

# Igranje igre mice korišćenjem genetskih algoritama

Djordje Pantelić, Nikola Dimić

Januar 2019

## Sažetak

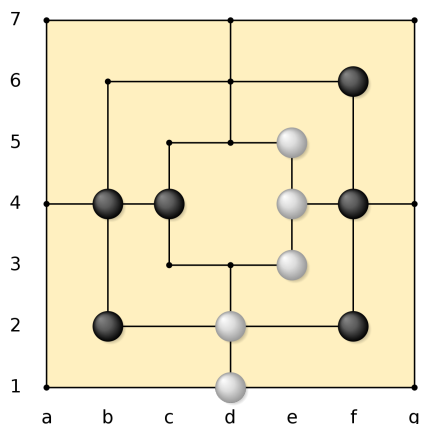
Ovaj rad se bavi prikazom i implementacijom genetskih algoritama u svrhu rešavanja problema nalaženja optimalne strategije za igranje igre Mice (eng. Nine men's morris). Rad je pisan u sklopu kursa Računarska inteligencija koji se drži na Matematičkom fakultetu u Beogradu.

## Sadržaj

<b>1</b>	<b>Uvod</b>	<b>2</b>
1.1	Strategija . . . . .	2
1.1.1	Prva faza igre . . . . .	3
1.1.2	Druga faza igre . . . . .	3
1.1.3	Treća faza igre . . . . .	4
<b>2</b>	<b>Opis rešenja</b>	<b>5</b>
2.1	Genetski algortam . . . . .	6
<b>3</b>	<b>Eksperimentalni rezultati</b>	<b>8</b>
<b>4</b>	<b>Zaključak</b>	<b>9</b>

# 1 Uvod

Mica je strateška igra koja datira još iz vremena rimske imperije. Igra se sastoji od tri faze.[1] U prvoj fazi svaki od igrača postavlja po 9 figura na tablu koja ima 24 polja.



Slika 1: Tabla

Cilj igre je postaviti 3 figure na pozicijama koje su spojene linijama na tabli. Ovakva konfiguracija figura naziva se mica. Igrač koji uspe da kreira micu odnosi jednu protivničku figuru sa table koja se ne nalazi u mici osim u slučaju da takvih figura nema. U drugoj fazi igre se već postavljene figure pomeraju na neku od slobodnih susednih pozicija, dok se u trećoj fazi igre figure pomeraju na bilo koju slobodnu poziciju na tabli. Igrač igra po pravilima treće faze igre ukoliko mu je ostalo 3 figure na tabli. Igra se završava u trenutku kada je nekom od igrača ostalo 2 figure na tabli ili više nema legalnih poteza.

U ovom radu biće reči o optimizaciji strategija koje se koriste u igri Mice, korišćenjem genetskih algoritama. Strategija koja će biti optimizovana zapravo predstavlja koeficijente funkcije evaluacije u minimaks algoritmu.

## 1.1 Strategija

Strategija za igranje igre opisana u radu Nine Men's Morris: Evaluation Functions, predstavljena je sa 17 koeficijenata heuristika koji su predstavljeni celim brojevima. Ovi koeficijenti podeljeni su po fazama igre. Svaki koeficijent predstavljen je svojom pozicijom u nizu strategije.

