

INTELLIGENTNI SISTEMI

GENETSKI ALGORITMI IN KLASIFIKACIJA

GENETSKI ALGORITMI

Prof. Jurij F. Tasič

Emil Plesnik

Genetski algoritmi
Simulirano kaljenje
Tabu iskanje

Genetski algoritmi

Evolucijski algoritmi so prisotni že od zgodnjih šestdesetih let.

Uporabljajo naravna pravila: evolucija prek naravnega izbora najboljših posameznikov, ki v tem primeru predstavljajo rešitve matematičnih problemov.

Genetski algoritmi so do sedaj najboljši in **najbolj robustni** evolucijski algoritmi.

Iznašel jih je **John Holland** in jih predstavil v knjigi "Adaption in Natural and Artificial Systems" leta 1975.

Kaj je posebnega pri genetskih algoritmih?

- Genetski algoritem (GA) je hevrističen, kar pomeni, da poišče približek rešitve, kar se tudi ujema z večino problemov v realnem življenju: približek je pogosto povsem zadovoljiva rešitev, pa tudi poiskati ga je lažje kot natančno vrednost.
- Za večino problemov ne obstaja enačba za rešitev zaradi prevelike kompleksnosti. Če pa enačba obstaja, traja njen izračun predolgo.
- Med genetskimi in drugimi hevrističnimi algoritmi je več razlik, med katerimi je najpomembnejša, da **GA deluje na množici možnih rešitev**, medtem ko drugi hevristični algoritmi v svojih iteracijah uporabljajo eno samo rešitev. Druga pomembna ralička je, da so GA verjetnostni (stohastični) in ne deterministični.
- Vsaka posameznik v populaciji GA predstavlja možno rešitev problema. Predlagana rešitev je zakodirana v "genih" posameznika. Tako ima lahko en posameznik naslednje gene: "1100101011", spet drugi pa: "0101110001".

Kaj je posebnega pri genetskih algoritmih?

Izvirna rešitev: **uporaba naravnih pravil evolucije** nad posamezniki. Poiščemo posameznike, za katere menimo, da predstavljajo najboljši predlog rešitve našega problema in jih potem **kombiniramo** z novimi posamezniki.

S ponavljanjem te metode, se bodo iz populacije razvile primerne/dobre rešitve. Specifični elementi GA so: selekcija (merilo pripravljenosti), križanje (metoda reprodukcije, "parjenje" posameznikov, da dobimo nove posameznike) in mutacija (dodajanje majhnega deleža naključnega šuma potomcem, da spremenimo njihove "gene"). Opazimo lahko, da imajo Darwinovi principi velik vpliv na GA.

Zaradi česa je GA "genetski"?

Algoritem mora vsebovati naslednje elemente, da postane "genetski":

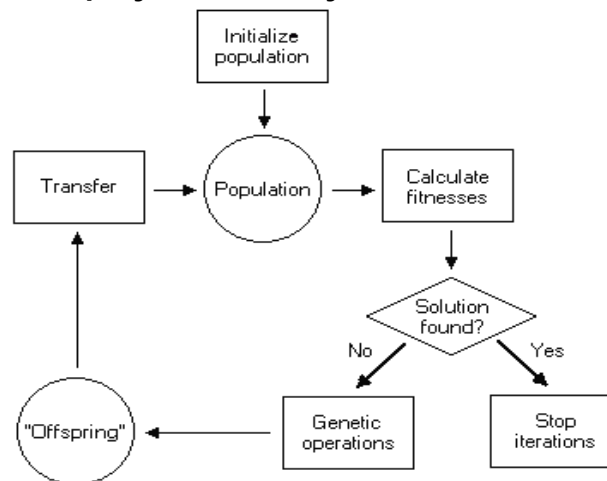
- Matematična predstavitev rezultatov. To je niz vrednosti. Vrednost na določenem položaju ima določen pomen (podobno kot človeški geni). Vzemimo primer optimizacije prostora, pri katerem s številčnim nizom predstavimo postavitev 3 škatel v x,y,z koordinatah. Zaradi enostavnosti je vsaka koordinata predstavljena z decimalnim številom tako, da velja: položaj 1 nam pove koordinato x škatle 1, položaj 2 koordinato y škatle 1 in položaj 3 koordinato z škatle 1, Položaj 4 predstavlja koordinato x škatle 2 itd. Slika prikazuje primer takšnega niza, ki ga sestavlja 9 elementov in predstavlja specifično rešitev:

→	209	680	145
	xyz	xyz	xyz
	box 1	box 2	box 3

- Metodo določanja začetne populacije. Sami določimo koliko posameznikov želimo. Določitev začetne populacije lahko prilagodimo najboljši rešitvi, ki jo predvidevamo. Kljub temu je v večini primerov bolje začeti z naključnimi vrednostmi.

Zaradi česa je GA "genetski"?

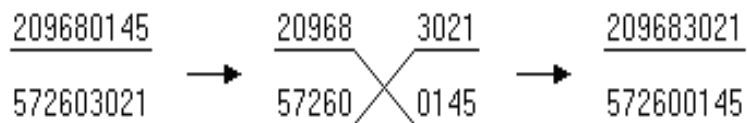
- Metoda za oceno/merjenje pripravljenosti. Potrebujemo merilo pripravljenosti, da lahko izberemo najboljše posameznike. V tem primeru je najbolj očitna izbira merila, koliko prostega prostora bo določena rešitev ponujala. Več prostega prostora pomeni boljšo rešitev.
- Genetske funkcije. Te funkcije so selekcija, križanje in mutacija, ki smo jih že omenili. Podrobnejša razlaga sledi kasneje; za zdaj bo dovolj, če povzamemo, da te funkcije rekombinirajo obstoječe posameznike v nove, ki predstavljajo nove rešitve.
- Število parametrov. Vnaprej moramo določiti velikost populacije, število staršev za izbiro, stopnjo mutacije itd..



Genetske operacije GA

Delovanje genetskih operacij:

- Vzemimo še enkrat primer optimizacije prostora. Metoda pripravljenosti enostavno izračuna velikost prostega prostora za vsako ponujeno rešitev. Najboljše so izbrane za nadaljnje iteracije.
- Metoda križanja za kombiniranje izbranih posameznikov v nove. Spomnimo, da so posamezniki v tem primeru nizi števil. Križanje razdeli "starše" in jih potem rekombinira. Prikazan je primer križanja dveh "staršev" v nove "otroke":



Mutacija "genom" posameznikov preprosto doda naključni šum (običajno "otrokom"). To je način spreminjanja "genetskega bazena".

Obstaja več različnih metod selekcije (koliko staršev izbrati)? Kako meriti pripravljenost? itd., veliko metod križanja (število točk za križanje, način sestave ločenih delov) in veliko metod mutacije (naključni šum, zamenjava vrednosti, brez mutacije, kaj mutirati itd.).

