Bevezetés a lágy számítás módszereibe

Genetikus algoritmusok
Rekombináció, mutáció

Werner Ágnes

Villamosmérnöki és Információs Rendszerek Tanszék

e-mail: werner.agnes@virt.uni-pannon.hu

Rekombináció

- szelekció után keletkező szülő állomány
  - tartalmazhat ismétlődéseket
  - részben vagy egészben azonos lehet a populációval
- kettő-több szülő felhasználásával képez utódot (jellemzően kettőből egyet vagy kettőt)
- célja: a szülőkből minél jobb, újabb megoldások összeállítása az átvett, "örökölt" tulajdonságok alapján
- formáját befolyásolja a változók típusa és a problémák sajátosságai
- $P_r \approx 0,7$

Diszkrét rekombináció

- többféle változó típus esetén alkalmazható: egész, valós, bináris, sztring, szimbólumok
- a művelet csak néhány diszkrét értékkel dolgozhat
- két szülő változóiból véletlenszerűen képezzük az utódot
- jelölje $(x_1, x_2, \ldots, x_n)$, $(y_1, y_2, \ldots, y_n)$ és $(u_1, u_2, \ldots, u_n)$ a két szülő és az utód egyedet változóikkal együtt
- minden változónál képezzük az

  $u_i = ax_i + (1-a)y_i \quad i = 1, 2, \ldots, n$ műveletet, ahol $a \in \{0, 1\}$ véletlen szám és minden változó esetén újra generálásra kerül
- bővített változat

Egész és valós típusú változók rekombinációja

- **Köztes rekombináció**
  - $u_i = ax_i + (1-a)y_i \quad i = 1, 2, \ldots, n)$, ahol $a \in [-h, 1+h]$ véletlen szám és a minden változó esetén újra generálásra kerül
    a h értékét általában $0,5$-nek választjuk
  - bővített változat
- **Lineáris rekombináció**
  - $u_i = ax_i + (1-a)y_i \quad i = 1, 2, \ldots, n)$, ahol $a \in [-h, 1+h]$ véletlen szám és minden változó esetén ugyanazon a értéket alkalmazzuk
  - speciális változat $a = 0,5$

Bináris sztringek rekombinációja

- **Egypontos keresztezés**
  - két szülőből két utód
  - véletlenszerűen választunk keresztezési pontot az $\{1, 2, \ldots, L-1\}$ pozíciók közül
- **Többpontos keresztezés**
  - két szülőből két utód
  - n számú keresztezési pontot választunk
  - a kapott keresztezési pontokat növekvő sorrendbe rendezzük, majd a megfelelő, egymás után következő keresztezési pontok közti bitsorozatokat rendre más-más szülőtől választjuk

Bináris sztringek rekombinációja

- **Uniform keresztezés**
  - minden bit pozíción külön-külön döntjük el, melyik szülőtől választjuk a következő bit értékét
- **Keverő keresztezés**
  - mindkét szülőben a bit pozíciókat azonos módon, véletlenszerűen összekeverjük
  - egypontos keresztezést alkalmazunk
  - visszaállítjuk a bitpozíciók eredeti sorrendjét

Permutációk rekombinációja

az egyedek csak érvényes permutációk lehetnek

Uniform sorrend alapú rekombináció

2 szülőből 2 utódot állít elő és a szülők relatív sorrendjét örökíti át

1. előállít egy bit maszkot, amely minden pozíción véletlenszerűen 0 vagy 1 értéket tárol; azon pozíciókon, ahol 1 értéket tárol a bitmaszk, az 1. szülő megfelelő pozícióján lévő permutáció sorszámát átmásolja az 1. utód azonos pozíciójára
2. az 1. utód többi pozícióját üresen hagyja
3. a 2. utódot hasonlóan állítja elő, csak itt a 0 bitmaszk értékű pozíciókat veszi elő és a 2. szülő megfelelő pozícióján lévő permutáció sorszámát másolja át a 2. utód azonos pozícióira
4. a 2. utód többi pozícióját üresen hagyja (az eredmény egy köztes állapot, amelyben mindkét utódnál csak részben ismerjük a keresett permutációt)
5. az 1., majd a 2. utódban tölti fel a hiányzó helyeket, ehhez listát készít az 1. szülő azon sorszámairól, amelyek nem kerültek át az 1. utódba
6. a listát úgy rendezi, hogy minél több sorszámegyezést érjen el a 2. szülő azonos pozícióin
7. a kész lista elemeivel balról-jobbra haladva feltölti az 1. utód üres pozícióit, a 2. utód üres pozícióit hasonló módon tölti fel

