestice,b_g.



14. PSO algoritam. Modifikacije PSO algoritma.

Modifikacije – faktor inercije:

- U osnovnoj varijanti PSO algoritma faktor inercije je W=1.
    - Neophodno je veštački ograničiti maksimalnu brzinu svake čestice.

- Performanse algoritma se bitno poboljšavaju (naročito u okolini rešenja) uvođenjem promenljivog inercionog faktora.
    - Preporučuje se da se W smanjuje od 0.9 do 0.4.

Modifikacije – ograničenje brzine:

- Velika brzina kretanja jedinke – značajna promena pozicije i moze doći do divergencije roja tj. tzv. eksplozije roja.
- Uvodi se ograničenje – maksimalno dozvoljena brzina V_M ili po svakom pravcu i $V_{Mi} = k * (b_i - a_i)$,

  b_i , a_i – donja i gornja granica vrednosti po pravcu i, $k \in [0.1, 1]$

15. Diskretni PSO algoritam.

- PSO je u osnovi algoritam za rešavanje kontinuiranih problema.
- Varijante algoritma za diskretne sisteme:
    - Binarni PSO
        - svaka jedinka iz populacije može uzeti binarne vrednosti (0 ili 1).
    - Celobrojni PSO
        - zaokruživanje dostignutog realnog optimuma na najbliži ceo broj.
        - Ili se u okolini dobijenog rešenja traži najbolje celobrojno rešenje

Binarni PSO:

Umesto

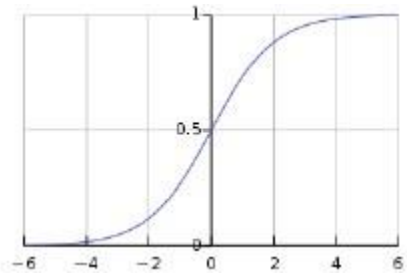
$$x_i^{(k+1)} = x_i^{(k)} + v_i^{(k)}$$

Pozicija čestice se određuje prema

$$x_{i,d}^{(k+1)} = \begin{cases} 1, & rand() < S(v_i^{(k)}) \\ 0 & \end{cases}$$

gde je d indeks bita čestice, a S – sigmoidna funkcija

$$S(x) = \frac{1}{1 - e^{-x}} = \frac{e^x}{e^x + 1}$$



Celobrojni PSO:

- Neophodno je prilagoditi PSO diskretnom problemu.
  - Definisati dozvoljeni prostor za pozicioniranje čestica.

- **Brzina** se zaokružuje na najbliži ceo broj

$$v_i^{(k+1)} = round(v_i^{(k+1)})$$

- Pozicija čestice mora imati celobrojne vrednosti iz nekog intervala
  - npr. za interval (0, m] na kraju se vrši modifikacija:

$$x_i^{(k+1)} = \left|\mathrm{mod}(x_i^{(k+1)}, m)\right|$$

gde je mod(p,q) – ostatak prilokom deljena p sa q

16. Distribuirani PSO algoritam.

Zasniva se na paralelnom pretraživanju prostora dopustivih rešenja.

Ima analogije sa konceptom okoline – globalno se može posmatrati kroz više nivoa pretrage.

Populacija (roj) se sastoji iz više distribuiranih podrojeva.

Princip:

1. PSO nad svakim lokalnim podrojem.
2. Razmena rezultata.
3. Ponavljanje koraka 1. i 2. do završetka algoritma.

Sinhronizacija distribuiranog PSO:

- Period sinhronizacije (T_{sync}),
    - određuje frekvenciju komunikacije između podrojeva.
    - broj iteracija posle kojeg će se ažurirati globalni optimum (bg).
    - kada podrojevi dobiju informaciju o globalnom optimumu – menjaju brzinu.
- Promenom perioda singronizacije se menja I način izvrsavanja algoritma.
- Malo T_{sync} – podrojevi su više međusobno zavisni I efekat paralelizma manje dolazi do izražaja.
    - algoritam je sporiji zbog češće komunikacije između podrojeva.
- Veće T_{sync} - podrojevi su nezavisniji, ali se to može odraziti na kvalitet rešenja.