### 1.1.1 Prva faza igre

0. Heuristika koja predstavlja indikator upravo zatvorene mice. Opseg vrednosti koeficijenta ove heuristike predstavljen je sledećim intervalom [9,25]
1. Heuristika predstavlja razliku broja figura u micama.  
Opseg vrednosti koeficijenta ove heuristike predstavljen je intervalom [5,25]
2. Heuristika predstavlja razliku broja blokiranih figura.  
Blokirana figura je ona koja nema ni jednu slobodnu susednu poziciju. Opseg vrednosti koeficijenta ove heuristike predstavljen je intervalom [1,5]
3. Heuristika predstavlja razliku broja figura igrača na tabli.  
Opseg vrednosti koeficijenta ove heuristike predstavljen je intervalom [5,35]
4. Heuristika predstavlja razliku broja 2-konfiguracija igrača na tabli.  
2-konfiguracija predstavlja konfiguraciju od 2 figure kojoj se može dodati jedna figura tako da se formira mica.  
Opseg vrednosti koeficijenta ove heuristike predstavljen je intervalom [5,30]
5. Heuristika predstavlja razliku broja 3-konfiguracija.  
3 konfiguracija predstavlja par 2-konfiguracija koje imaju jednu zajedničku figuru. Igrač u ovoj konfiguraciji ima dve pozicije koje zatvaraju micu.  
Opseg vrednosti koeficijenta ove heuristike predstavljen je intervalom [5,25]

### 1.1.2 Druga faza igre

6. Heuristika koja nalik na prvi element u strategiji predstavlja indikator upravo zatvorene mice. Ovaj parametar može imati različitu važnost u prvoj i drugoj fazi pa su nam potrebne dve vrednosti.  
Opseg vrednosti koeficijenta ove heuristike predstavljen je sledećim intervalom [15,40]
7. Heuristika koja nalik na drugi element u strategiji predstavlja razliku broja figura u micama.  
Opseg vrednosti koeficijenta ove heuristike predstavljen je sledećim intervalom [10,30]
8. Heuristika koja nalik na treći element u strategiji predstavlja razliku broja blokiranih figura.  
Opseg vrednosti koeficijenta ove heuristike predstavljen je sledećim intervalom [7,20]
9. Heuristika koja nalik na četvrti element u strategiji predstavlja razliku broja figura igrača na tabli.  
Opseg vrednosti koeficijenta ove heuristike predstavljen je sledećim intervalom [10,35]

10. Heuristika koja predstavlja razliku između broja figura koje u jednom potezu formiraju micu.  
Opseg vrednosti koeficijenta ove heuristike predstavljen je sledećim intervalom [5,30]
11. Heuristika koja predstavlja razliku između broja trakalica. Trakalica predstavlja micu čija jedna figura kada je pomerenata zatvara drugu micu.  
Opseg vrednosti koeficijenta ove heuristike predstavljen je sledećim intervalom [20,55]
12. Heuristika koja predstavlja indikator zadovoljenosti uslova kraja igre.  
Opseg vrednosti koeficijenta ove heuristike predstavljen je sledećim intervalom [5000,25000]

### 1.1.3 Treća faza igre

13. Heuristika koja nalik na četvrti element predstavlja razliku broja 2-konfiguracija. Opseg vrednosti koeficijenta ove heuristike predstavljen je intervalom [5,25]
14. Heuristika koja nalik na peti element predstavlja razliku broja 3-konfiguracija. Opseg vrednosti koeficijenta ove heuristike predstavljen je intervalom [5,30]
15. Heuristika koja nalik na prvi element u strategiji predstavlja indikator upravo zatvorene mice. Opseg vrednosti koeficijenta ove heuristike predstavljen je sledećim intervalom [15,45]
16. Heuristika koja predstavlja proveru da li su ispunjeni uslovi završetka igre. Opseg vrednosti koeficijenta ove heuristike predstavljen je sledećim intervalom [5000,10000]

U radu je opisan proces korišćenja genetskih algoritama radi dobijanja najbolje moguće strategije tj jedinke za igranje igre mice. Najbolje jedinke predstavljene su u sledećoj tabeli

	Phase 1	Phase 2	Phase 3
E32	18 26 1 6 12 7 14 43 10 8 7 42 1086	10 1 16 1190	
E34	18 26 1 6 21 7 42 28 16 8 24 19 949 23 18 5 1096		
E30	18 26 1 6 12 7 14 43 10 1 30 40 958 6 32 7 1041		
E46	14 37 4 14 20 2 16 43 11 8 7 42 1086	10 1 16 1190	

Slika 2: Najbolje jedinke/strategije [3]

## 2 Opis rešenja

Cilj projekta je da korišćenjem genetskih algoritama pronađemo što bolje vrednosti koeficijenata koje će predstavljati strategiju za igranje igre mice. Usled nedostataka prethodno opisanog modela, dodate su nove dve heuristike.

