

Modell projektek

C programozás

Zachar István

1. Evolúció természetes szelekcióval

Terület: Evolúció, mutáció, string manipuláció, valószínűség számítás

Probléma: A probléma Dawkins 1986-os, "evolúció természetes szelekcióval" modelljére épül. Jól demonstrálja a valószínűségeket, illetve azt, hogy kumulatív evolúcióval az alapvetően kis valószínűségű állapotok is könnyen előállhatnak. Dawkins mintájára szimuláljuk az evolúciót a következő karakterláncon: "METHINKSITISAWASEL". (Az idézet eredete: "Methinks it is like a weasel", Shakespeare Hamletjében Hamlet és Polonius nézi a felhőket, és Hamlet szerint az egyik felhő menyét alakúnak tűnik.) Az egyszerűség kedvéért tekintsünk el a szóközöktől és egyéb, nem-betű karakterektől. A szimuláció induljon egy tetszőleges random karaktersorozattal (a 26 betűből választva), és minden lépésben egy állandó kis valószínűséggel (*mutációs ráta*) egy betű változzon át egy másik betűvé (azaz önmagává nem változhat). A szimuláció akkor áll meg, amikor a kívánt karaktersort ("METHINKSITISAWASEL") elértük.

Kérdések

1. Hány ilyen 19 betű hosszúságú üzenet lehetséges?
2. Mekkora a valószínűsége, hogy legalább egy betű jó a 19-ből?
3. Mekkora a valószínűsége, hogy pontosan egy betű jó a 19-ből?
4. Hogyan függ a konvergenciaidő (azaz amíg megtaláljuk a jó karaktersort) a mutációs rátától?

Források

Richard Dawkins, 1994, A vak órásmester

John Maynard Smith, 1998, Evolutionary Genetics, Oxford University Press, p. 12.

2. Hamming távolság

Terület: temlátok, kombinatorika, string manipuláció

Probléma: Írjunk egy függvényt, amely tetszőleges (de azonos) hosszúságú szekvencia-párokat (karakter-tömböket) hasonlít össze és megállapítja a két szekvencia Hamming távolságát. Két azonos hosszú szekvencia Hamming távolsága a szekvenciák közti különbségek száma (azaz hány pozícióban nem egyeznek). A program elején válasszunk egy tetszőleges random 7 bázis hosszú szekvenciát (ez lesz a mesterkópia). Gyártsunk le 1000 random szekvenciát (7 hosszúak, 4 bázisból) és mérjük meg a Hamming távolságukat a mesterkópiától. Számoljuk meg, hogy hány szekvencia esik az egyes Hamming-osztályokba. Azok a szekvenciák, amelyek azonos Hamming-távolságra vannak a mesterkópiától, alkotnak egy Hamming-osztályt.

Kérdések:

1. Hány Hamming-osztály van N hosszúságú szekvenciák esetén?
2. Hány lehetséges szekvencia van (azaz mekkora a szekvenciater) N szekvenciahosszúság és 4 bázis esetén? És N hossz és B különféle bázis esetén?

Források: [Hamming-távolság wiki](#)

3. Életjáték

Terület: sejtautomaták, térben explicit modellezés, mátrixok

Probléma: Az **életjátékot** (*game of life*) John Conway, a Cambridge Egyetem matematikusa találta ki. A "játék" egy 2 dimenziós sejtautomata, az elnevezés arra utal, hogy egyszerű kezdeti feltételeket megadva is a térbeli mintázatok időfejlődése kis túlzással élőlényekhez hasonlatos komplexitású viselkedést mutat, miközben a mögöttes szabályok rendkívül egyszerűek. Szimuláljunk egy tetszőleges mátrixot (pl. 20x20-ast), amelyben minden cella 0 vagy 1 értéket vehet fel: 0 ha üres, 1 ha élő „sejt” foglalja el. Minden körben minden cellát kiértékelünk: a szomszédos 8 cella alapján a központi cella az új körben vagy marad ami volt, vagy az ellentétére változik. A kiértékelés szabályai:

1. A sejt *túléli* a kört (azaz ha 1 volt, 1 marad), ha pontosan két vagy három szomszédja van.
2. A sejt elpusztul (azaz 0-ra változik), ha kettőnél kevesebb (*elszigetelődés*), vagy háromnál több (*túlnépesedés*) szomszédja van.
3. Új sejt születik minden olyan cellában, melynek környezetében pontosan három sejt található.

Az eredménymátrixokat minden körben írjuk ki fájlba, és tetszőleges programmal (R, *Mathematica*, stb.) ábrázoltassuk!

