Iterativni izračun Nashevega ravnovesja v matričnih igrah

Eva Babnik
Mentor: prof. dr. Sergio Cabello Justo

Fakulteta za matematiko in fiziko

2. junij 2022

Cilji naloge in uporabljeno programsko okolje

- Implementacija dveh iterativnih metod, ki vrneta vrednost igre in optimalno strategijo v matričnih igrah z ničelno vsoto za dva igralca.
- Analiza konvergence implementiranih metod.
- Preprosta aplikacija, ki iterativno izračuna rešitev igre.
- *Python*, programski jezik *R*.

Problem

- Matrika izplačil $(a_{i,j})$, $i = 1, \cdots, m$, $j = 1, \cdots, n$.
- 1.igralec izbere eno izmed m vrstic z verjetnostjo x_i, 2. igralec pa enega izmed n stolpcev z verjetnostjo y_j.
- Veljati mora:

$$x_i \geq 0, \tag{1}$$

$$\sum x_i = 1, \tag{2}$$

$$y_i \geq 0, \tag{3}$$

$$\sum y_i = 1. \tag{4}$$

- Vrednost igre je definirana kot

$$v = \min_j \sum_i a_{ij} x_i = \max_i \sum_j a_{ij} y_j,$$

pri čemer (x, y), ki rešita zgornjo enakost predstavljata rešitev igre.

Metoda I

Definirajmo:

$$A_i = [a_{1i}, \cdots, a_{mi}, \underbrace{0, \cdots, 0}_{n-\text{komponent}}, -1],$$

$$A_{0i} = [\underbrace{1, \cdots, 1}_{m-\text{komponent}}, \underbrace{0, \cdots, 0}_{n-\text{komponent}}, 0]$$

$$\text{za } i = 1, \cdots, n$$

in

$$A_i = [\underbrace{0, \cdots, 0}_{m-\text{komponent}}, -a_{i-n,1}, \cdots, -a_{i-n,n}, 1],$$

$$A_{0i} = [\underbrace{0, \cdots, 0}_{m-\text{komponent}}, \underbrace{1, \cdots, 1}_{n-\text{komponent}}, 0]$$

$$\text{za } i = n+1, \cdots, m+n.$$

Metoda I

Definirajmo še rešitev igre $z^* = [x^*, y^*, v^*]$, ki mora poleg prej naštetih pogojev, ustrezati še:

$$A_i \cdot z^* \geq 0 \text{ za } i = 1, \cdots, m + n. \tag{5}$$

Metoda I

- Izberemo poljuben začetni vektor $z^{(1)}$, ki zadošča 1, 2, 3, 4 in 5 .
- Predpostavimo, da smo prišli na k-ti korak in dobili vektor $z^{(k)}$, ki ustreza 1, 2, 3 in 4. Če velja tudi 5, je $z^{(k)}$ rešitev igre in smo zaključili. Sicer pa:
  - Najdemo tak j_k, da velja $A_{j_k} \cdot z^{(k)} \leq A_i \cdot z^{(k)} \ \ \forall \ \ i = 1, \cdots, m+n$.
  - $\bar{z}^{(k+1)} = z^{(k)} + \alpha B_{j_k} + \beta B_{0j_k}$, kjer je

$$\alpha = -z^{(k)} \cdot B_{j_k} [1 - \cos^2 \theta_{j_k}]^{-1},$$

$$\beta = b_{0j_k} - [z^{(k)} + \alpha B_{j_k}] \cdot B_{0j_k},$$

$$b_{0j_k} = \frac{1}{(A_{0j_k} \cdot A_{0j_k})^{1/2}},$$

$$B_{j_k} = \frac{A_{j_k}}{(A_{j_k} \cdot A_{j_k})^{1/2}},$$

$$B_{0j_k} = \frac{A_{0j_k}}{(A_{0j_k} \cdot A_{0j_k})^{1/2}}$$

in

$$\cos \theta_{j_k} = \frac{A_{0j_k} \cdot A_{j_k}}{(A_{0j_k} \cdot A_{0j_k})^{1/2} (A_{j_k} \cdot A_{j_k})^{1/2}}.$$