17. Evolutivni PSO algoritam.

- Ovo je hibridni algoritam koji je podstaknuk idejom da se kreira algoritam koji bi bio kombinacija PSO algoritma i evolutivnih algoritama.
- Specifičnost algoritma: umesto klasične mutacije i rekombinacije korišćeno pravilo kretanja čestica definisano PSO algoritmom.
- Razmatrani parametric su grupisani na:
    - objektne (pozicija H),
    - strateške (težinee W).

Faze EPSO algoritma:

1. Replikacija – svaka čestica se replicira r puta;
2. Mutacija – svaka čestica ima mutirane strateške parameter;
3. Reprodukcija – mutirana čestica stvara potomka na osnovu kretanja čestice zasnovanog na PSO algoritmu;
4. Evaluacija – računanje kriterijumske funkcije za svakog potomka;
5. Selekcija – na osnovu izabrane procedure (slučajnom, turnirskom, proporcionalnom i dr.) najbolje čestice opstaju i prenose se na sledeću generaciju.

Koncepti komunikacije izmedju čestica EPSO algoritma:

- Stohastička zvezda
- Zasniva se na stepenu komunikacije P
- P – verovatnoća uticaja globalno najbolje čestice na pojedinačnu česticu.
- Manje P:
    - poboljšana lokalna pretraga.
    - smanjuje mogućnost “zaglavljivanja ” u lokalnom optimumu.
- Vrednost P nije fiksna i jedinstvena – zavisi od rešavanog problema.

18. ACO algoritam. Osnovni principi.

Mravlji algoritam (Ant Colony Agoritham–ACO) je optimizacioni algoritam koji imitira ponašanje mravlje kolonije.

Jedan od najefikasnijih algoritama za pronalaženje približnog rešenja problema trgovačkih putnika.

Vrste ACO algoritma:

- **AS (Ant System)** – osnovni algoritam
- **EAS (Elitistic ant system)**
  - globalno najbolje rešenje pojačava feromon.
- **As_{rank} (Rank – based ant system)**
  - w najboljih mrava u iteraciji pojačavaju feromon.
- **MMAS (MIN – MAX Ant System)**
- **ACS (Ant Colony System)**

Osnovna verzija ACO algoritma:

- Atraktivnost grane (i,j) - A_{ij}

$$A_{ij} = \tau_{ij}^{\alpha} \cdot \eta_{ij}^{\beta}$$

- Verovatnoća prelaza mrava preko grane i-j - P_{ij}

$$p_{ij} = \frac{A_{ij}}{\sum_{k \in S} A_{ik}}, \qquad \forall j \in S$$

  gde je S skup suseda čvora i

- Izbor grane na osnovi verovatnoća P_{ij} + npr. princip ruletnog točka.
- Mravi ostavljaju trag nakon uspešnog pređenog puta.
- Količina feromona zavisi od vrednosti/kvaliteta nađene putanje.

19. ACO algoritam. Promena količine feromona – strategije.

- Nakon što svi mravi generišu putanje feromoni se koriguju.

1. **Smanjenje – Evaporacija** (isparavanje) feromona se primenjuje po

   $T_{ij} = (1 - p)\, T_{ij}$, $\forall(i,j) \in G$

   gde je $0 < p \leq 1$ – koeficijent isparenja feromona

   - Cilj:
     - Izbegavanje preuranjene konvergencije u lokalnom optimumu.
     - Podsticanje raznovrsnosti (divesifikacije) pretraživanja.
     -

2. **Povećanje – Pojačanje** feromona
   - Strategija pojačavanja feromona zavisi od problema koji ACO rešava.

Strategije osvežavanja feromona:

1. **Online – u svakom koraku**
   - Trag feromona T_{ij} se osvežava od strane svakog mrava u svakom koraku konstrukcije rešenja.
2. **Online – sa zadrškom**
   - Trag feromona T_{ij} se osvežava tek kada mrav generiše kompletno rešenje.
   - Mrav osvežava feromonski trag proporcionalno kvalitetu dobijenog rešenja.
3. **Offline varijante**

Trag feromona T_{ij} se osvežava tek kada svi mravi generišu kompletno rešenje. Ova strategija se najčeće koristi u različitim vidiovima.

a. **Na osnovu kvaliteta rešenja**
   - Trag feromona se osvežava vrednošću koja je proporcionalna najboljem pronađenom rešenju, ili k najboljih rešenja.

b. **Na osnovu rangiranja**
   - Samo mravi koji su našli w najboljih rešenja mogu da osveže trag feromona, u skladu sa rangom kvaliteta rešenja.

c. **Korekcija najgoreg rešenja**
   - Mravi koji generišu najgore rešenje će se smanjiti trag feromona.

d. **Elitizam**
   - Samo mrav koji je našao najbolje rešenje će pojačati feromon da bi usmerio pretragu u tom smeru.

