

Programiranje logičkih igara.

Danijela Petrović

May 7, 2014

Uvod

- Zasnovani su na Darwinovoj teoriji evolucije:

Uvod

- Zasnovani su na Darwinovoj teoriji evolucije:
 - nasleđivanje

Uvod

- Zasnovani su na Darwinovoj teoriji evolucije:
 - nasleđivanje
 - mutacija

Uvod

- Zasnovani su na Darvinovoj teoriji evolucije:
 - nasleđivanje
 - mutacija
 - selekcija

Uvod

- Zasnovani su na Darvinovoj teoriji evolucije:
 - nasleđivanje
 - mutacija
 - selekcija
 - ukrstanje

Uvod

- Zasnovani su na Darvinovoj teoriji evolucije:
 - nasleđivanje
 - mutacija
 - selekcija
 - ukrstanje
- **Koriste se:** za nalaženje tačnog ili približnog rešenja pretrage ili opimizacije;

Uvod

- Zasnovani su na Darvinovoj teoriji evolucije:
 - nasleđivanje
 - mutacija
 - selekcija
 - ukrstanje
- **Koriste se:** za nalaženje tačnog ili približnog rešenja pretrage ili optimizacije;
- Popularni su zbog jednostavnosti i uspešnosti

Uvod

- Zasnovani su na Darvinovoj teoriji evolucije:
 - nasleđivanje
 - mutacija
 - selekcija
 - ukrstanje
- **Koriste se:** za nalaženje tačnog ili približnog rešenja pretrage ili opimizacije;
- Popularni su zbog jednostavnosti i uspešnosti
- Nalaze ekstremume, ali ne postoji garancija da li je u pitanju lokalni ekstremum i sa kolikom greškom je nađeno rešenje

Osnovni genetski algoritmi

- Treba da se postupno približavamo rešenju

Osnovni genetski algoritmi

- Treba da se postupno približavamo rešenju
- Jedinke su reprezentovane nulama i jedinicama

Osnovni genetski algoritmi

- Treba da se postupno približavamo rešenju
- Jedinke su reprezentovane nulama i jedinicama
- Postupak se odvija kroz generacije

Osnovni genetski algoritmi

- Treba da se postupno približavamo rešenju
- Jedinke su reprezentovane nulama i jedinicama
- Postupak se odvija kroz generacije
- **Početna generacija:** slučajno generisan skup jedinki ili dobijen korišćenjem neke druge metode optimizacije

Osnovni genetski algoritmi

- Treba da se postupno približavamo rešenju
- Jedinke su reprezentovane nulama i jedinicama
- Postupak se odvija kroz generacije
- **Početna generacija:** slučajno generisan skup jedinki ili dobijen korišćenjem neke druge metode optimizacije
- **Funkcija cilja** (funkcija prilagođenosti, kvaliteta) – za svaku jedinku računa njen kvalitet

Osnovni genetski algoritmi

- Treba da se postupno približavamo rešenju
- Jedinke su reprezentovane nulama i jedinicama
- Postupak se odvija kroz generacije
- **Početna generacija:** slučajno generisan skup jedinki ili dobijen korišćenjem neke druge metode optimizacije
- **Funkcija cilja** (funkcija prilagođenosti, kvaliteta) – za svaku jedinku računa njen kvalitet
- **Selekcija** – iz jedne generacije se biraju jedinke koje će biti prenete u novu generaciju i iskorišćene za pravljenje novih jedinki

Osnovni genetski algoritmi

- Treba da se postupno približavamo rešenju
- Jedinke su reprezentovane nulama i jedinicama
- Postupak se odvija kroz generacije
- **Početna generacija:** slučajno generisan skup jedinki ili dobijen korišćenjem neke druge metode optimizacije
- **Funkcija cilja** (funkcija prilagođenosti, kvaliteta) – za svaku jedinku računa njen kvalitet
- **Selekcija** – iz jedne generacije se biraju jedinke koje će biti prenete u novu generaciju i iskorišćene za pravljenje novih jedinki
- **Ukrštanje, Mutacija** – primenjuje se nad izabranim jedinkama da bi se dobile nove jedinke