Metoda I

- Predpostavimo, da velja $j_k < (n+1)$.
- Če $A_i \cdot z^* \geq 0 \ \forall i = 1 : (m+n)$:
  - $z^{k+1} = \bar{z}^{k+1}$
- Sicer:
  - Vse negativne x-komponente vektorja $\bar{z}^{k+1}$ nastavimo na 0.
  - Izračunamo vsote $\bar{x}_i^{(k+1)} + \frac{\sum_{i=1}^{r} \bar{x}_i^{(k+1)}}{m-r}$ za $i = r+1, \cdots, m$, kjer so $\bar{x}_1^{(k+1)}, \cdots, \bar{x}_r^{(k+1)}$, $r < m$ negativne komponente vektorja $\bar{z}^{(k+1)}$.
  - Za vsak tak i, za katerega je vsota negativna: $x_i^{(k+1)} = 0$.
  - Če nobena vsota ni negativna, lahko tvorimo preostanek vektorja $z^{(k+1)}$.
  - Če so nekatere vsote za $i = r+1, \cdots, r+s$ negativne, ponovno izračunamo vsote $\bar{x}_i^{(k+1)} + \frac{\sum_{i=1}^{r+s} \bar{x}_i^{(k+1)}}{m-(r+s)}$ za $i = r+s, \cdots, m$ in postopek ponavljamo, dokler nobena od vsot ni nenegativna.

Metoda I

- Predpostavimo, da za $i = 1, \cdots, t$ velja, da je $\bar{x}_i^{(k+1)} \leq 0$ ali pa, da je $\bar{x}_i^{(k+1)}$ tak, da je zanj katera od zgoraj definiranih vsot negativna.
- Tvorimo preostanek vektorja $z^{(k+1)}$:

$$x_1^{(k+1)} = \cdots = x_t^{(k+1)} = 0,$$

$$x_i^{(k+1)} = \bar{x}^{(k+1)} + \frac{\sum_{i=1}^{t} \bar{x}_i^{(k+1)}}{m-t} \text{ za } i = t+1, \cdots, m,$$

$$y_j^{(k+1)} = \bar{y}_j^{(k+1)} \text{ za } j = 1, \cdots, n,$$

$$v^{(k+1)} = \bar{v}^{(k+1)}.$$

Metoda II

- 1. igralec igra poljubno čisto strategijo x_{i_1}.
- Definiramo vektor akumulativnih vsot $A^{(1)} = [a_1^{(1)}, \cdots, a_n^{(1)}]$ $(A^{(1)} = i_1$-ta vrstica $A)$.
- 2. igralec igra čisto strategijo y_{j_1} (za j_1 velja: $a_{j_1}^{(1)} = \min\{a_1^{(1)}, \cdots, a_n^{(1)}\}$).
- Definiramo vektor akumulativnih vsot $B^{(1)} = [b_1^{(1)}, \cdots, b_m^{(1)}]$, $(B^{(1)} = j$-ti stolpec matrike).
- 1. igralec igra čisto strategijo x_{i_2} (za i_2 velja: $b_{j_2}^{(1)} = max\{b_1^{(1)}, \cdots, b_m^{(1)}\}$).
- 1. igralec vektorju $A^{(1)}$ prišteje i_2 - to vrstico matrike, zatem pa 2. igralec igra na enak način kot prej.
- Postopek ponavljamo.

Metoda II

Zaporedji $\frac{\sum_{n=1}^{k} x_{in}}{k}$, $\frac{\sum_{n=1}^{k} y_{in}}{k}$ ali konvergirata k optimalni strategiji x^*, y^*.