Izbira metod je odvisna od problema in osebnih preferenc.

Prednosti in slabosti GA

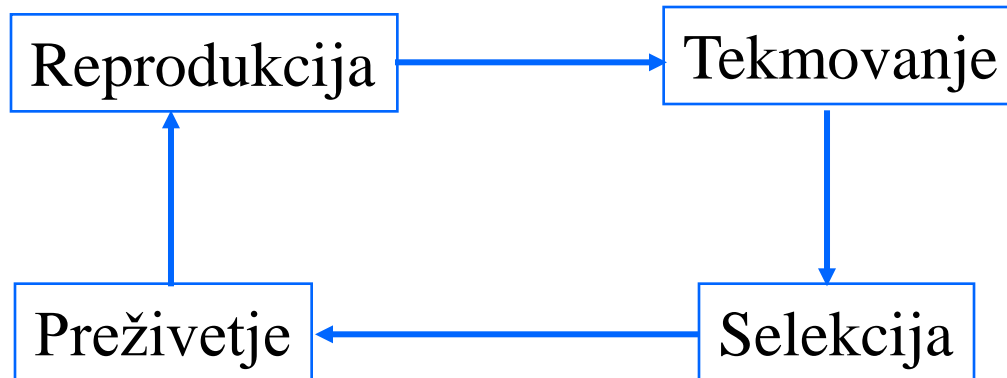
Genetski algoritmi imajo številne **prednosti**. Hitra obdelava velikih zbirk rešitev. Slabi predlogi na vplivajo slabo na končno rešitev, saj so preprosto izločeni. Induktivna narava GA pomeni, da ni potrebno poznavanje pravil problema – ta deluje po svojih (notranjih) pravilih. To se izkaže za zelo uporabno pri slabo definiranih problemih.

Velika prednost GA je, da iščejo rešitev z evolucijo, kar pa je obenem tudi največja slabost. Evolucija je induktivna: v naravi se življenje na razvija proti dobrim rešitvam, ampak proč od slabih pogojev. To lahko povzroči, da se določena vrsta razvije v evolucijsko slepo ulico. Podobno, je tudi pri GA možno, da je ponujena rešitev neoptimalna.

Primer: GA naj poišče najvišjo točko reliefa. Algoritem bo seveda dal prednost tistim točkam, ki so vrhovi vzpetin. Postopoma se začno ponujati rešitve v okolici teh vrhov in postopoma so si te rešitve vse bolj podobne. Na kocu je lahko ponujenih več rešitev, ki so si skoraj popolnoma enake. Najboljše med temi predlagajo kot rešitev vrh vzpetine. Vprašanje je: kaj če je na drugem koncu reliefa drug, višji vrh? Za skupino rešitev je odmik od trenutnega vrha prezahteven. Tiste, ki se odmaknejo so hitro izločene, saj so slabše od trenutno najboljših. Posamezni rešitvi se odmik lahko posreči, vendar pa to pomeni, da so njeni "geni" zelo različni od ostalih, kar pa ni verjetno. are very different from the rest of the population, so this is unlikely. Algoritem je tako ponudil neoptimalno rešitev, česar se morda niti ne zavedamo.

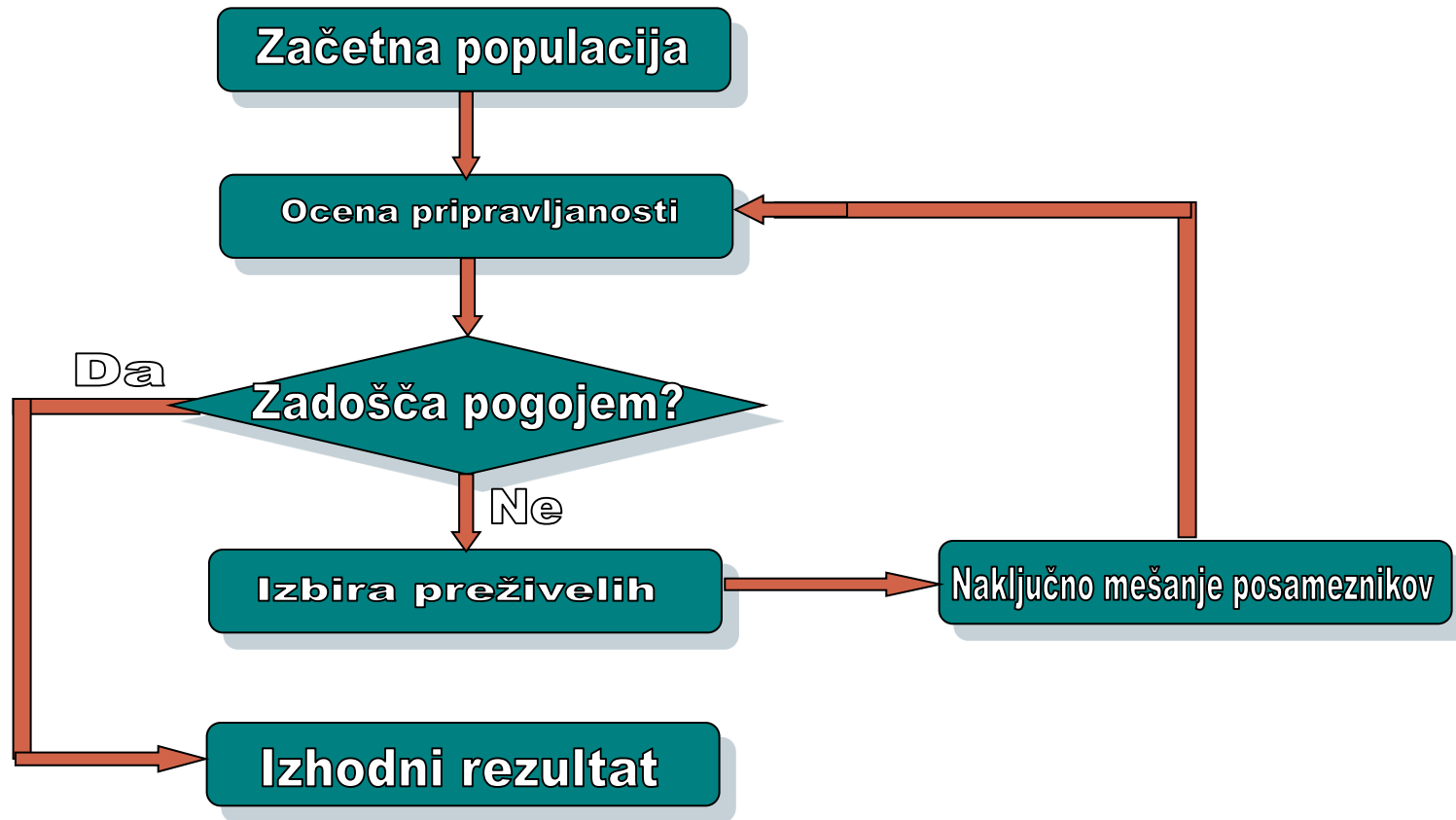
Genetski algoritem

- Osnovan na Darwinovi paradigmi



- Robustni iskalni in optimizacijski mehanizem

Konceptualni algoritem



GA – Uvod 1

- Sledijo **naravni evoluciji**
- **Populacija** posameznikov
 - Posameznik predstavlja možno rešitev problema
- Vsak posameznik je ocenjen s **funkcijo pripravljenosti**
 - Boljša pripravljenost pomeni boljšo rešitev
- Starši so izbrani glede na njihovo pripravljenost za reprodukcijo **potomcev** za novo **generacijo**
 - boljše pripravljene posamezniki imajo več možnosti za reprodukcijo
 - nova generacija ohranja velikost stare, ki umre
- Potomci imajo **kombinacijo** lastnosti staršev
- Ob dobrem načrtovanju bo populacija **konvergirala** proti optimalni rešitvi