1. Heuristika predstavlja razliku broja legalnih poteza druge faze igre dva protivnika.  
Opseg vrednosti koeficijenta ove heuristike predstavljen je intervalom [1,6]
2. Heuristika koja nalik na sedmi element strategije predstavlja razliku broja legalnih poteza druge faze igre dva protivnika.  
Opseg vrednosti koeficijenta ove heuristike predstavljen je intervalom [1,3]

```
16 25 2 10 11 11 2 30 17 11 16 7 52 1 6200 12 11 32 7811
18 7 2 14 8 21 6 26 16 20 17 16 49 3 8865 20 22 24 6239
13 25 4 31 27 7 1 17 12 12 11 9 37 1 8682 21 17 18 9690
17 11 1 14 12 15 4 36 19 18 11 22 38 1 9062 14 5 30 8835
```

Slika 3: Primer jedinki sa dodatim heuristikama

Vrednost funkcije cilja genetskog algoritma predstavlja broj poena koje određena strategija osvoji protiv unapred zadatih protivnika. Poeni se dodeljuju na sledeći način

broj poena	slučaj u kom se dodeljuju poeni
7	pobeda igrača
2	pobeda igrača sudijskom odlukom
0	nerešen rezultat
-2	poraz igrača sudijskom odlukom
-7	poraz igrača

Tabela 1: Opis tabele player\_stats.csv

Pobeda ili poraz sudijskom odlukom predstavlja ishod partije u kojoj su igrači usli u bezizlaznu poziciju ali je jedan od njih u tom trenutku imao više odnosno manje figura. Pozicija se definiše bezizlazna ukoliko igrači naizmenično igraju ista dva poteza.

## 2.1 Genetski algoritam

Primenom genetskog algoritma opisanog narednim pseudokodom dobijene su ciljne jedinke.

---

**Algorithm 1:** Genetski algoritam

---

```
Input: lista 8 kontrolnih protivnika (jedinki)
Output: lista najboljih 100 jedinki
/* Inicijalno se napravi 100 slučajno generisanih jedinki koje
   predstavljaju inicijalnu populaciju gde je svaka jedinka jedna
   strategija. Zatim se računa fitnes funkcija (igraju turnir protiv
   sudija). Sve jedinke se posle sortiraju po vrednosti fitnes funkcije */
1 population ← initPopulation()
2 population ← findFitness(population)
3 population ← sort(population)
4 i = 0
5 while i < 10 do
    /* bira se 50 jedinki koje će ući u ukrštanje i ostaviti potomstvo.
       Bira se 10 najboljih (elitizam) i 40 ruletskom selekcijom. */
6 eliteSelect ← eliteSelection(population, 10)
7 wheelSelect ← wheelSelection(population, 40)
8 selection ← eliteSelect + wheelSelect
    /* Od dobijenih 50 formiramo slučajnim izborom 25 parova jedinki koje
       ostavljaju potomstvo. Za ukrštanje koristi se metoda 4 point
       crossover. Zatim se nad svakim dobijenim potomkom primeni operator
       mutacije sa 8% šanse da će svaki gen jedinke biti izmenjen za 1 a da
       pritom ne izađe iz opsega. */
9 pairs ← generatePairs(selection)
10 for (unit1, unit2) ∈ pairs do
11     child1, child2 ← 4PointCrossOver(unit1, unit2)
12     child1 ← mutate(child1)
13     child2 ← mutate(child2)
14     children.add(child1)
15     children.add(child2)
    /* 50 novih jedinki se dodaje u prethodnu populaciju od 100 i ukoliko ima
       eventualnih duplikata, oni se odstrane. Zatim se, za one jedinke
       koje nemaju izračunatu vrednost fitnes funkcije, izračuna fitnes
       funkcija. Ukoliko imamo manje od 150 jedinki, populacija će biti
       dopunjena nasumičnim jedinkama za koje će takođe biti izračunata
       fitnes funkcija. */
16 population ← removeDuplicates(population + children)
17 population ← calculateFitness(population)
    /* Jedinke se zatim sortiraju i 100 najboljih jedinki se prebacuje u
       narednu generaciju. */
18 sortedPopulation ← sort(population)
19 population ← eliteSelection(sortedPopulation, 100)
20 i ++
    _
```