$v_1 = \max_{k=1,2,\cdots} \frac{\min_{i\in(1,\cdots,n)} a_i^{(k)}}{k}$,

$v_2 = \min_{k=1,2,\cdots} \frac{\max_{i\in(1,\cdots,m)} b_i^{(k)}}{k}$.

Analiza konvergence, časovna zahtevnost in rezultati

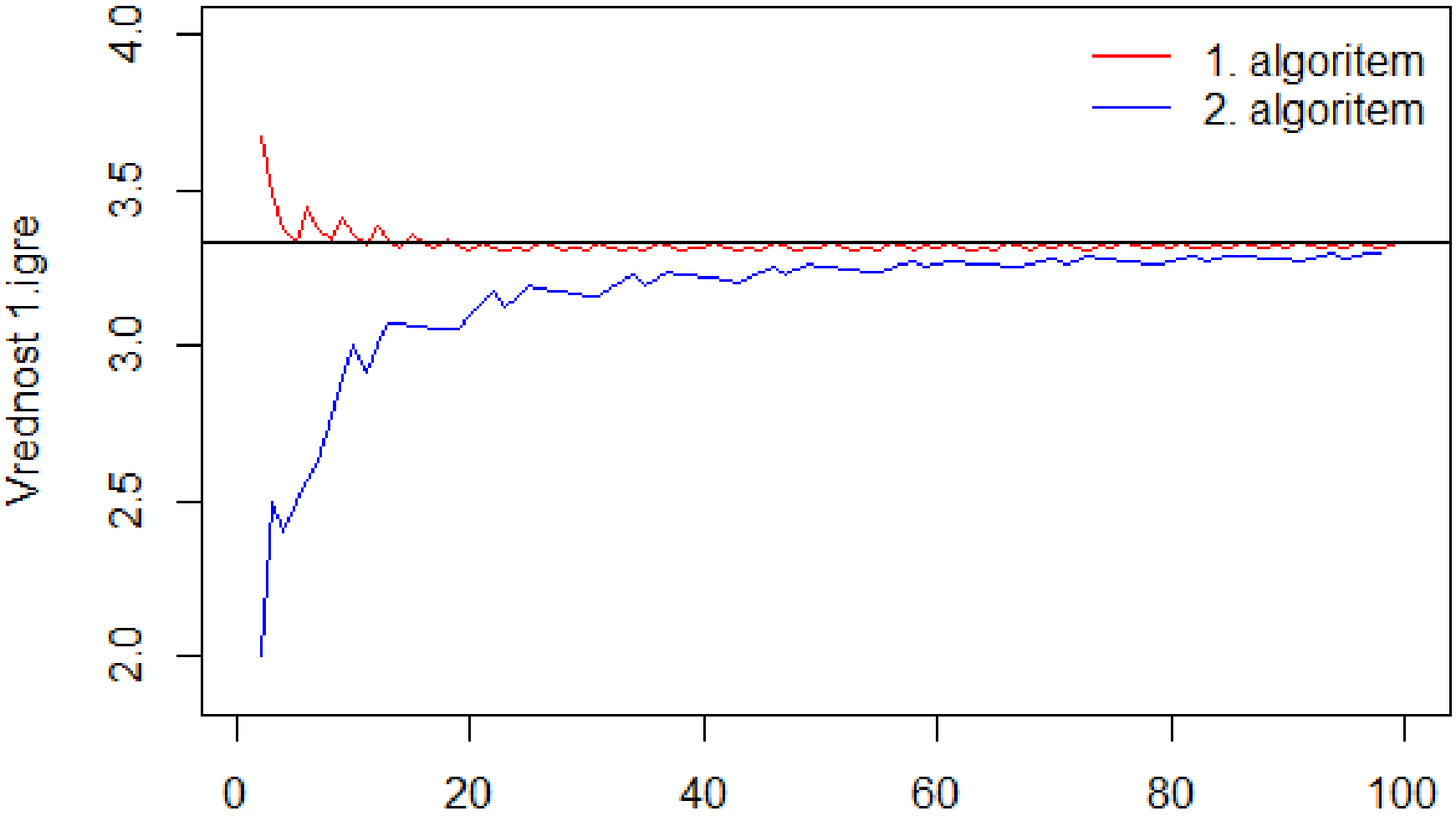
- Časovna zahtevnost prvega algoritma: $\mathcal{O}(k(m^2 \vee n^2))$.
- Časovna zahtevnost drugega algoritma: $\mathcal{O}(k)$.
- Meritve opravljene s prenosnim računalnikom s procesorjem 2.40GHz Intel(R) Core(TM) i5-6300U CPU in 8GB spomina.
- Število ponovitev poskusov: 10.