Algoritem

BEGIN

Ustvari začetno populacijo;

Izračunaj pripravljenost vsakega posameznika;

REPEAT /* Nova generacija/*

FOR velikost_populacije / 2 DO

Izberi dva starša iz stare generacije;

/* prednost imajo bolj pripravljeni */

Rekombinacija staršev v dva potomca;

Izračunaj pripravljenost potomcev;

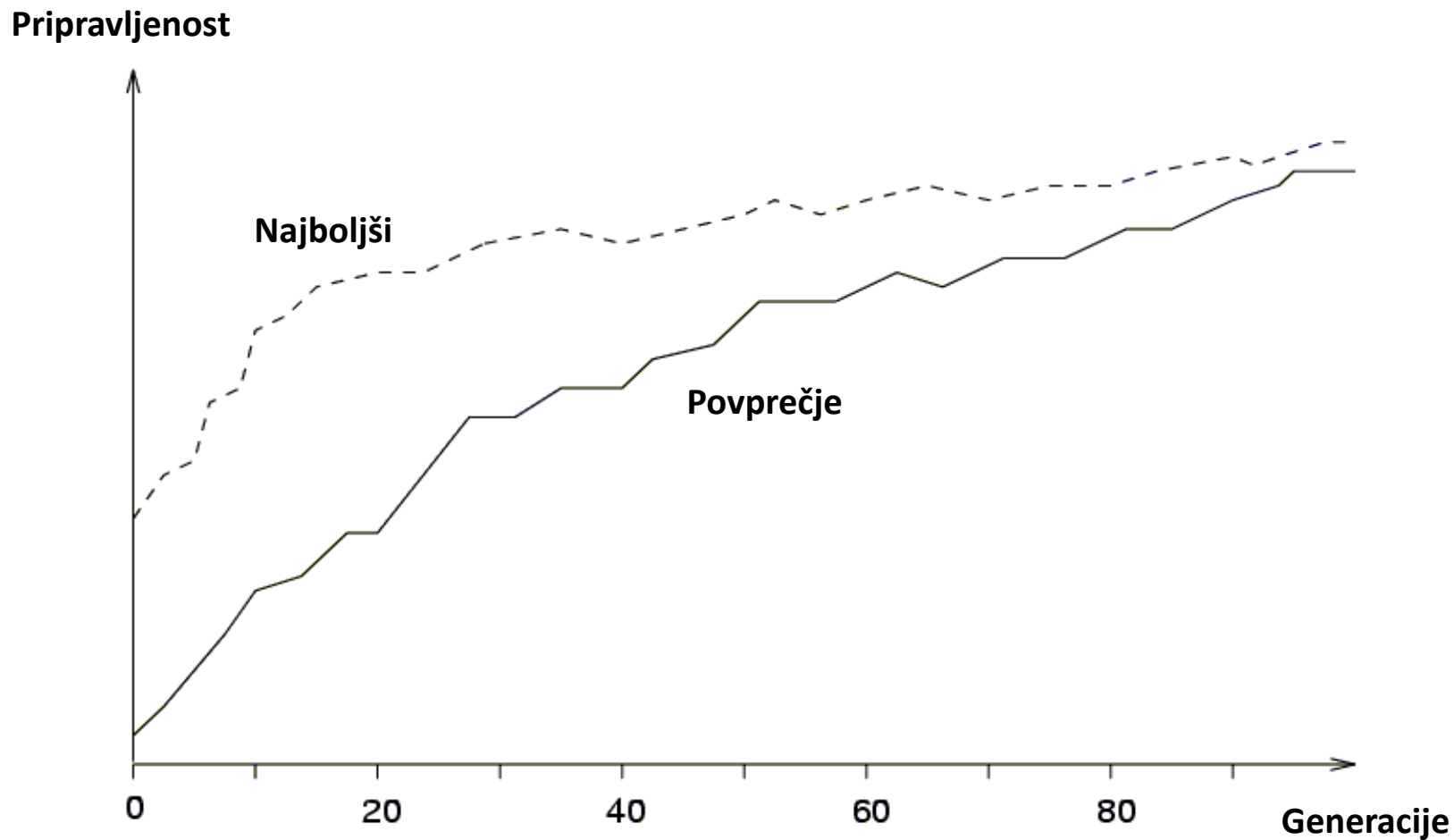
vstavi potomce v novo generacijo

END FOR

UNTIL konvergenca populacije

END

Primer konvergence



Uvod 2

- Reprodukcijski mehanizmi nimajo informacije o problemu, ki se rešuje
- Povezava med GA in problemom:
 - Kodiranje
 - Funkcija pripravljenosti

Osnovni principi 1

- Kodiranje ali predstavitev
 - Niz z vsemi parametri
- Funkcija pripravljenosti
 - Izbira staršev
- Reprodukciija
 - Križanje
 - Mutacija
- Konvergenca
 - Ustavitveni pogoj

Osnovni principi 2

- Posameznik je opisan z naborom lastnosti: **Geni**
 - Geni so združeni v niz: **Kromosomi**
-
- Kromosomi sestavljajo **genotip**
 - Genotip vsebuje vse potrebne informacije za sestavo organizma: **fenotip**
-
- **Reprodukcija** je “neumen” proces nad kromosomi **genotipa**
 - **Pripravljenost** je izmerjena v realnem svetu (‘boj za obstanek’) **fenotipa**

Kodiranje

- Parametri rešitve (**geni**) so sestavljeni v niz (**kromosom**)
- Za kromosom so lahko uporabljene vse vrste **abeced** (številke, črke), v splošnem pa se uporablja **binarna abeceda**
- **Vrstni red** genov v kromosomu je lahko pomemben
- V splošnem je lahko za parametre rešitve možnih več **različnih kodiranj**
- **Dobro kodiranje je najverjetneje najpomembnejši faktor za zmogljivost GA**
- Pogosto za kromosome ni možno kodiranje izvedljivih rešitev

Genetski algoritem

- Kodiranje
- Ocena pripravljenosti
- Reprodukција
- Izbira preživelih/najboljših

Kodiranje

- Oblikovanje alternative → posamezniki (kromosom)
- Izbira enotnega načrta → geni
- Cilji načrta → pripravljenost

Primer

Problem

Razvrstimo n opravil na m procesorjev tako, da se minimizira največji razpon.