20. Mašinsko učenje. Definicija i osnovna podela.

Mašinsko učenje je oblast veštačke inteligencije koja se bavi izgradnjom računarskih sistema koji uče iz iskustva.

Komponente mašinskog učenja:

- Model;
- Funkcija cilja za procenu dobrog modela;
- Metoda optimizacije za učenje modela koji optimizuje funkciju cilja.

Podela mašinskog učenja:

- Nadgledano učenje (Supervised Learning)
- Nenadgledano učenje (Unsupervised Learning)
- Polunadgledano učenje (Semisupervised Learning)
- Učenje sa podsticajem (Reinforcement Learning)

Nadgledano učenje

- Polazi od skupa parova (vector svojstava(obeležja), atribut = vrednost)
- Cilj : na osnovu ulaznih podataka–parova izvesti pravilo koje predviđa vrednost asociranu sa novim vektorom svojstava. x → y
- **Regresivni** modeli asociraju realan broj sa svakim vektorom svojstava.
- **Klasifikacioni** modeli asociraju jedan simbol iz konačnog skupa sa svakim vektrom svojstava.

- *Algoritmi nadgledanog učenja:*
    - **k-Nearest Neighbors**
    - **Linear Regression**
    - **Logistic Regression**
    - **Support Vector Machines (SVMs)**
    - **Decision Trees and Random Forests**
    - **Neural networks**

Nenadgledano učenje

- Nenadgledano učenje je concept koji koristi neoznačene podatke samo X i nalaze interesantne stvari o njima.
- Najčešće se primenjuje za grupisanje podataka.
    - Primer: Google new web site – izdvaja članke na istu temu od različitih autora.

- *Algoritmi nenadgledanog učenja:*
    - **Grupisanje:**
        - K-Means
        - Support Vector Machine (SVM)
        - HCA
    - **Detekcija anomalija:**
        - One-class SVM
        - Isolation Forest
    - **Vizuelizacija I smanjenje dimenzionalnosti:**
        - Principal Component Analysis (PCA)
        - Karnel PCA
        - Locally Linear Embedding (LLE)
    - **Učenje pravila udruživanja:**
        - Apriori
        - Eclat

Polunadgledano učenje

- Delimično obeleženi podaci + neobeleženi podaci

Učenje sa podsticajem

- Agenti izvode aktivnosti po nekoj strategiji.
- Dobije se “nagrada” ili “kazna”.
- Uči koja je najbolja strategija – pravilo sa najviše nagrada.

21. Merenje rastojanja između podataka.

- **Euklidsko rastojanje** – Pitagorina teorma.

$$d(A,B) = \sqrt{\sum_{i=1}^{n}(a_i - b_i)^2}$$

- **Menhetn rastojanje** – suma apsolutnih vrednosti razlika ulaza u vektorima.

$$Manhattan = \sum_{i=1}^{n} |x_i - y_i|$$

- **Čebiševo rastojanje** – maksimum apsolutnih vrednosti razlika između svih elemenata vektora.

$$Chebyshev = max\left(|x_i - y_i|\right)$$

- **Sličnost kosinusa** – vektori usmereni u istom smeru su slični (određuje se kosinus između dva vektora, tj. tačkasti proizvod / dve dužine)

$$similarity(a,b) = \frac{a \,.\, b}{\|a\| \,.\, \|b\|}$$

22. Skaliranje podataka.

- Skaliranje podataka je prvi korak u cilju pravilnog poređenja svojstava kod merenja rastojanja između tačaka.
- Npr. ako se klasifikuju ljudi prema plati i broju kućnih ljubimaca, vrednosti za platu su preko 30000, a vrednosti za broj kućnih ljubimaca je <10.
- Jedan od načina je da se po svakom svojstvu u svim podacima odrede srednja vrednost i standardna devijacija i da se skaliraju vrednosti tog svojstva tako da imaju srednju vrednost 0 i standardna devijacija 1.
- Drugi način je da je minimalna vrednost 0, a najviša 1.