---

Ovaj proces se ponavljao 10 puta ili do pojave konvergencije. Kako evaluacija fitnes funkcije nije konstantna odnosno neće uvek za istu jedinku imati istu vrednost, broj iteracija je podeljen na segmente od 10 iteracija. Ovo doprinosi tome da će se faktor slučajnosti maltene eliminisati i da će najbolje jedinke biti zaista najbolje.

```

36 [11, 16, 3, 24, 14, 24, 5, 31, 13, 8, 30, 19, 31, 1, 7454, 24, 5, 19, 9500]
36 [10, 17, 3, 23, 14, 23, 5, 31, 13, 8, 30, 19, 30, 1, 8268, 24, 14, 40, 9500]
36 [11, 16, 3, 22, 14, 24, 5, 31, 13, 8, 30, 19, 32, 1, 7454, 24, 5, 19, 9501]
36 [11, 16, 3, 23, 14, 24, 5, 31, 13, 8, 30, 19, 32, 1, 8269, 24, 10, 40, 9501]
36 [9, 16, 3, 22, 14, 23, 5, 31, 13, 11, 25, 26, 31, 1, 5194, 25, 13, 23, 7042]
36 [10, 16, 3, 22, 14, 24, 5, 31, 13, 8, 30, 19, 32, 1, 7454, 23, 13, 39, 9502]
36 [11, 16, 3, 22, 14, 23, 5, 30, 13, 8, 30, 19, 31, 1, 7455, 24, 10, 40, 9501]
=====
Best unit: 10 16 3 22 14 24 5 31 13 8 30 19 31 1 7456 16 14 18 9500
Fitness: 41
Worst unit: 11 16 3 22 14 23 5 30 13 8 30 19 31 1 7455 24 10 40 9501
Fitness: 36
=====

```

Slika 4: Deo rezultata genetskog algoritma

Ovaj algoritam je pokrenut na dva računara protiv različitih protivnika, kako bi bio izbegnut slučaj da su inicijalni protivnici bili loše strategije. Zatim su najbolje strategije iz oba skupa upoređene i odabrana je najbolja.

### 3 Eksperimentalni rezultati

Najbolja jedinka dobijena prethodno opisanim procesom upoređena je sa najboljim jedinkama iz gore pomenutog rada. To je urađeno na sledeći način:

Najpre su obe jedinke odigrale protiv 100 nasumično odabranih protivnika. Najbolja jedinka iz rada je ostvarila rezultat od -106 poena dok je naša najbolja jedinka ostvarila rezultat od 296 poena. Ovo se može pripisati činjenici da strategija predložena u pomenutom radu ne uzima u obzir dve dodatne heuristike koje su nasumične jedinke imale u vidu.

```
10 10 2 6 9 25 6 15 17 8 21 28 48 3 5779 23 27 31 5949
18 26 1 6 12 7 0 14 43 10 8 7 42 0 1086 10 1 16 1190
```