Analiza konvergence in časovne zahtevnosti za prvo igro

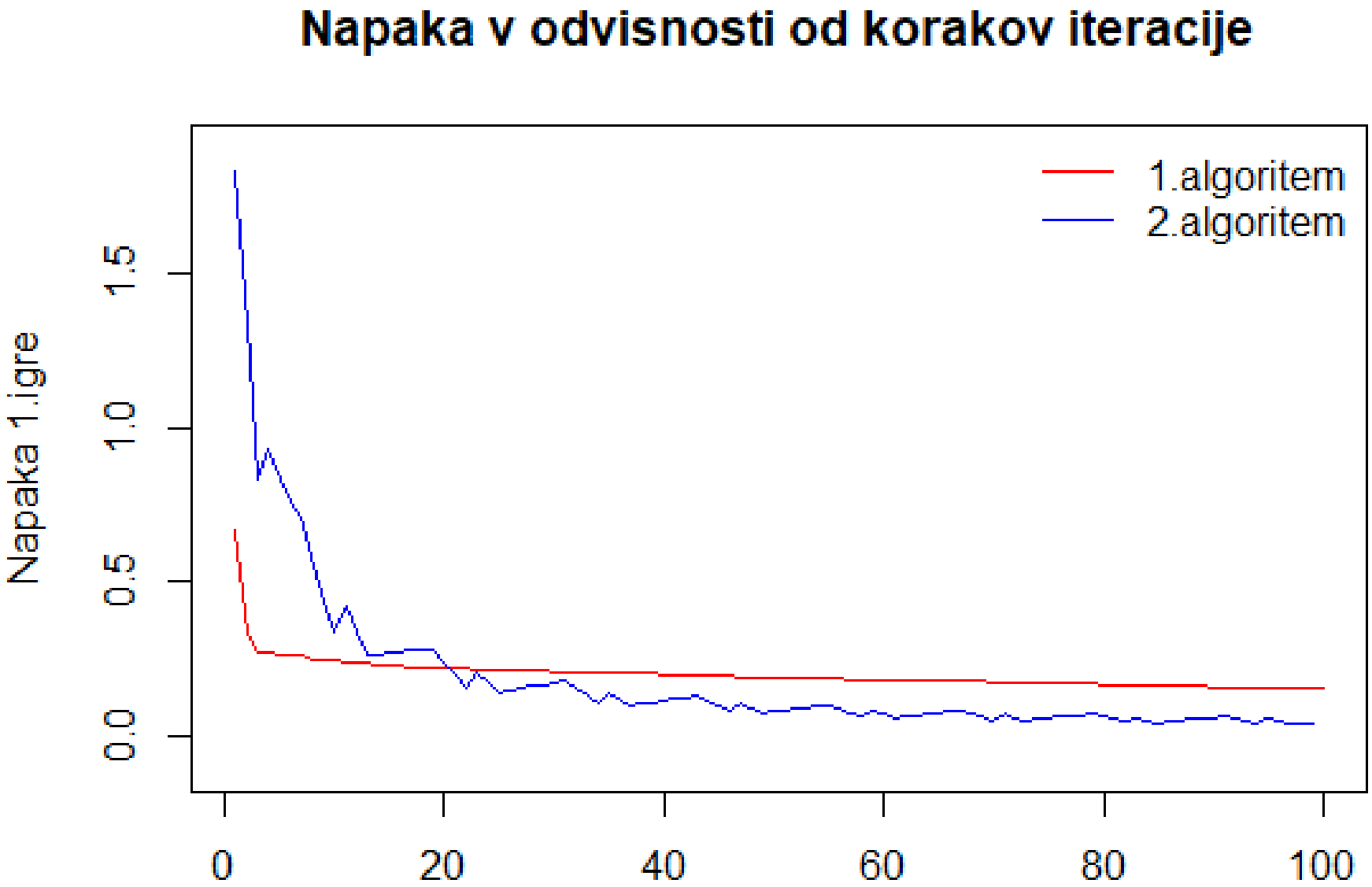
$$\begin{bmatrix} 4 & 3 \\ 2 & 4 \\ 5 & 2 \end{bmatrix}$$