Osnovni genetski algoritmi

- Treba da se postupno približavamo rešenju
- Jedinke su reprezentovane nulama i jedinicama
- Postupak se odvija kroz generacije
- **Početna generacija:** slučajno generisan skup jedinki ili dobijen korišćenjem neke druge metode optimizacije
- **Funkcija cilja** (funkcija prilagođenosti, kvaliteta) – za svaku jedinku računa njen kvalitet
- **Selekcija** – iz jedne generacije se biraju jedinke koje će biti prenete u novu generaciju i iskorišćene za pravljenje novih jedinki
- **Ukrštanje, Mutacija** – primenjuje se nad izabranim jedinkama da bi se dobile nove jedinke
- Neke jedinke će biti bolje, a neke gore, ali se prosečan kvalitet popravlja

Osnovni genetski algoritmi

- Treba da se postupno približavamo rešenju
- Jedinke su reprezentovane nulama i jedinicama
- Postupak se odvija kroz generacije
- **Početna generacija:** slučajno generisan skup jedinki ili dobijen korišćenjem neke druge metode optimizacije
- **Funkcija cilja** (funkcija prilagođenosti, kvaliteta) – za svaku jedinku računa njen kvalitet
- **Selekcija** – iz jedne generacije se biraju jedinke koje će biti prenete u novu generaciju i iskorišćene za pravljenje novih jedinki
- **Ukrštanje, Mutacija** – primenjuje se nad izabranim jedinkama da bi se dobile nove jedinke
- Neke jedinke će biti bolje, a neke gore, ali se prosečan kvalitet popravlja
- **Zaustavljanje:** dostignut zadati kvalitet ili dostignut zadati broj generacija ...

Reprezentacija jedinki

- Da bude takva da je moguće lako primeti sve segmente algoritma

Reprezentacija jedinki

- Da bude takva da je moguće lako primeti sve segmente algoritma
- **Binarna reprezentacija**

Reprezentacija jedinki

- Da bude takva da je moguće lako primeti sve segmente algoritma
- **Binarna reprezentacija**
- n – broj bitova (dužina hromozoma)

Reprezentacija jedinki

- Da bude takva da je moguće lako primeti sve segmente algoritma
- **Binarna reprezentacija**
- n – broj bitova (dužina hromozoma)
- $[a, b]$ – interval kome rešenja pripadaju

Reprezentacija jedinki

- Da bude takva da je moguće lako primeti sve segmente algoritma
- **Binarna reprezentacija**
- n – broj bitova (dužina hromozoma)
- $[a, b]$ – interval kome rešenja pripadaju
- binarnom broju $x = 1100 \dots 001110101$ koji ima n bita odgovara broj

$$a + \frac{x}{2^n - 1} \cdot (b - a)$$

Reprezentacija jedinki

- Da bude takva da je moguće lako primeti sve segmente algoritma
- **Binarna reprezentacija**
- n – broj bitova (dužina hromozoma)
- $[a, b]$ – interval kome rešenja pripadaju
- binarnom broju $x = 1100 \dots 001110101$ koji ima n bita odgovara broj

$$a + \frac{x}{2^n - 1} \cdot (x - a)$$

- a realnom broju $x \in [a, b]$ odgovara binarni broj

$$\left[\frac{x - a}{b - a} \cdot (2^n - 1) \right]$$

Funkcija prilagođenosti

- Ocena kvaliteta jedinke

Funkcija prilagođenosti

- Ocena kvaliteta jedinke
- Izuzetno važna za efikasnost algoritma

Selekcija

- Nekada se jedinke biraju *slučajno*

Selekcija

- Nekada se jedinke biraju *slučajno*
- Nekada se biraju sa nekom *verovatnoćom*

Selekcija

- Nekada se jedinke biraju *slučajno*
- Nekada se biraju sa nekom *verovatnoćom*
- Najpopularnije strategije selekciju su: **turnirska** i **ruletska**

Selekcija

- Nekada se jedinke biraju *slučajno*
- Nekada se biraju sa nekom *verovatnoćom*
- Najpopularnije strategije selekciju su: **turnirska** i **ruletska**
- Nekada se može desiti da se izgubi neka kvalitetna jedinka.
Elitizam je varijacija procesa selekcije tako da se čuvaju najbolje jedinke.