23. K-means klasterizacija. Izbor optimalnog k.

- Ovo je nenadgledana tehnika učenja.
- Postoji veliki broj tačaka predstavljenih vektorima (elementi vektora su atributi) koje nisu klasifikovane ili označene. **Cilj je pametno grupisati tačke.**
- Svaka grupa je asocirana svojim **centoridom**, tj. **težistem.**

Primena K-means klasterizacije:

- Grupisanje neobeleženih podataka na osnovu sličnosti njihovih osobina.
- Primer: Optimalan raspored parkirališta u gradu.
- Primer: Optimizacija veličine vrata i dužine ruku košulje.

Vrste problema klasifikacije:

1. Traženje strukture unutar skupova neklasifikovanih podataka, sa pretpostavkom o kategorijama.
   - Primer: podaci o modelima automobile
     - cene, efikasnost, gorivo, veličina točkova itd.

2. Veštačko deljenje podataka čak i ako ne postoji očigledno grupisanje.
   - Proizvođač escajga želi da pakuje pribor za jelo – kriterijum: broj viljušaka i noževa i "otmenost"(cena) pribora.

K-means algoritam:

- Korak 0: Skaliranje podataka.
- Korak 1: Slučajan izbor K težišta.
- Korak 2: Odrediti udaljenost tačaka do težišta svakog klastera k.
  - Svaka tačka se pridružuje najbližem klasteru.
- Korak 3: Pronaći nobih K težišta za formirane klastere.
  - Povratak na korak 2 i ponavlja se postupak sve do konvergenice težišta

24. Linearna regresija.

- Predstavlja jednu od osnovnih tehnika mašinskog učenja. Iz grupe je algoritama nadgledanog učenja.
- Koristi se da se pronađe linearna zavisnost između nezavisne promenljive X^n (X^n predstavlja vektor sa n vrednosti) i zavisne promenljive Y, koja zavisi od X^n.
- Koristi se za **predviđanje** vrednosti Y, ukoliko je poznata vrednost X^n. Vrednost Y najčešće predstavlja jednu vrednost iz skupa realnih brojeva.

Ocena kvalliteta linearne regresije:

1. **Koficijent determinacije, r^2** je deskriptivna mera jačine regresione veze, koja meri koliko se dobro regresiona linija prilagođava podacima, tj. koliko ona odstupa od podataka.
   Koeficijent determinacije se uvek kreće u interval [0, 1], pri čemu veće vrednosti pokazuju bolji stepen veze između podataka.

2. **Greške:**
   - Standardna greška u predviđanju računa se kao $\mathbf{e_i = y_i - yp_i}$, pri čemu je y_i izmerena vrednost, a yp_i vrednost dobijena regresijom.
   - Koristeći ovu vrednost, često se koriste sledeće mere za grešku:
     - Prosek apsolutne vrednosti greške (**MAE**).
     - Prosečna relativna greška (**MAPE**).
     - Prosek kvadrata greške (**MSE**).
     - Koren proseka kvadrata greške (**RMSE**).
   - Za sve ove vrednosti ne postoje bitne granice, već samo treba da budu što manje.

3. **Overfitt:**
   - Za linearnu regresiju kažemo da previše dobro predviđa (**overfitt**) ukoliko su predviđanja sa njom mnogo bolja od očekivanog. Pod očekivanim predviđanjima podrazumeva se često predviđanje nad skupom za obuku. Ako je npr. **RMSE** skupa za obuku veća od **RMSE** skupa za testiranje, tada kažemo da je model **overfitted** i da nije pogodan za korišćenje.

25. Vrste gradijentnog algoritma.

- **Paketni ili "šaržni" (Batch Gradient Descent)**
  - Svi podaci se uzimaju odjednom u obzir.
  - Nedostatak:
    - Za veliku količinu podataka (npr. 10^6) svi se moraju učitati iz baze i sporije dolazi do rešenja.