Alternativa: opravilo i ($i=1,2,\dots,n$) je dodeljeno procesorju j ($j=1,2,\dots,m$)

Posamezno: Vektor \mathbf{x} , da velja $x_i = 1, \dots, \text{or } m$

Cilj: minimizacija največjega razpona

Pripravljenost: največji razpon vsakega procesorja

Reprodukcija

- Operatorji reprodukcije
 - Križanje
 - Mutacija

Reprodukcija

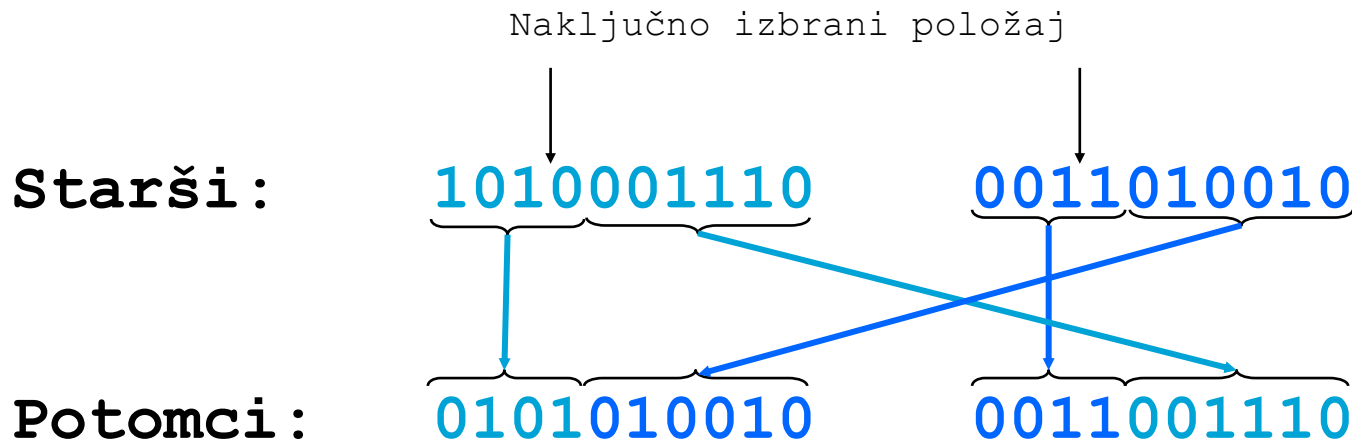
- **Križanje**
 - Dva starša imata dva potomca
 - Možno je, da se kromosomi staršev nespremenjeni skopirajo na potomce
 - Možno je, da se kromosomi staršev naključno rekombinirajo (se križajo) v potomce
 - V splošnem je verjetnost križanja med 0.6 in 1.0
- **Mutacija**
 - Tmožno je, da se gen potomca naključno spremeni
 - V splošnem je verjetnost mutacije nizka (npr. 0.001)

Operatorji reprodukcije

- Križanje
 - Generiranje potomcev iz dveh staršev
 - Križanje v eni točki
 - Križanje v dveh točkah (Križanje v več točkah)
 - Enotno križanje

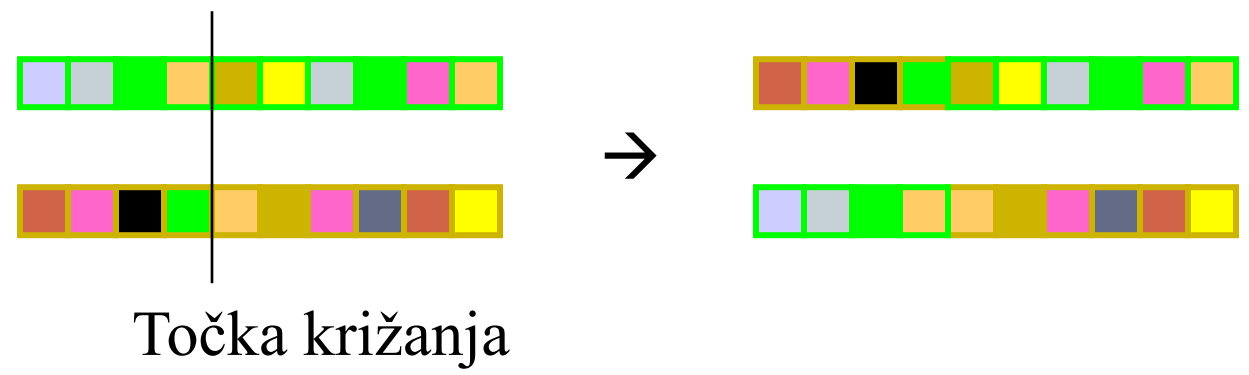
Križanje v eni točki 1

- Točka križanja je naključno izbrana v kromosomu
- Potomec 1 prevzame glavo kromosoma starša 1 in rep kromosoma starša 2
- Potomec 2 prevzame glavo kromosoma starša 2 in rep kromosoma starša 1