$v^* = \frac{10}{3}$,
$x^* = [\frac{2}{3}, \frac{1}{3}, 0]$, $y^* = [\frac{1}{3}, \frac{2}{3}]$.

Vrednost prve igre v odvisnosti od števila iteracij



Napaka algoritmov $||z^* - z^k||_\infty$ za prvo matrično igro



Analiza konvergence in časovne zahtevnosti za prvo igro

	1. metoda				2. metoda			
	čas	v	x	y	čas	v	x	y
k = 1	0.0020	3	0.33333 0.33333 0.33333	0 1	0.0009	1.5	0.5 0.5 0	0 0.5
k = 2	0.0020	3.49020	0.33333 0.33333 0.33333	0.5 0.5	0.0014	2	0.66667 0.33333 0	0.33333 0.33333
k = 10	0.0155	3.31150	0.34199 0.45804 0.19997	0.34425 0.65575	0.0020	3	0.45454 0.45454 0	0.27273 0.63636
k = 50	0.0156	3.33100	0.47646 0.40436 0.11918	0.33100 0.66900	0.0030	3.25490	0.64706 0.33333 0.01961	0.35294 0.62746
k = 100	0.0369	3.33147	0.51447 0.39017 0.09536	0.33146 0.66854	0.0049	3.28713	0.66337 0.32673 0.00990	0.33663 0.65347
k = 500	0.0953	3.33301	0.64108 0.34289 0.01603	0.33302 0.66670	0.0156	3.32335	0.66467 0.33333 0.00200	0.33134 0.66667
k = 1000	0.1207	3.33330	0.66391 0.33436 0.00173	0.33330 0.66670	0.0312	3.33290	0.66633 0.33257 0.00010	0.33367 0.66533
k = 15000	2.4360	3.33333	0.66667 0.33333 0	0.33333 0.66667	0.4397	3.33304	0.66656 0.33338 0.00007	0.33338 0.66656