- **Stohastički (Stochastic Gradient Descent)**
  - Izvršava se u svakom koraku za npr. jedan slučajan podatak.
    - Npr. za cenu kuće, 1.korak za kuću 1, 2. korak za kuću 2……
  - Za veliko m je brži od paketnog gradijentnog algoritma.

26. Lokalno ponderisana linearna regresija.

27. Logistička regresija. Postupak određivanja parametara.

- Predstavlja jednu od osnovnih tehnika mašinskog učenja. Iz grupe je algoritama nadgledanog učenja.
- U pitanju je statistički model koji koristi logističku funkciju za modelovanje sistema.
- Koristi se za određivanje verovatnoće da neki podatak zadovoljava određeni uslov, kao što je npr. pripadnost u nekoj grupi. Može se koristiti za klasifikaciju podataka u unapred određebe grupe. Najčešce se vrši klasifikacija podataka u dve grupe.
- U logističkoj regresiji koristi se zavisna promenjiva Y koja može da ima dva stanja, npr. 0 i 1, tačno i netačno,….

Ocena kvaliteta klasifikacije:

- Kvalitet klasifikacije ocenjujemo pomoću:
  - **Preciznost** klasifikacije.
  - **Osetljivost** – procenat tačno klasifikovanih podataka klase 1.
  - **Specifičnost** – procenat tačno klasifikovanih podataka klase 0.

  Ove vrednosti treba da su što veće, tj. što bliže vrednosti 1. Potrebno je da su veće od 0.5, a poželjno da su veće od 0.7 ili čak 0.9.

Postupak oređivanja parametara:

- Pretpostavlja se probabilistički model.
- Primenjuje se princip maksimalne verodostojnosti za ceo skup.
- Jedna od numeričkih metoda za određivanje parametara je Njutn-Rapsonov postupak.

28. K najbližih suseda. Izbor optimalnog k.

- K najbližih suseda je algoritam mašinskog učenja iz grupe algoritama nadgledanog učenja.
- Podaci su predstavljeni vektorom obeležaja X.
- Motivacija: **klasifikacija** – obojene tačke u prostoru, svaka dimenzija odgovara jednoj osobini, a boja odgovara kategoriji.
    - Cilj: klasifikovati novi tačku (odrediti joj boju) ako joj je zadatko mesto u prostoru.
- Meri se udaljenost nove tačke i K najbližih već klasifikovanih podataka.

Algoritam KNN

- Korak 1 : Skaliranje
    - Prilagođavanje atributa da budu uporedivi. Npr. u email poruci skalirati da prosečan broj reči bude 0 sa standardnom devijacijom 1. Uraditi ovo za sve atribute.
- Korak 2: Nova tačka
    - Meri se udaljenost nove tačke do svih ostalih tačaka. Posnatra se K tačaka sa najmanjim rastojanjem – najzastupljenija klasa među njima je i klasa nove tačke.

Izbor optimalnog K

- Ukoliko izaberte malo K dobićete malu pristrasnost (Bias), ali i veliku varijansu (Variance).
- Ukoliko zaberete veliko K dobićete veliku pristrasnost, ali malu varijansu.

Mane KNN

- Klasifikacija novih tačaka može biti veoma spora.
- Neravnomerni podaci (mnogo više podataka jedne klase)
    - Dovode do tendencije dodele novih tačaka toj klasi.
    - Treba odraditi neku vrstu inverzne udaljenosti ili opadajuće eksoponencijalne.

29. Kriptografija (primena, procesi, šifrovanje i bezbednosne pretnje).

- Kriptografija je nauka koja se bavi metodama očuvanja tajnosti informacija šifrovanjem. U bitnim programskim sistemima podaci i servisi se moraju očuvati od bezbedonosnih pretnji.

- Bezbednost u računarskim sistemima je usko povezana sa pojmom **oslonjivosti**(*dependability*).

  Oslonjivost uključuje:

  - **Tajnost** (*confidentiality*) –informacija se daje samo ovlašćenim (autorizovanim) licima/servisima;
  - **Integritet** (*integrity*) –modifikacija podataka zahteva autorizovan pristup.