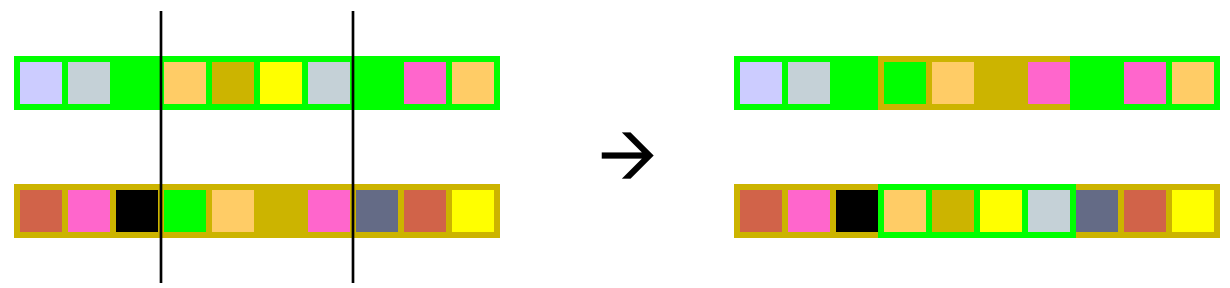


Primerjava operatorjev reprodukcije

- Križanje v eni točki

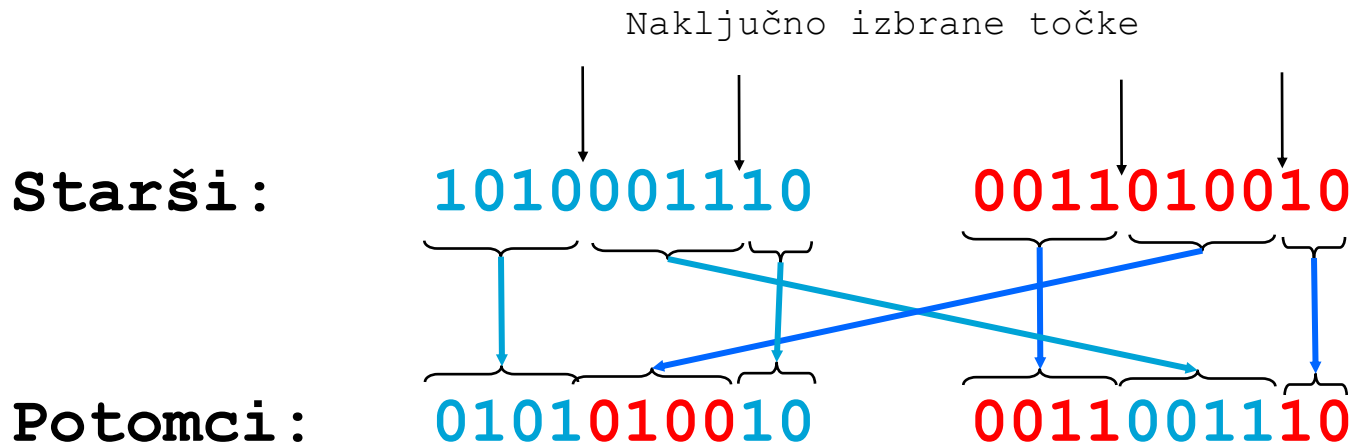


- Križanje v dveh točkah (Križanje v več točkah)



Križanje v dveh točkah

- Naključno sta izbrani dve točki križanja v kromosomu
- Preprečuje, da bi se geni v glavi in repu kromosoma vedno delili ob rekombinaciji



Enotno križanje

- Ustvarjena je naključna maska
- Maska določa, kateri deli kromosoma se kopirajo od določenega starša
- Maska določa tudi količino materiala, ki se vzame od staršev

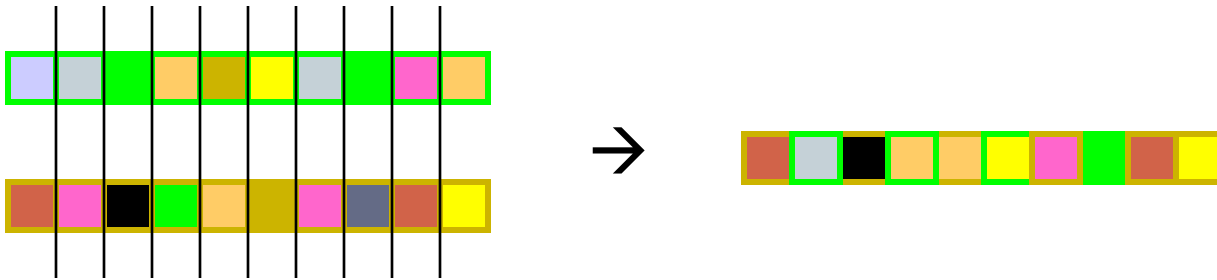
Maska: 0110011000 (naključna)

Starši: 1010001110 0011010010

Potomci: 0011001010 1010010110

Operatorji reprodukcije

- Enotno križanje



- Ali je enotno križanje boljše kot križanje v eni točki?
 - Kompromis med
 - **Raziskovanjem**: uvajanje novih kombinacij lastnosti
 - **Izkoriščanjem**: obdržati dobre lastnosti obstoječih rešitev

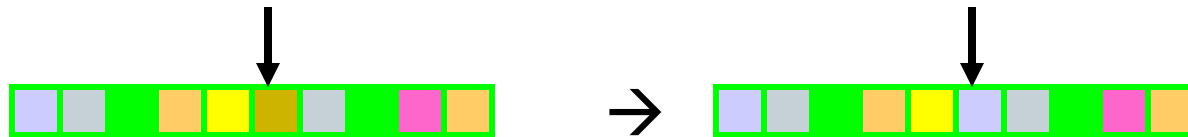
Problemi križanja

- Odvisno od kodiranja, lahko imajo preprosta križanja veliko verjetnost generiranja nedovoljenih potomcev
 - npr. pri TSP s preprostim binarnim kodiranjem bo večina potomcev nedovoljenih, saj vsa mesta ne bodo med njimi, nekatera pa se bodo večkrat ponovila
- Enotno križanje se ob ustreznih nastavitvah lahko temu izogne
 - Npr. pri TSP s kodiranjem “simple path”:
 - Kjer je maska 1, kopiraj mesta od enega starša
 - Kjer je maska 0, izberi preostala mesta od drugega starša

Operatorji reprodukcije

- **Mutacija**

- Ustvarjanje potomcev iz enega samega starša



- Ohranjanje raznovrstnosti posameznikov
 - Križanje ponuja zgolj različne kombinacije iz trenutnega bazena genov
 - Mutacija lahko ustvarja nove gene

Operatorji reprodukcije

- **Nadzorni parametri:** velikost populacije, verjetnost križanja/mutacije
 - Odvisni od vrste problema
 - Povečanje velikosti populacije
 - Povečanje raznovrstnosti in časa računanja za vsako generacijo
 - Povečanje verjetnosti križanja
 - Povečanje možnosti rekombinacije, hkrati pa tudi izgubo dobre kombinacije
 - Povečanje verjetnosti mutacije
 - Približanje naključni izbiri
 - Pomoč pri vpeljavi novih ali izgubljenih genov
- **Spreminja populacijo**

Običajno se operatorji križanja uporabijo za rekombinacijo genov za novo generacijo, nad katero se nato uporabijo še operatorji mutacije.

Izbira staršev/preživelih

- Strategije
 - Izbira preživelih
 - Vedno obdrži najboljšega
 - Elitistično: izbris K najslabših
 - Izbira verjetnosti: obratno s pripravljenostjo
 - itd.