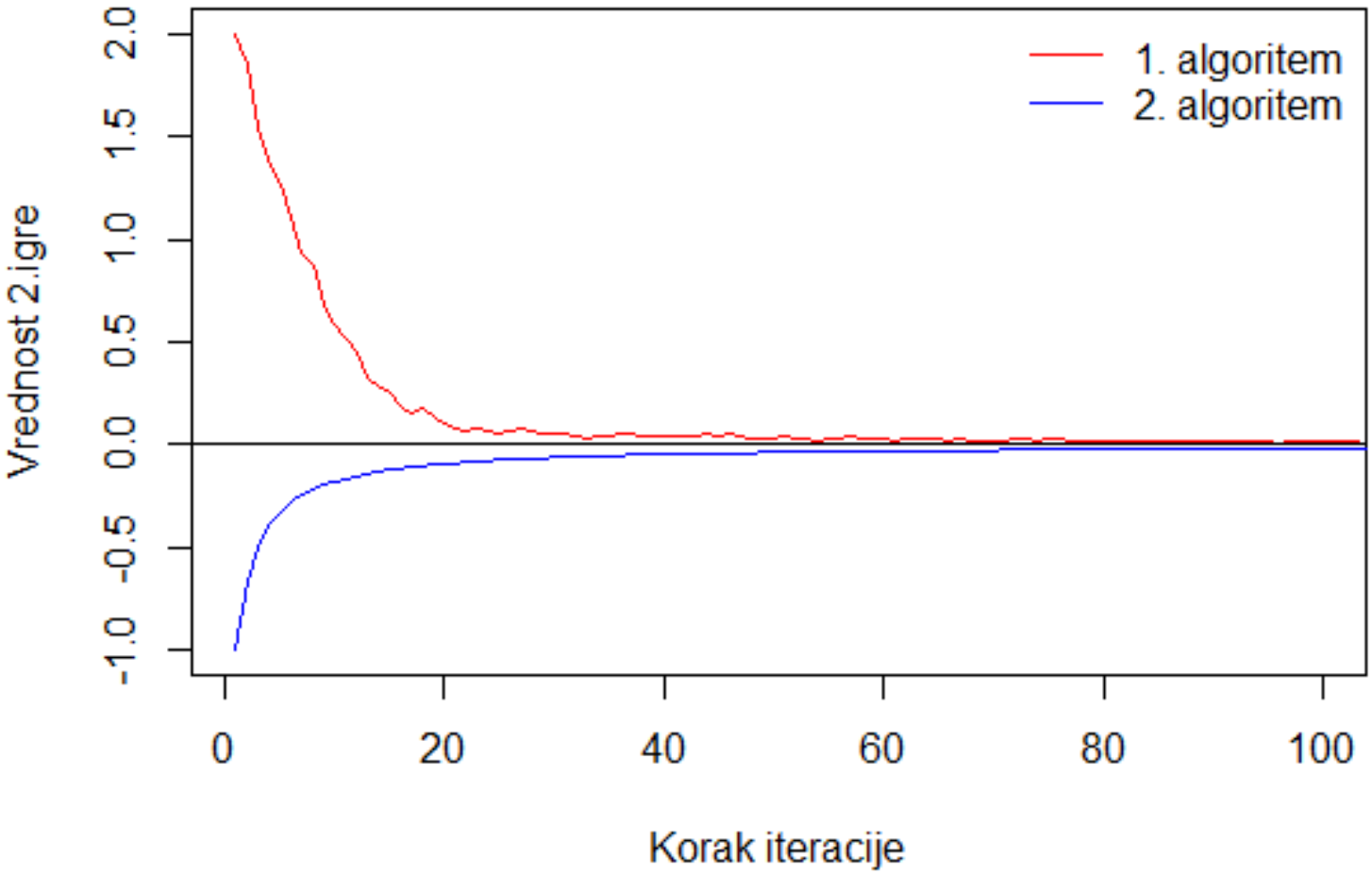
Tabela: Primerjava rezultatov za 1. igro