Osnovni pojmovi:

- **poruka**(P) –podatak ili deo podatka koji strana A šalje strani B.
- **šifrovana poruka** (C) –poruka koja je promenjena sa ciljem da bude nerazumljiva za potencijalne napadače.
- **ključ**(K) –tajni podatak koji omogućava stranama da poruke šifriraju.
- **šifrovanje** (**enkripcija**) (E) –proces u kojem predajna strana modifikuje poruku sa ciljem da ona bude nerazumljiva za potencijalne napadače. Rezultat ovog procesa je *šifrovana poruka.*
- **dešifrovanje** (**dekripcija**) (D) –proces u kojem se od šifrovane poruke pravi izvorna poruka.

Tipovi bezbednosnih pretnji:

- **Tipovi pretnji na nivou mreže:**
  - **Presretanje**(*interception*) –neautorizovana stranka dobije pristup podacima ili servisima. Ovde spada i ilegalno kopiranje podataka.
  - **Prekidanje**(*interruption*) –servis ili podatak postane nedostupan, neupotrebljiv ili uništen. Npr. *denial of service* (DoS).
  - **Modifikacija**(*modification*) –neautorizovana izmena podataka ili izmena servisa, koji više nije u skladu sa specifikacijom.

    - **Fabrikacija**(*fabrication*) –generisanje dodatnih podataka ili aktivnosti koji ne bi postojali u normalnim situacijama. Npr. ilegalno ponavljanje ranijeg zahteva za prenos novca ili dodavanja zapisa u datoteku sa lozinkama.

- Malver (**malware – Malicious Software**) je pretnja.

Mehanizmi sprovođenja bezbednosti:

- **Šifrovanje** (*encryption*) –transformiše podatke u format koji nije razumljiv za napadače.
- **Autentifikacija**(*authentication*) –verifikacija identiteta korisnika, klijenata, servisa, itd.
- **Autorizacija**(*authorization*) –provera da li korisnik ili servis ima pravo na izvršavanje određene akcije.
- **Beleženje istorijata aktivnosti** (*auditing*) –upisivanje u dnevnike događaja (uglavnom tekstualne) koji je entitet čemu pristupio i na koji način.
    - Ovaj mehanizam ne obezbeđuje direktnu zaštitu od napada, ali omogućava naknadnu analizu bezbednosnih problema.

Sifrovanje:

- Šifrovanje je transformacija informacije tako da je njeno pravo značenje skriveno.
    - Potrebno je “posebrno znanje” da se preuzme informacija.
- Šifrovanje koristi tajne ključeve kao “posebno znanje”.
    - Savremeni algoritmi sprečavaju pokušaje otkrivanja ključa grubom silom.

Primer:

- Ako strana A želi da pošalje poruku *m* strani B, onda da bi zaštitila poruku od bezbednosnih pretnji
    1. A će izvršiti šifrovanje (enkripciju) poruke m (dobija se m’);
    2. A će poslati šifrovanu poruku (m’);
    3. B će izvršiti dešifrovanje (dekripciju) poruke (m’ i dobiće m).

30. Osnovni algoritmi kriptografije (podela, primitivni i napredni algoritmi).

- **Primitivan algoritam šifrovanja** :
  - Jednostavna zamena slova je način kodiranja teksta gde se neko slovo zamenjuje drugim, a dekodiranje ponovi zamenu.
  - Ovo nije dovoljno sigurno jer ako se iskoristi ShiftCipher dovoljno je isprobati samo 26 slučajeva.

- **„Binarna podloga“ -One-Time Pad algoritam:**
  - Posmatramo niz bita koji čine poruku p.
  - „Binarna podloga“ k je ključ –slučajno izabran niz bita iste dužine kao poruka.
  - Poruku „postavimo na podlogu“ i dobijamo kodiranu poruku: c = p XOR k
    - Operacija XOR se primenjuje na svaki par bita ki i pi
    - Kodirana poruka je u obliku gde se teško može rekonstruisati neka dodatna informacija o originalnoj poruci.
  - Prijemna strana takođe poseduje isti ključ i poruku postavlja na podlogu: p_2 = c XOR k = (p XOR k) XOR k = p
  - *Osobine algoritma:* Čini kodiranu poruku bezbednom, koristi ključ dužine poruke, i ponovna upotreba ključa je rizična.