Ruletska selekcija

- Ruletska selekcija je proces selekcije u kojem veće sanse da učestvuju u reprodukciji imaju *prilagođenije jedinke*.

Ruletska selekcija

- Ruletska selekcija je proces selekcije u kojem veće sanse da učestvuju u reprodukciji imaju *prilagođenije jedinke*.
- $f(i)$ – funkcija prilagođenosti za i -tu jedinku. N – broj jedinki. Verovatnoća izbora je

$$p_i = \frac{f(i)}{\sum_j^N f(j)}$$

Ruletska selekcija

- Ruletska selekcija je proces selekcije u kojem veće sanse da učestvuju u reprodukciji imaju *prilagođenije jedinke*.
- $f(i)$ – funkcija prilagođenosti za i -tu jedinku. N – broj jedinki. Verovatnoća izbora je

$$p_i = \frac{f(i)}{\sum_j^N f(j)}$$

- funkcija prilagođenosti mora biti *pozitivna* (ili transformisana u neku pozitivnu)

Ruletska selekcija

- Ruletska selekcija je proces selekcije u kojem veće sanse da učestvuju u reprodukciji imaju *prilagođenije jedinke*.
- $f(i)$ – funkcija prilagođenosti za i -tu jedinku. N – broj jedinki. Verovatnoća izbora je

$$p_i = \frac{f(i)}{\sum_j^N f(j)}$$

- funkcija prilagođenosti mora biti *pozitivna* (ili transformisana u neku pozitivnu)
- Može se desiti da se neka jedinka bira više puta

Turnirska selekcija

- k – veličnina turnira; p – verovatnoća

Turnirska selekcija

- k – veličnina turnira; p – verovatnoća
- Bira se k jedinki iz populacije i sortira po vrednosti funkcije prilagođenosti

Turnirska selekcija

- k – veličina turnira; p – verovatnoća
- Bira se k jedinki iz populacije i sortira po vrednosti funkcije prilagođenosti
- i -ta jedinka u sortiranom nizu bira se sa verovatnoćom $p(1 - p)^{i-1}$

Turnirska selekcija

- k – veličina turnira; p – verovatnoća
- Bira se k jedinki iz populacije i sortira po vrednosti funkcije prilagođenosti
- i -ta jedinka u sortiranom nizu bira se sa verovatnoćom $p(1 - p)^{i-1}$
- što je veća veličina turnira, veće su šanse da će lošije jedinke biti izabrane

Turnirska selekcija

- k – veličina turnira; p – verovatnoća
- Bira se k jedinki iz populacije i sortira po vrednosti funkcije prilagođenosti
- i -ta jedinka u sortiranom nizu bira se sa verovatnoćom $p(1 - p)^{i-1}$
- što je veća veličina turnira, veće su šanse da će lošije jedinke biti izabrane
- ako je $p = 1$ bira se najbolja jedinka u turniru

Turnirska selekcija

- k – veličina turnira; p – verovatnoća
- Bira se k jedinki iz populacije i sortira po vrednosti funkcije prilagođenosti
- i -ta jedinka u sortiranom nizu bira se sa verovatnoćom $p(1 - p)^{i-1}$
- što je veća veličina turnira, veće su šanse da će lošije jedinke biti izabrane
- ako je $p = 1$ bira se najbolja jedinka u turniru
- Efikasno se implementira na *paralelnim arhitekturama*