Kérdések:

4. Mi történik, ha csupa egyesből álló mátrixból indítjuk a szimulációt? És mi történik, ha sakktábla-elrendezésből indítjuk?
5. Hogyan lehet megoldani, hogy a sejtér a szélein önmagába forduljon? Azaz ha valami kimegy baloldalt, a jobb szélén lépjen be; ami kimegy felül, az alul térjen vissza; stb.
6. Változik a szimuláció, ha szekvenciálisan frissítjük a cellákat (azaz pl. a bal felső saroktól a jobb alsóig, jobbra és lefelé) vagy minden körben egy random választott cellát frissítünk?

Források:

4. Rekombináció

Terület: szexuális szaporodás, string manipuláció

Probléma: Írjunk egy függvényt, amely elvégzi két tetszőleges hosszúságú és összetételű DNS szekvencia rekombinációját. Ha a két szekvencia (egyszerűsítve) pl. xxxXXXXX és YYYYYY, akkor az eredmény a következő két string legyen: xxxyyyyy és YYXXXXX. Generáljunk egy 100 fős random szekvencia-populációt. Véletlenszerűen válasszunk a populációból két szekvenciát és rekombináljuk őket egy-egy belső töréspont mentén (azaz ne a szekvencia végénél hasítsunk). Az eredmény-szekvenciákkal írjuk felül a populációból választott párost. Ismételjük a rekombinációt random választott szekvenciapárokon sokszor egymás után.

Kérdések:

1. Ha azonos hosszú szekvenciákból indulunk ki, akkor sok rekombinációs lépés után mi lesz az átlagos szekvenciahossz?
2. Ha tetszőleges hosszúságú szekvenciákból indulunk ki (pl. minimum 4 és maximum 20 bázis között), akkor sok rekombinációs lépés után (azaz egyensúlyban) mi lesz az átlagos szekvenciahossz?

3. Mi lesz az egyensúlyi szekvenciák bázis-eloszlása? Ehhez segítség, ha uniform szekvenciákból indulunk ki, azaz a populáció csak AAA..., CCC..., TTT... GGG... szekvenciákat tartalmaz.

Források:

5. Interaktív labirintus

Terület: algoritmusok, mátrixok, 2D random bolyongás

Probléma: Írjunk programot, amely egy véletlenszerű, kétdimenziós $X*Y$ méretű labirintust generál. Egy $X*Y$ méretű mátrix minden cellája egy szobának felel meg a labirintusban, amelynek 0-4 ajtaja lehet a szomszédos 4 szobához. Ha segít, tároljuk az ajtókat egy másik mátrixban. Gondoskodjunk róla, hogy a labirintusunk átjárható legyen a bal felső saroktól ($lab[0][0]$) a jobb alsóig ($lab[X][Y]$). Írjunk egy függvényt, amely megkeresi az átvezető utat a labirintusban, és az érintett szobákhoz kiírja, hogy ott milyen szörny lakik és milyen kincs van elrejtve. Randomizáljuk a szörnyek és kincsek eloszlását is labirintusban.

Kérdések:

1. Tetszőleges X és Y esetén átlagosan milyen hosszú (hány szobát ill. Szörnyet érint) egy bolyongás a labirintusban?
2. Ha a szörnyfajok száma S , a különféle kincsek típusa meg K , mennyi az esélye, hogy egy bolyongás során olyan szobába botlunk, aminek az összetételével (szörny és kincs) már találkoztunk e bolyongás során?

+1. Algoritmikus vers

Terület: string manipuláció, költészet

Probléma: Írjunk lírai randomizáló programot: legyen a vers N soros, a rímképlete legyen választhatóan AABB vagy ABAB (vagy más), és a sorvégi rímeket tetszőleges, előre megadott összezsengő szavak listájából válassza ki a program! Figyeljünk rá, hogy a szótagszám és esetleg a verslábak is kövessék a klasszikus költészet szabályait.

Források: “– Írjon kiberotikus verset! [...] Legfeljebb öt sor legyen, de szóljon szerelemről, árulásról és halálról, a néger kérdésről és a nimfomániáról, legyen benne a bonyolult női lélek extrém konfliktushelyzetben bekövetkező meghasonlásának ábrázolása, a középkori feudális viszonyok és erkölcsök maró bírálata, rímeljen, és minden szó k betűvel kezdődjön!” [Stanisław Lem: Kiberiáda - Első Pótutazás avagy az Elektrubadúr.](#)