

Oddaja Celtrinega izziva – Avtomati

Repozitorij

<https://github.com/t4c1/avtomati>

Izziv sem razvijal v veji master, končen program pa je v veji final.

Navodila za zagon

Program se zažene s Python-om 2.7, z ukazom:

```
python avtomati.py <url streznika skupaj s številko primera>
```

npr:

```
python avtomati.py http://celtra-jackpot.com/4
```

Cilji, pristop k reševanju, rezultati

Najprej sem napisal ogrodje za komunikacijo s strežnikom (razred *AvtomatiBase*), tako da je bilo razvijane algoritma popolnoma ločeno od načina dostopa do avtomatov, kar mi je kasneje olajšalo testiranje in konfiguracijo. Algoritem sem zastavil tako, da se za vsak poteg izbere avtomat, ki ima največje število točk. Točke so uteženo povprečje osnovnih faktorjev.

Prvi faktor je povprečna dosedajšnja nagrada na avtomatu, izračunana, kot razmerje uspešnih in vseh potegov. Ker se v polovici primerov lahko verjetnost uspešnih potegov spreminja sem navadno povprečje zamenjal s tekočim povprečjem, tako da imajo zadnji potegi največji vpliv. Dodal pa sem tudi povprečje le zadnjih nekaj potegov.

Naslednji je faktor nenatančnosti, ki je na začetku zelo velik, ko pa se število vseh potegov na avtomatu povečuje, pa se približuje 0. Uporaben je zato, da ne bi algoritem dobil dovolj dobrega rezultata iz enega avtomata in nato ne bi preizkusil tudi drugih. Kasneje se je ta izkazal za nepotrebnega saj ga je zamenjal naslednji, neimenovani faktor.

Ker se lahko verjetnost uspeha na »slabih« avtomatih poveča, sem dodal še en faktor, ki je odvisen od tega, koliko potegov je že minilo od zadnjega potega na tem avtomatu. Tako se na vsakih nekaj potegov na boljših avtomatih zgodi tudi en na vsakem izmed slabših. Da se lahko algoritem hitreje prilagodi povečanju verjetnosti uspeha na »slabem« avtomatu, se s tem, ko na določenemu avtomatu potegov že dalj časa ni bilo poveča tudi utež tekočega povprečja.

Nato je bilo potrebno še določiti uteži posameznih faktorjev in določene konstante za izračun faktorjev. Ker je povezovanje na strežnik počasno in sem nameraval narediti veliko število testov, sem napisal svojo kodo za testiranje, ki določa, kdaj je poteg uspešen. S testi sem nato za svoje primere določil optimalne uteži in konstante.

Iz končne verzije sem izbrisal kodo za testiranje in nekatere izmed faktorjev, ki so se v testiranju izkazali za nepotrebne (navadno povprečje, povprečje zadnjih nekaj elementov in drugo tekoče povprečje, faktor nenatančnosti).