Podle algoritama šifrovanja:

- **Simetrična kriptografija** – kriptografski sistemi sa deljenim ključem.
  - Koristi isti ključ za šifrovanje i dešifrovanje. Da bi komunikacija bila bezbedna , ključevi moraju biti tajni.
- **Asimetrična kriptografija**
  - Odvojeni ključevi za šifrovanje i dešifrovanje.
    - Svaki od učesnika u komunikaciji ima dva ključa:javni i tajni.
  - Ovakvi sistemi se zovu “sistemi sa javnim ključevima”.
  - Princip rada:
    - Proces šifrovanja koristi javni ključ K_E da napravi šifrovanu poruku.
    - Takvu poruku može da pročita samo process dešifrovanja koji koristi tajni kljuc K_D.

- **AES:**
  - **Advanced Encryption Standard (AES)** je javno dostupan i besplatan algoritam šifrovanja simetričnim ključem. Koristi ključ od 128, 192 ili 256 bita, a blokove dužine 128 bita.
  - Ovaj algoritam je:
    - Bezbedan – otporan na kriptoanalizu, ispravnost matematike, slučajnosti izlaza, itd.
    - Jeftin – brz i ima male memorijske zahteve.
    - Dobrih karakteristika – jednostavan, fleksibilan, pogodan za hardversku i softversku implementaciju.

31. RSA algoritam (principi, uključeni algoritmi).

- RSA je algoritam šifrovanja asimetričnim ključem.
- Algoritam se oslanja na broj koji proizvodi 2 velika prosta broja.
  - Dovoljno je velik da niko ne može otkriti činioce u razumno dugom vremenskom periodu, nrp. broj 2048 je dovoljno velik.
- *Princip rada:*
  1. Izabrati 2 velika (najmanje 1024 bita svaki) različita prosta broja p i q.
  2. Izračunati n = pq
  3. Izračunati r = (p-1)(q-1)
  4. Izabrati mali broj e tako da su e i r uzajamno prosti.
  5. Izračunati d kao inverzan element za množenje, tj. da je *ed mod r* = 1
  6. RSA javni ključ je tada par $K_E = (e,n)$
  7. RSA privatni ključ je $K_D = (d,n)$
  8. Funkcije šifrovanja i dešifrovanja su : $E_{KE}(x) = x^e \bmod n$, $D_{KD}(x) = x^d \bmod n$

- Uključeni algoritmi u ovaj algoritam su:
  - Algoritam da se generiše veliki prost broj za kratko vreme.
  - Euklidov algoritam za proveru izračunavanje broja tako da bude uzajamno prost drugom broju.

32. Lanac blokova (*Blockchain*).

- *Blockchain* predstavlja decentralizovanu, distribuiranu i (uglavnom) javnu kolekciju podataka koja je logički organizovana u blokove.
- Svaki blok je logički povezan sa prethodnikom čime se stvara lanac- i odatle je nastao naziv ove arhitekture(*Blockchain*).
- Povezanost blokova se ostvaruje pomoću kriptografije - svaki blok sadrži heš(*Hash*) vrednost prethodnog bloka.
- Podaci u svakom bloku predstavljaju skup transakcija koje korisnici kreiraju.

Osnovni principi:

- *Blockchain* se može posmatrati kao distribuirana knjiga podataka koja nema centralno skladište.
- Podaci su dostupni svakom čvoru u mreži čime se postiže transparentnost.
- Svaka promena (dodavanje novog bloka) je javna i svaki čvor dobija najnovije stanje u mreži.
- Dodavanje novih blokova vrše „rudari", tj. povezani računari koji koriste veliku količinu električne energije prilikom rešavanja kriptografskih zadataka.
- *Blockchain* tehnologija se najviše primenjuje u oblasti kriptovaluta. Takođe, može se koristiti za skladištenje medicinskih nalaza, kreiranje pametnih ugovora, za analizu poslovnih procesa, itd.

Glavni elementi Blockchain-a su: Heš, lanac blokova, direktna komunikacija, digitalni potpis, algoritmi koncenzusa.

Heš: Heš vrednost se dobija pomoću heš funkcije. Ulazne vrednosti su bilo koji tip podataka, dok su izlazne najčešće u heksadecimalnom zapisu. SHA256 algoritam je jedan od najčešće korištenih heš funkcija.