Izbira staršev/preživelih

- Selekcija, ki se preveč opira na pripravljenost lahko vodi v neoptimalen rezultat
- Premajhna prisotnost pripravljenosti pri izbiri pa povzroči neosredotočeno in vijugasto iskanje

Izbira staršev/preživelih

Verjetnost izbire starša je proporcionalna pripravljenosti

- Ruleta

Da se izognemo problemom s funkcijo pripravljenosti

- Turnir

Drugi

- Globalno optimalni
- Uглаševanje parametrov
- Paralelizem
- Generatorji naključnih števil

Primer kodiranja za TSP

TSP = Travelling Salesman Problem (problem trgovskega potnika)

- Binarno
 - Mesta so binarno kodirana; kromosom je niz bitov
 - Večina kromosomov predstavlja nedovoljene poti
 - Obstaja več kromosomov za isto pot
- Pot
 - Mesta so oštevilčena; kromosom je niz celih števil
 - Večina kromosomov predstavlja nedovoljene poti
 - Obstaja več kromosomov za isto pot
- Ordinalno (zaporedno)
 - Mesta so oštevilčena, koda je kompleksna
 - Vsi možni kromosomi so dovoljeni, za vsako pot obstaja točno en kromosom
- Več drugih

Turnir

- Binarni turnir
 - Two individuals are randomly chosen; the fitter of the two is selected as a parent
- Verjetnostni binarni turnir
 - Dva posameznika sta naključno izbrana; z verjetnostjo p , $0.5 < p < 1$, je bolje pripravljeni izbran za starša
- Večji turnirji
 - n posameznikov je naključno izbranih; najbolje pripravljeni je izbran za starša
- S spreminjanjem n in/ali p lahko GA dinamično prilagajamo

Težave z dosegom pripravljenosti

- Prezgodnja konvergenca

- Prevelika pripravljenost
- Relativno super pripravljeni posamezniki dominirajo v populaciji
- Populacija konvergira k lokalnemu maksimumu
- Prevelika izkoriščenost; premajhno raziskovanje

- Počasno končanje

- Premajhna pripravljenost
- Ni pritiska izbire
- Po več generacijah povprečna pripravljenost konvergira, vendar globalni maksimum ni dosežen; nezadostna razlika med najboljšo in povprečno pripravljenostjo
- Premajhna izkoriščenost; preveliko raziskovanje

Težave z dosegom pripravljenosti

- Uporabimo turnirsko izbiro
 - Implicitna ponastavitev pripravljenosti
- Prilagoditev funkcije pripravljenosti za ruleto
 - Eksplicitna ponastavitev pripravljenosti
 - Skaliranje
 - Oknjenje
 - Rangiranje

Funkcija pripravljenosti

Namen

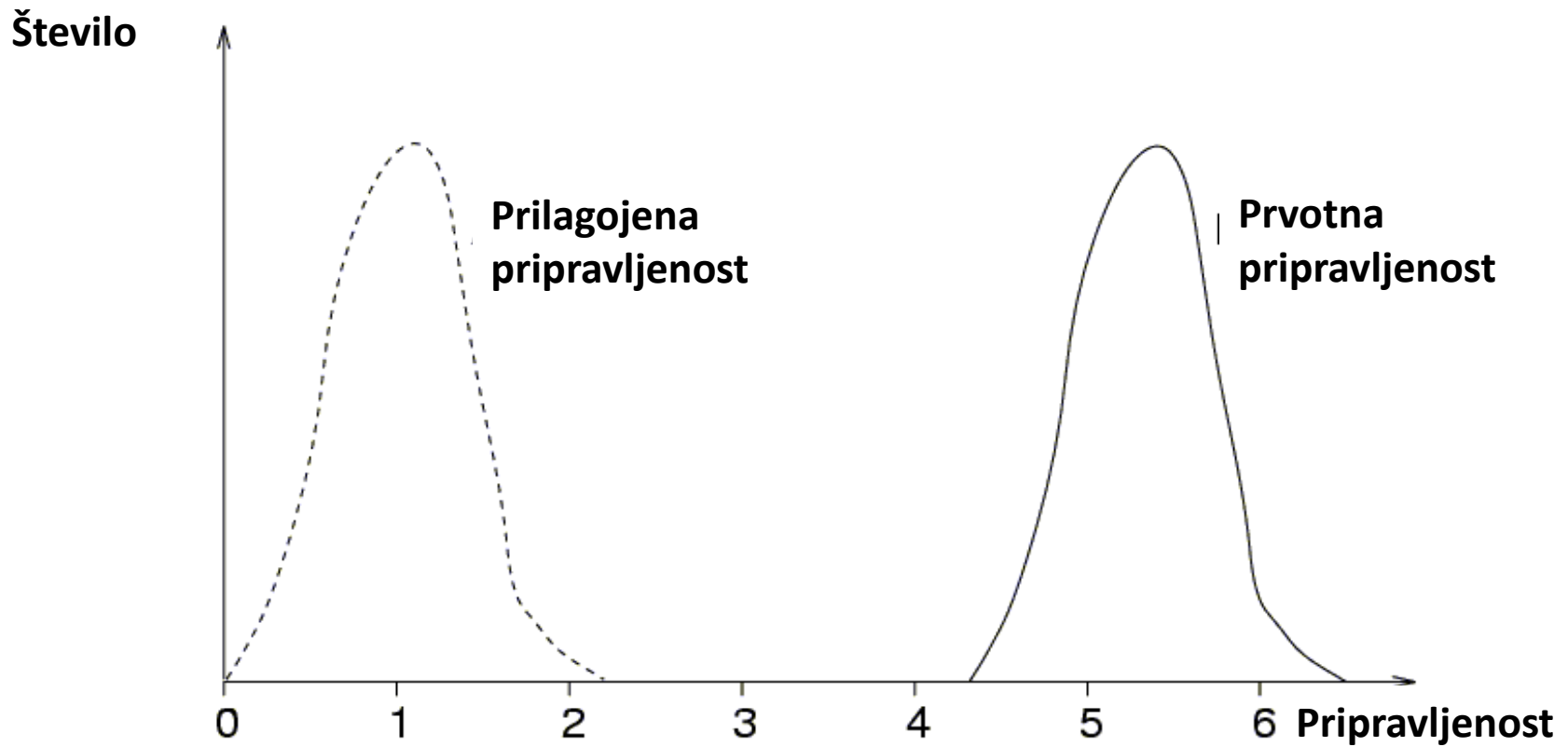
- Izbira staršev
- Merilo konvergence
- Za stabilno stanje: selekcija odpadlih/umrlih posameznikov

- Naj bi odražala vrednost kromosoma
- Poleg kodiranja najpomembnejši del GA

Skaliranje pripravljenosti

- Vrednosti pripravljenosti so skalirane z odštevanjem in deljenjem tako, da je najslabše vrednost 0, najboljša pa 2.
 - Verjetnost za najbolje pripravljenega posameznika je dvakrat večja od povprečja
 - Verjetnost za najslabše pripravljenega posameznika je 0
- Težave, ko je pripravljenost najboljšega posameznika ekstremno visoka (**super-fit**) ali nizka (**super-unfit**)
 - Rešitev je uvedba minimalnih/maksimalnih vrednosti pripravljenosti

Primer skaliranja pripravljenosti



Oknjenje pripravljenosti

- Podobno kot pri skaliranju, le da je vrednost, ki jo odštejemo dejansko minimum iz n prejšnjih generacij, npr. $n = 10$
- Enake težave kot pri skaliranju