Ukrštanje

- Učestvuju dve jedinke – *roditelji*

Ukrštanje

- Učestvuju dve jedinke – *roditelji*
- izabre se *tačka prekida*

Primer: neka je tačka prekida 3

001 10110101 – prvi roditelj

011 11000010 – drugi roditelj

001 11000010 – dete1

011 10110101 – dete2

Ukrštanje

- Učestvuju dve jedinke – *roditelji*
- izabre se *tačka prekida*

Primer: neka je tačka prekida 3

001 10110101 – prvi roditelj

011 11000010 – drugi roditelj

001 11000010 – dete1

011 10110101 – dete2

- Može biti ukrštanja i sa *više tačaka prekida*.

Ukrštanje

- **Uniformno ukrštanje** – Svaki bit prvog roditelja se uzima sa *verovatnoćom* p , a Svaki bit drugog roditelja se uzima sa *verovatnoćom* $1 - p$.

Ukrštanje

- **Uniformno ukrštanje** – Svaki bit prvog roditelja se uzima sa *verovatnoćom* p , a Svaki bit drugog roditelja se uzima sa *verovatnoćom* $1 - p$.
- **Poliuniformno ukrštanje** – $\frac{\text{broj_razlicitih_bitova_kod_roditelja}}{2}$ - broj bitova koji se prenose na decu

Mutacija

- Menja jedan ili više slučajno odabranih gena

Mutacija

- Menja jedan ili više slučajno odabranih gena
- Verovatnoća mutacije je obično između 0.01 i 0.001

Mutacija

- Menja jedan ili više slučajno odabranih gena
- Verovatnoća mutacije je obično između 0.01 i 0.001
- Sprečava jedinke suviše slične.
Primer: ukoliko u jednoj generaciji sve jedinke imaju istu vrednost jednog gena, onda taj gen samo ukrštanjem nikada ne bi mogao da se promeni.

Mutacija

- Menja jedan ili više slučajno odabranih gena
- Verovatnoća mutacije je obično između 0.01 i 0.001
- Sprečava jedinke suviše slične.
Primer: ukoliko u jednoj generaciji sve jedinke imaju istu vrednost jednog gena, onda taj gen samo ukrštanjem nikada ne bi mogao da se promeni.
- Omogućavaju razmatranje novih delova prostora pretrage.

Mutacija

- Menja jedan ili više slučajno odabranih gena
- Verovatnoća mutacije je obično između 0.01 i 0.001
- Sprečava jedinke suviše slične.
Primer: ukoliko u jednoj generaciji sve jedinke imaju istu vrednost jednog gena, onda taj gen samo ukrštanjem nikada ne bi mogao da se promeni.
- Omogućavaju razmatranje novih delova prostora pretrage.
- *verovatnoća* == 1 – algoritam slučajne pretrage.

Mutacija

- Menja jedan ili više slučajno odabranih gena
- Verovatnoća mutacije je obično između 0.01 i 0.001
- Sprečava jedinke suviše slične.
Primer: ukoliko u jednoj generaciji sve jedinke imaju istu vrednost jednog gena, onda taj gen samo ukrštanjem nikada ne bi mogao da se promeni.
- Omogućavaju razmatranje novih delova prostora pretrage.
- *verovatnoća* == 1 – algoritam slučajne pretrage.
- *verovatnoća* == 0 – algoritam će verovatno brzo dospeti do nekog lokalnog ekstremuma

Zaustavljanje

- Pronađeno je rešenje koje zadovoljava unapred zadate (minimalne) kriterijume.

Zaustavljanje

- Pronađeno je rešenje koje zadovoljava unapred zadate (minimalne) kriterijume.
- Dostignut je zadati broj generacija.

Zaustavljanje

- Pronađeno je rešenje koje zadovoljava unapred zadate (minimalne) kriterijume.
- Dostignut je zadati broj generacija.
- Funkcija prilagođenosti je upotrebljena zadati broj puta.