Mutáció

A rekombináció nem alkalmas finom közelítések megvalósítására.
A mutáció az utód közvetlen környezetében keres jobb megoldásokat.
$P_m \approx 0,01$
Valós és egész típusú változók mutációja
különbség a mutációs lépés nagyságában
Schlierkamp-Voosen és Mühlenbein mutáció művelete:

az x_i változóból a $z_i = x_i \pm range_i * \delta$

a $+$ vagy $-$ előjelet $0,5$ valószínűséggel választjuk,
a $range_i$ az x_i változó szomszédsági környezetének szélességét jelöli,
a δ a pontosságot határozza meg, diszkrét vagy folytonos értékei lehetnek

Bináris típusú változók mutációja

az egyed egy bitsorozat, melynek egyes bitjeit mutációval változtatjuk

a mutáció egy x_i változónál a következő:

$$z_i = \begin{cases} x_i & , ha \;\; Rnd > P_m \\ 1 - x_i & , ha \;\; Rnd \leq P_m \end{cases}$$

Permutációk mutációja

- jelöljék az egyed változói a permutáció egyes pozícióin található értékeket,
- a mutáció ezt a sorrendet úgy változtatja meg, hogy néhány pozíción (változóban) más sorrendben helyezi el az értékeket,
  - beszúrás
  - csere
  - inverz
  - Scramble-mutáció

Visszahelyezés

A szelekció, rekombináció, mutáció műveletsorral minden generációban új utódokat kapunk.
Ezen utódokkal, vagy egy részükkel bővíteni kell a populációt, lecserélve velük a korábbi egyedek egy részét.
bevezetünk 2 rátát:

- utódképzési ráta
- visszahelyezési ráta
- relációk a ráták között:
  - $utodkepzesirata = visszahelyezesirata = 1$
  - $utodkepzesirata \leq visszahelyezesirata < 1$
  - $utodkepzesirata > visszahelyezesirata$

Elitizmus

Az EA ciklus kialakítása

- **stratégiai paraméterek megadása**
  - populáció mérete
  - a rekombináció alkalmazásának P_r valószínűsége
  - a mutáció alkalmazásának P_m valószínűsége
  - az utódképzési ráta értéke
  - a visszahelyezési ráta értéke
- **kezdő populáció kialakítása**
  - véletlenszerű előállítás
  - előző feladat eredményeinek felhasználása
  - korábbi eredmények módosított felhasználása

Az EA ciklus kialakítása

megállási feltétel

- maximális generációszám elérése
- maximális futási idő elérése
- adott idő alatt nem javul a megoldás minősége
- hasonlóak az egyedek
- előre adott érték megközelítése
- a populáció minősége megfelelő (mérőszámok):
  - a célfüggvény értékének standard szórása az aktuális generációban
  - a célfüggvény értékek átlagának és a legjobb értéknek az eltérése az aktuális generációban
  - a legjobb és a legrosszabb célfüggvény érték eltérése az aktuális generációban

Fitnesz kiértékelés

A problémák legtöbbjénél ismerünk egy **célfüggvényt**, amely az értelmezési tartomány, azaz a keresési tér pontjaihoz egy valós számot vagy vektort rendel.

Az esetek többségében a fitneszfüggvényt azonosnak választjuk a célfüggvénnyel.

Sokszor nincs célfüggvényünk és a fitneszfüggvény megfogalmazása a probléma megoldásának egyik fontos kulcseleme.

Konkrét formája függ a reprezentációtól.