Slika 5: Najbolja jedinka iz rada i najbolja jedinka dobijena ovim istraživanjem

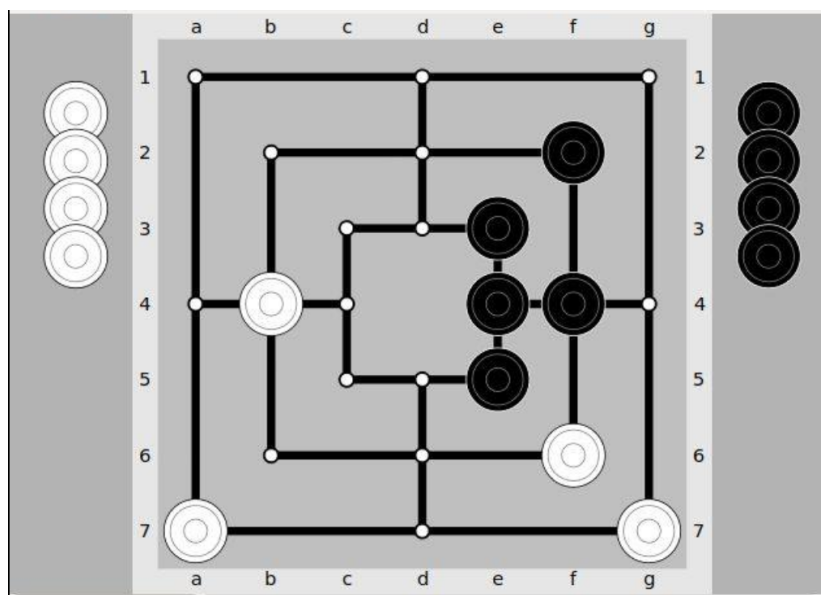
Strategija dobijena u ovom radu upoređena je i direktno protiv najbolje dobijene strategije iz prethodno navedenog rada, i tu je naša strategija bila bolja u svih četrdeset partija koliko su ove dve jedinke odigrale i ostvarila 80 poena.

Najbolja strategija takođe je testirana i protiv engine-a za micu iz paketa *morris* koji se nalazi u oficijalnim repozitorijumima većine linux distribucija, a moguće ga je preuzeti i sa sledećeg linka. Strategija priložena u ovom radu je pobedila.

Move	White	Black
1	b4	f4
2	a7	f2
3	f6	e4
4	g4	e3
5	g7	e5xg4
6	g4	d5
7	c5	d7
8	d6	b2
9	g1xb2	b6
10	a7-a4	e3-d3
11	c5-c4xe4	d3-e3
12	g1-d1	f4-e4xd1
13	a4-a7	e3-d3
14	a7-a4xe4	f2-f4
15	a4-a7	d3-e3
16	a7-a4xe3	e5-e4
17	a4-a7	d5-e5
18	d6-d5	b6-d6
19	g4-g1	e4-e3
20	a7-a4xf4	e3-d3
21	g7-g4	d3-d2
22	g1-d1	e5-e4
23	a4-a7	e4-f4
24	a7-a4xd6	d2-g7
25	d5-d6	f4-b6
26	a4-a1	b6-d2
27	g4-g1xd2	+++

Slika 6: Rekonstrukcija partije





Slika 7: Trenutno stanje na tabli u desetom potezu

## 4 Zaključak

Pokazalo se da je dodatno uvođenje heuristika kao i razdvojeno treniranje modela genetskim algoritmom bilo ključno u ostvarivanju znatno boljih rezultata prilikom optimizacije zadatog problema.

Za dodatno unapređenje treba razmisliti o uvođenju dodatnih heuristika i unapređenju genetskog algoritma za pronalaženje najboljih koeficijenata. [2]

## Literatura

- [1] Ralph Gasser. *Solving Nine Men's Morris*. 1996.
- [2] Gábor E. Gévay and Gábor Danner. *Calculating Ultra-Strong and Extended Solutions for Nine Men's Morris, Morabaraba, and Lasker Morris*. 2015.
- [3] Stefan Holban Simona-Alexandra Petcu. *Nine Men's Morris: Evaluation Functions*. 2008.