

# Uporaba umetne inteligence za zmago igre 2048

Seminarska naloga pri predmetu Osnove umetne inteligence

Sebastian Mežnar

23. november 2018

2048 je preprosta računalniška igra v kateri združuješ skupaj ploščice z istimi števili in iz njih narediš ploščo, na kateri je njun seštevek. Cilj te igre je narediti ploščico z številom 2048. V nalogi sem to igro poiskoval to igro zmagati s pomočjo hevrističnih preiskovalnih algoritmov.

## 0.1 Uvod

Če je naloga zasnovana tako, da vključuje analizo izbranih podatkov, v tem razdelku opišeš, kakšni so ti podatki in navedeš nekaj osnovnih statističnih lastnosti teh podatkov. Slednje vključujejo velikost podatkov (na primer število primerov, število in vrsto atributov), delež manjkajočih podatkov, opis in porazdelitev vrednosti ciljnih spremenljivk, in podobno. Če si podatke pridobil sam, tu opišeš, na kakšen način, kje in kako.

## 1 O igri

2048 je odprtokodna igra, dostopna na internetnem naslovu <https://github.com/gabrielecirulli/2048>.

Cilj te igre je sestaviti ploščico s številom 2048 iz manjših ploščic. Igralec lahko v vsaki potezi ploščice potisne v eno izmed štirih smeri (gor, dol, levo, desno), ob tem pa se zaletijo ob sosednje in se združijo, če je na dveh sosednjih isto število. Po vsaki potezi se na igralnem polju pojavi nova ploščica z številom 2 oziroma 4. Igra se konča, ko igralec ne more opraviti nobene poteze več.

Rezultat se v računa tako, da se vsakič, ko igralec dve ploščici združi, rezultatu prišteje vrednost nove ploščice. Igra se najpogosteje igra na polju velikosti 4X4. Za polje take velikosti lahko naredimo največ ploščico z vrednostjo 131972, največji možen rezultat pa je 3,932,156.

## 2 Rezultati.

V tem poglavju podaš rezultate s kratkim (enoodstavčnim) komentarjem. Rezultate lahko prikažeš tudi v tabeli (primer je tabela 1).

Odstavke pri pisanju poročila v LaTeX-u ločiš tako, da pred novim odstavkom pustiš prazno vrstico. Tudi, če pišeš poročilo v kakšnem drugem urejevalniku, morajo odstavki biti vidno ločeni. To narediš z zamikanjem ali pa z dodatnim presledkom.

Tabela 1: Atributi in njihove zaloge vrednosti.

ime spremenljivke	definijsko območje	opis
cena	[0, 500]	cena izdelka v EUR
teža	[1, 1000]	teža izdelka v dag
kakovost	[slaba—srednja—dobra]	kakovost izdelka

Podajanje rezultati naj bo primerno strukturirano. Če ima naloga več podnalog, uporabi podpoglavja. Če bi želel poročati o rezultatih izčrpno in pri tem uporabiti vrsto tabel ali grafov, razmisli o varianti, kjer v tem poglavju prikažeš in komentiraš samo glavne rezultate, kakšne manj zanimive detajle pa vključite v prilogo (glej prilogi A in B).

### 3 Izjava o izdelavi domače naloge.

Domačo nalogo in pripadajoče programe sem izdelal sam.

## Priloge

### A Podrobni rezultati poskusov.

Če je rezultatov v smislu tabel ali pa grafov v nalogi mnogo, predstavi v osnovnem besedilu samo glavne, podroben prikaz rezultatov pa lahko predstaviš v prilogi. V glavnem besedilu ne pozabi navesti, da so podrobni rezultati podani v prilogi.

### B Programska koda.

Za domače naloge bo tipično potrebno kaj sprogramirati. Če ne bo od vas zahtevano, da kodo oddate posebej, to vključite v prilogo. Čisto za okus sem tu postavil nekaj kode, ki uporablja Orange (<http://www.biolab.si/orange>) in razvrščanje v skupine.

```
import random
import Orange

data_names = ["iris", "housing", "vehicle"]
data_sets = [Orange.data.Table(name) for name in data_names]

print "%10s_3s_3s_3s" % (" ", "Rnd", "Div", "HC")
for data, name in zip(data_sets, data_names):
    random.seed(42)
    km_random = Orange.clustering.kmeans.Clustering(data, centroids = 3)
    km_diversity = Orange.clustering.kmeans.Clustering(data, centroids = 3,
        initialization=Orange.clustering.kmeans.init_diversity)
    km_hc = Orange.clustering.kmeans.Clustering(data, centroids = 3,
        initialization=Orange.clustering.kmeans.init_hclustering(n=100))
    print "%10s_3d_3d_3d" % (name, km_random.iteration, \
```

```
km_diversity.iteration, km_hc.iteration)
```