Lanac blokova: Transakcije između korisnika u mreži se skladište u blokove koji su međusobno povezani, čime se kreira lanac.

Blok sadrži: Veličinu, broj transakcija, transakcije, zaglavlje.

33. Dokazi bez otkrivanja informacija (*Zero Knowledge proof*).

- **ZKP (*Zero-Knowledge Proof*)** protocol dokazuje Viktoru da Pegi ima neke informacije, ali nikako ne otkriva njihov sadržaj.
- Odnos Prover (Pegi) –Verifier (Viktor).
- Viktor postavlja Pegi niz pitanja.
- Ako Pegi zna tajnu –tačno će odgovoriti na svaki pitanje.
- Suština je da Pegi ne otkriva tajnu

Osnovni protokol faze:

1. Pegi koristi svoju informaciju i slučajno generisan broj kako bi svoj težak problem transformisala u drugi i izomorfni težak problem
2. Pegi predaje rešenje novog primera
3. Pegi otkriva Viktoru novi primer (ali time ne otkriva izvorne informacije)
4. Viktor traži od Pegi da:
    a. Da su stari i novi problem izomorfni.
    b. Otvori rešenje iz koraka 2. i dokaže da je to rešenje novog problema
5. Pegi pristaje
6. Pegi I Viktor ponavljaju korake od 1. do 5. n puta

Ovaj protokol je interaktivan.

Treća osoba (Kerol) ne može da bude ubeđena u dokaz (nije uključena u interakciju).

Neinteraktivno dokazivanje bez otkrivanja potpunih informacija

- Neinteraktivno dokazivanje – Pegi može da objavi dokaz svima koji imaju vremena da provere da je on validan.

1. Pegi koristi svoje informacije i n slučajnih brojeva – transformiše težak problem u n različitih teških problema.
2. Pegi predaje rešenja n novih problema.
3. Pegi sve što je predala koristi ko ulazni podatak jednosmerne hash funkcije. Ona zatim uzima i čuva prvih n bita dobijenog rezultata.

4. Pegi uzima prvih n bita generisanih u koraku 3. i za svaki i-ti novi težak, redom, ona uzima i-ti bit (od n):
    a. Ako je 0, ona dokazuje da su stari i novi problem izomorfni.
    b. Ako je 1, ona otvara rešenje iz koraka 2 i dokazuje da je to rešenje novog problema.
5. Pegi objavljuje sva predate rešenja iz koraka 2 i rešenja iz koraka 4
6. Viktor, Kerol i drugi zainteresovani potvrđuju da su koraci 1-5 ispravni.

34. Problem vizantijskih generala.

Problem: Nekoliko divizija Vizantijske vojske logoruje izvan neprijateljskog grada. Svakom divizijom komanduje njen general. Generali mogu komunicirati jedni sa drugima samo preko glasnika. Nakon posmatranja neprijatelja, moraju se odlučiti za zajednički plan akcije. Generali treba da postignu konsenzus o planu kada treba biti napad, a kada povlačenje.
Mogu postojati dva tipa generala: **lojalni** i **izdajnici.**
Svi lojalni generali bi trebali postići konsenzus.

Pojednostavljena verzija: *Komandantski general* šalje naređenje svojim n-1 general – potpukovnicima tako da:

1. Svi lojalni generali su poslušni – sporazum.
2. Ako je komandujući general lojalan, onda svaki odani general-potpukovnik poštuje naređenje koje šalje komandujući – validnost.

Uslovi koji moraju biti ispunjeni:

1. Svi lojalni generali odlučuju o istom planu akcije.
2. Mali broj izdajnika ne može izazvati loš plan odanih generala.

Problem neće uspeti ako su 1/3 ili više generala izdajice.
Nema rešenja kada je manje od 3m + 1 generla, gde je m broj izdajnika.

Komunikacioni modeli:

- Usmene poruke:
    - Ekvivalentno neautorizovanim porukama.
    - Sadržaj je u potpunosti pod kontrolom pošiljaoca.
- Potpisane poruke:
    - Ekvivalentno potvrđenim porukama.
    - Može se otkriti svaka izmena sadržaja.