Rangiranje pripravljenosti

- Posamezniki so oštevilčeni v zaporedju povečevanja pripravljenosti
- Rang v tem vrstnem redu pomeni popravljeno pripravljenost
- Začetna vrednost in inkrement sta lahko izbrana na več načinov in vplivata na rezultat
- Ni težav s super-fit ali super-unfit posamezniki
- Pogosto boljša kot skaliranje in oknjenje

Ocena pripravljenosti

- Ključna komponenta GA
- Kompromis med časom in kvaliteto
- Več kriterijska pripravljenost

Več kriterijska pripravljenost

- Dominanca in indiferenca
 - Za optimizacijski problem z več kot eno ciljno funkcijo (f_i , $i=1,2,\dots,n$)
 - Za katerikoli rešitvi X_1 in X_2 velja
 - X_1 dominira X_2 ($X_1 \succ X_2$), če
 $f_i(X_1) \geq f_i(X_2)$, za vse $i = 1, \dots, n$
 - X_1 je indiferenten X_2 ($X_1 \sim X_2$), če X_1 ne dominira X_2 , in X_2 ne dominira X_1

Več kriterijska pripravljenost

- Pareto-optimalni set
 - Če v iskalnem prostoru ne obstaja rešitev, ki bi dominirala nad vsakim članom P , potem rešitve iz P sestavljajo globalni Pareto-optimalni set.
 - Pareto-optimalna fronta
- Preverjanje dominantnosti

Glede na začetno razporeditev dobrin med posameznike je sprememba lokacije, ki vsaj enemu posamezniku prinese napredek brez poslabšanja za druge, se imenuje **Pareto napredek**. Razporeditev je definirana kot "Pareto učinkovita" ali "Pareto optimalna", ko ni več mogoče izvesti Pareto napredka.

Več kriterijska pripravljenost

- Utežena vsota
 - $F(\mathbf{x}) = w_1 f_1(x_1) + w_2 f_2(x_2) + \dots + w_n f_n(x_n)$
 - *Problemi?*
 - *Conveksna in conveksna Pareto optimalna fronta*
 - *Občutljiva na obliko Pareto-optimalne fronte*
 - *Izbira uteži?*
 - *Potrebno predznanje*
 - *Nezanesljivo za probleme z negotovostmi*

Več kriterijska pripravljenost

- Optimizacija enega samega cilja
 - *Maximizacija: $f_k(\mathbf{X})$*

Glede na :

$$f_j(\mathbf{X}) \leq K_j \quad i \in \{1, \dots, k\}$$

$\mathbf{X} \in F$ kjer je F prostor možnih rešitev.

Več kriterijska pripravljenost

- Utežena vsota na osnovi preferenc
(**ISMAUT** *Imprecisely Specific Multiple Attribute Utility Theory*)
 - $F(\mathbf{x}) = w_1 f_1(x_1) + w_2 f_2(x_2) + \dots + w_n f_n(x_n)$
 - Preference
 - Glede na dva znana posameznika \mathbf{X} in \mathbf{Y} , če preferiramo \mathbf{X} nad \mathbf{Y} , potem

$$F(\mathbf{X}) > F(\mathbf{Y}),$$
 to je

$$w_1(f_1(x_1) - f_1(y_1)) + \dots + w_n(f_n(x_n) - f_n(y_n)) > 0$$

Več kriterijska pripravljenost

- Vse preference sestavljajo linearni prostor

$$W_n = \{w_1, w_2, \dots, w_n\}$$

$$w_1(f_1(x_1) - f_1(y_1)) + \dots + w_n(f_n(x_n) - f_n(y_n)) > 0$$

$$w_1(f_1(z_1) - f_1(p_1)) + \dots + w_n(f_n(z_n) - f_n(p_n)) > 0, \text{ itd.}$$

- Kako za vsaka nova posameznika Y' in Y določimo, kateri je preferiran?

Već kriterijska pripravljenost

$$\text{Min} : \mu = \sum_k w_k [f_k(\mathbf{Y}') - f_k(\mathbf{Y}'')]]$$

$$\text{s.t.} : \quad W_n$$

$$\text{Min} : \mu' = \sum_k w_k [f_k(\mathbf{Y}'') - f_k(\mathbf{Y}')]]$$

$$\text{s.t.} : \quad W_n$$

Več kriterijska pripravljenost

Potem,

$$\mu > 0 \Rightarrow Y' \succ Y''$$

$$\mu' > 0 \Rightarrow Y'' \succ Y'$$

Drugače,

$$Y' \sim Y''$$

Ustvari dominantni odnos med indiferentnimi glede na preference.

Drugi parametri GA 1

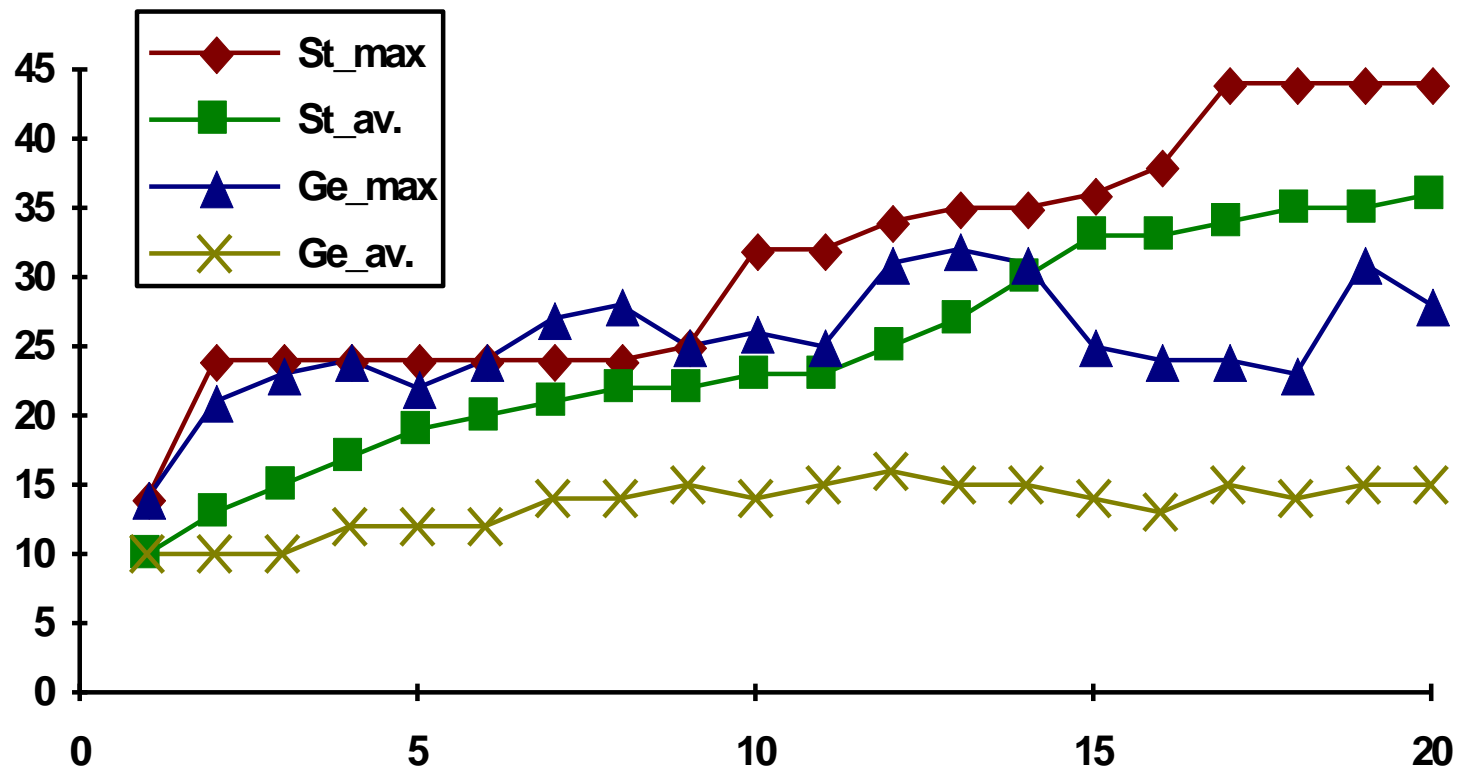
- Inicializacija:
 - Velikost populacije
 - Naključnost
 - Namenski požrešni algoritem
- Reprodukcijska:
 - Generacijska: kot opisano prej (žuželke)
 - Generacijska, elitistična: določeno število najbolj pripravljenih posameznikov je nespremenjenih prenesenih v novo generacijo
 - Stabilno stanje: dva starša sta izbrana za reprodukcijo in dva starša sta izbrana za smrt; dva potomca sta takoj prenesena v bazen (sesalci)