Zaustavljanje

- Pronađeno je rešenje koje zadovoljava unapred zadate (minimalne) kriterijume.
- Dostignut je zadati broj generacija.
- Funkcija prilagođenosti je upotrebljena zadati broj puta.
- Vrednost prilagođenosti najboljih jedinki se u nekoliko generacija ne popravlja.

Zaustavljanje

- Pronađeno je rešenje koje zadovoljava unapred zadate (minimalne) kriterijume.
- Dostignut je zadati broj generacija.
- Funkcija prilagođenosti je upotrebljena zadati broj puta.
- Vrednost prilagođenosti najboljih jedinki se u nekoliko generacija ne popravlja.
- Kombinacija nekoliko uslova.

Primer:

- Vožnja automobila po stazi sa preprekama

Primer:

- Vožnja automobila po stazi sa preprekama
- napred, levo, desno

Primer:

- Vožnja automobila po stazi sa preprekama
- napred, levo, desno
- senzor za brzinu – *sporo* i *brzo*

Primer:

- Vožnja automobila po stazi sa preprekama
- napred, levo, desno
- senzor za brzinu – *sporo* i *brzo*
- senzor za prepreke – *levo, desno, jedno polje napred* i *dva polja napred*

Primer:

- Vožnja automobila po stazi sa preprekama
- napred, levo, desno
- senzor za brzinu – *sporo* i *brzo*
- senzor za prepreke – *levo, desno, jedno polje napred i dva polja napred*
- može da skreće samo ako se kreće sporo

Primer:

- Vožnja automobila po stazi sa preprekama
- napred, levo, desno
- senzor za brzinu – *sporo* i *brzo*
- senzor za prepreke – *levo, desno, jedno polje napred i dva polja napred*
- može da skreće samo ako se kreće sporo
- nema slepih krakova i grananja

Primer:

- Vožnja automobila po stazi sa preprekama
- napred, levo, desno
- senzor za brzinu – *sporo* i *brzo*
- senzor za prepreke – *levo, desno, jedno polje napred i dva polja napred*
- može da skreće samo ako se kreće sporo
- nema slepih krakova i grananja
- Poželjno: da ubrza kada su dva polja ispred slobodna, a da uspori kada nisu (i kada treba da skrene)

Primer:

Reprezentacija podataka:

- Razlikujemo *5 stanja* i *5 akcija*

Primer:

Reprezentacija podataka:

- Razlikujemo *5 stanja* i *5 akcija*
- Akcije:

Primer:

Reprezentacija podataka:

- Razlikujemo *5 stanja* i *5 akcija*
- Akcije:

0 – ne radi ništa

Primer:

Reprezentacija podataka:

- Razlikujemo *5 stanja* i *5 akcija*
- Akcije:
 - 0* – ne radi ništa
 - 1* – uspori

Primer:

Reprezentacija podataka:

- Razlikujemo *5 stanja* i *5 akcija*
- Akcije:
 - 0* – ne radi ništa
 - 1* – uspori
 - 2* – ubrzati

Primer:

Reprezentacija podataka:

- Razlikujemo *5 stanja* i *5 akcija*
- Akcije:
 - 0* – ne radi ništa
 - 1* – uspori
 - 2* – ubrzati
 - 3* – skrenuti levo

Primer:

Reprezentacija podataka:

- Razlikujemo *5 stanja* i *5 akcija*
- Akcije:
 - 0* – ne radi ništa
 - 1* – uspori
 - 2* – ubrzati
 - 3* – skrenuti levo
 - 4* – skrenuti desno

Primer:

Reprezentacija podataka:

- Stanja:

Primer:

Reprezentacija podataka:

- Stanja:

brzina – sporo, brzo

Primer:

Reprezentacija podataka:

- Stanja:

brzina – sporo, brzo

polje ispred – slobodno, prepreka

Primer:

Reprezentacija podataka:

- Stanja:

brzina – sporo, brzo

polje ispred – slobodno, prepreka

dva polja ispred – slobodno, prepreka

Primer:

Reprezentacija podataka:

- Stanja:

brzina – sporo, brzo

polje ispred – slobodno, prepreka

dva polja ispred – slobodno, prepreka

polje levo – slobodno, prepreka

Primer:

Reprezentacija podataka:

- Stanja:

brzina – sporo, brzo

polje ispred – slobodno, prepreka

dva polja ispred – slobodno, prepreka

polje levo – slobodno, prepreka

polje desno – slobodno, prepreka

Primer:

Reprezentacija podataka:

- Stanja:

brzina – sporo, brzo

polje ispred – slobodno, prepreka

dva polja ispred – slobodno, prepreka

polje levo – slobodno, prepreka

polje desno – slobodno, prepreka

- Za svako stanje imamo 2 mogućnosti – $2^5 = 32$ ukupan broj stanja

Primer:

Reprezentacija podataka:

- Stanja:
 - brzina* – sporo, brzo
 - polje ispred* – slobodno, prepreka
 - dva polja ispred* – slobodno, prepreka
 - polje levo* – slobodno, prepreka
 - polje desno* – slobodno, prepreka
- Za svako stanje imamo 2 mogućnosti – $2^5 = 32$ ukupan broj stanja
- Imaćemo hromozom koji ima 32 polja i na svakom polju ima neku vrednost sa intervala 0-4

Primer:

Reprezentacija podataka:

- Stanja:
 - brzina* – sporo, brzo
 - polje ispred* – slobodno, prepreka
 - dva polja ispred* – slobodno, prepreka
 - polje levo* – slobodno, prepreka
 - polje desno* – slobodno, prepreka
- Za svako stanje imamo 2 mogućnosti – $2^5 = 32$ ukupan broj stanja
- Imaćemo hromozom koji ima 32 polja i na svakom polju ima neku vrednost sa intervala 0-4
- Polje označava stanje, a vrednost, akciju u tom stanju

Primer:

Reprezentacija podataka:

- Stanja:

brzina – sporo, brzo

polje ispred – slobodno, prepreka

dva polja ispred – slobodno, prepreka

polje levo – slobodno, prepreka

polje desno – slobodno, prepreka

- Za svako stanje imamo 2 mogućnosti – $2^5 = 32$ ukupan broj stanja
- Imaćemo hromozom koji ima 32 polja i na svakom polju ima neku vrednost sa intervala 0-4
- Polje označava stanje, a vrednost, akciju u tom stanju
- $(10011)_2 = 19$ *hromozom*[19] = 2
stanje: brzo + prepreka levo + prepreka desno + slobodno napred + slobodno dva napred
akcija: ubrzati

Primer:

- **Ukrštanje:** jedna tačka prekida

Primer:

- **Ukrštanje:** jedna tačka prekida
- **Mutacija:** Sa malom verovatnoćom menjamo svaku poziciju, svakog hromozoma

Primer:

- **Ukrštanje:** jedna tačka prekida
- **Mutacija:** Sa malom verovatnoćom menjamo svaku poziciju, svakog hromozoma
- **Selekcija:** Ruletska

Primer:

Funkcija prilagođenosti:

- Koristeći dati hromozom, pokušaćemo da prođemo trku

Primer:

Funkcija prilagođenosti:

- Koristeći dati hromozom, pokušaćemo da prođemo trku
- Prelazak polja brzo nosi 2 poena

Primer:

Funkcija prilagođenosti:

- Koristeći dati hromozom, pokušaćemo da prođemo trku
- Prelazak polja brzo nosi 2 poena
- Prelazak polja sporo nosi 1 poen

Primer:

Funkcija prilagođenosti:

- Koristeći dati hromozom, pokušaćemo da prođemo trku
- Prelazak polja brzo nosi 2 poena
- Prelazak polja sporo nosi 1 poen
- Favorizujemo one koji su završili trku – na trenutani zbir dodaje se zbir svih polja