Drugi parametri GA 2

- Ustavitveni kriterij:
 - Število novih kromosomov
 - Število novih in edinstvenih kromosomov
 - Število generacij
- Merilo:
 - Najboljši iz populacije
 - Povprečje populacije
- Dvojniki
 - Razen vsi dvojniki
 - Izogibamo se preveč dvojnikom, saj kvarijo populacijo
 - Brez dvojnikov

Primer

Maximmumi in povprečja stabilnih stanj in generacijska zamenjava



Simulirano kaljenje

- Kaj?
- Analogija med procesom kaljenja in iskanjem optimuma v splošnem sistemu.

Proces kaljenja

- Z visokim dvigom temperature (do tališča) imajo atomi višje energijsko stanje in veliko možnost za prerazporejanje v kristalni strukturi.
- Med počasnim ohlajanjem se niža tudi energijsko stanje atomov in njihova možnost prerazporejanja v kristalni strukturi.

Simulirano kaljenje

- Analogija
 - Kovina \leftrightarrow Problem
 - Energijsko stanje \leftrightarrow stroškovna funkcija
 - Temperatura \leftrightarrow kontrolni parameter
 - Povsem urejena kristalna struktura
 \leftrightarrow optimalna rešitev problema

Globalno optimalno rešitev lahko dosežemo dokler je proces ohlajanja dovolj počasen.

Zanka izvedbe

- Osnovna karakteristika simuliranega kaljenja
- Določitev naključnega pregledovanja nove rešitve, zavrnitve ali sprejetja nove rešitve pri konstantni temperaturi T .
- Končano, ko je dosežen ekvilibrij/ravnovesje.

Izvedbeni kriterij

- Naj bo
 - x trenutna rešitev in x' nova rešitev
 - $C(x)$ ($C(x')$) naj bo energijsko stanje (cena) od x (x')
- Verjetnost $P_{\text{accept}} = \exp [(C(x)-C(x'))/ T]$
- Naj bo $N = \text{Random}(0,1)$
- Brezpogojno sprejeto če
 - $C(x') < C(x)$, nova rešitev je boljša
- Verjetno sprejeto, če
 - $C(x') \geq C(x)$, nova rešitev je slabša.
Sprejeto samo ko $N < P_{\text{accept}}$

Algoritem

določimo začetno rešitev \mathbf{x} , najvišjo temperaturo T_h , in najnižjo temperaturo T_l

$T = T_h$

Ko je temperatura višja kot T_l

While brez ravnovesja

išči novo rešitev \mathbf{x}'

sprejmi ali zavrne \mathbf{x}' glede na Metropolis kriterij

End

zmanjšaj temperaturo T

End

Simulirano kaljenje

- Definicija rešitve
- Iskalni mehanizem, tj. definicija soseščine
- Stroškovna funkcija

Kontrolni parametri

- Definicija ravnovesja
 - Ni bistvenega izboljšanja po določenem številu zank
 - Konstantno število zank
- Urnik kaljenja (tj. kako zmanjševati temperaturo?)
 - Konstantna vrednost, $T' = T - T_d$
 - Konstantni skalirni faktor, $T' = T * R_d$
 - Skalirni faktor običajno bolj zmogljiv

Kontrolni parametri

- Določitev temperature
 - Brez fizikalne teže
 - Začetna temperatura
 - 80-90% stopnja sprejetja
 - Končna temperatura
 - Konstantna verdnost, tj. osnovana na skupnem številu možnih rešitev
 - Brez napredka med izvajanjem zanke
 - stopnja sprejetja majhna
- Odvisna od problema in jo je potrebno usklajevati

Primer

- Potujoči trgovski potnik (TSP)
 - Podanih je 6 mest in stroški potovanja med njimi
 - Potnik naj začne v mestu 1, obižče vsa ostala mesta in se vrne v mesto 1
 - Minimizarati moramo strošek poti

Primer

- Predstavitev rešitve
 - Seznam številskih vrednosti, npr. (1,4,2,3,6,5)
- Iskalni mehanizem
 - Zamenja katerikoli dve števili v nizu (razen prve)
 - (1,4,2,3,6,5) \rightarrow (1,4,3,2,6,5)
- Stroškovna funkcija

Primer

- Temperatura
 - Določitev začetne temperature
 - Približno 80% stopnja sprejetja za “slabo potezo”
 - Določimo sprejemljivost ($C_{\text{new}} - C_{\text{old}}$)
 - Določitev končne temperature
 - Ustavitveni kriterij
 - Stopnja pokritosti prostora rešitev
- Urnik kaljenja
 - Konstanto število (npr. 90%)
 - Odvisno od stopnje pokritosti prostora rešitev

Drugi

- Uглаševanje parametrov
 - Aarts, E. and Korst, J. (1989). *Simulated Annealing and Boltzmann Machines*. John Wiley & Sons.
- Ni enostavno za paralelo implementacijo
- Naključni generator

Optimizacijske tehnike

- Matematično programiranje
- Analiza mreže
- Branch & Bound
- Genetski algoritem
- Simulirano kaljenje
- Tabu iskanje

Plezanje

Pripravljenost