Analiza konvergence in časovne zahtevnosti za drugo igro

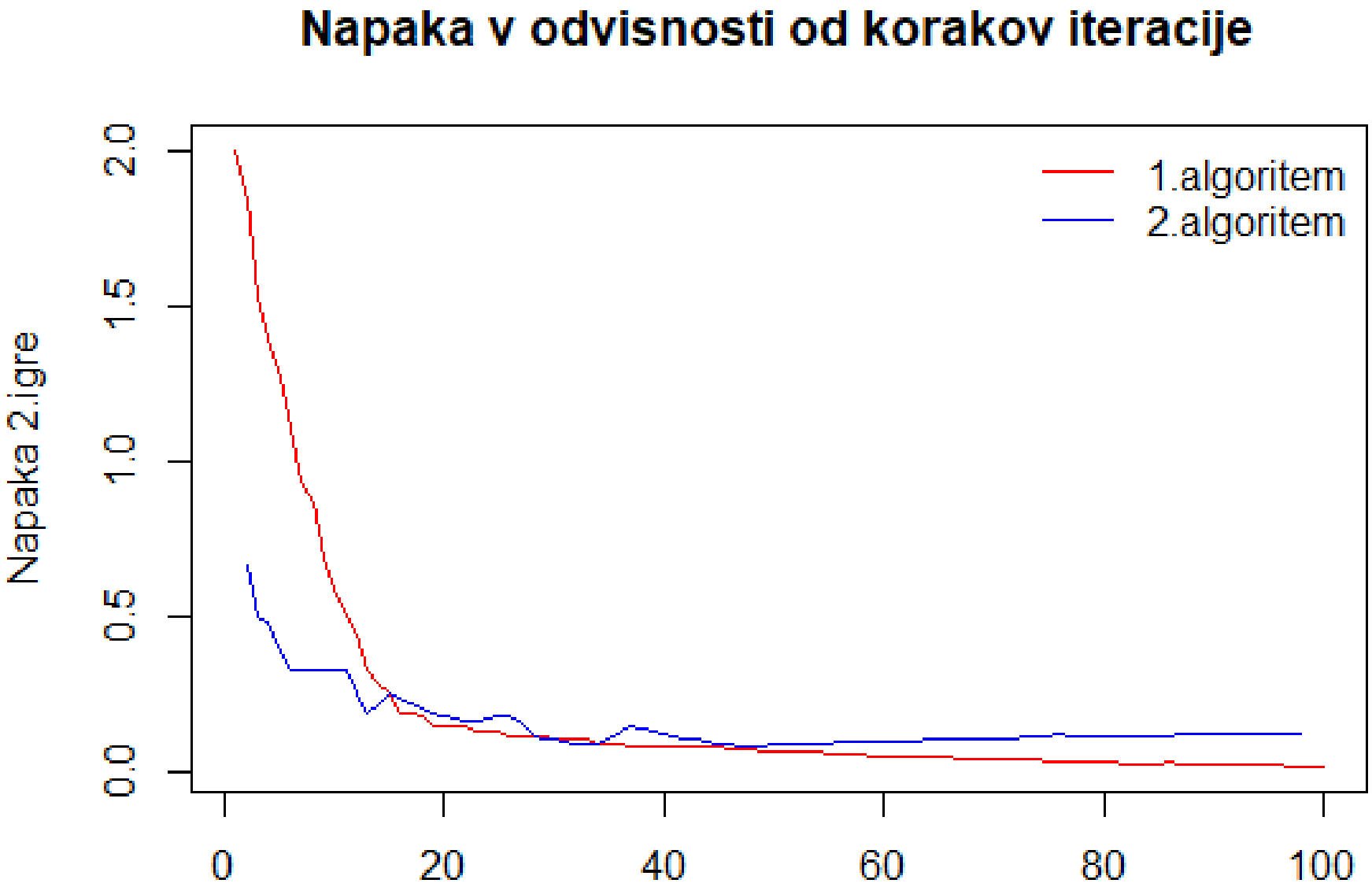
$$\begin{bmatrix} 0 & -1 & -2 & 0 & 0 & 2 & 1 \\ 1 & 0 & 0 & 4 & 0 & 0 & -1 \\ 2 & 0 & 0 & -1 & 1 & 0 & -1 \\ 0 & -4 & 1 & 0 & 0 & -1 & 1 \\ 0 & 0 & -1 & 0 & 0 & -3 & 1 \\ -2 & 0 & 0 & 1 & 3 & 0 & -1 \\ -1 & 1 & 1 & -1 & -1 & 1 & 0 \end{bmatrix}$$

$v^* = 0,$
$x^* = [1/9, 1/9, 15/90, 5/90, 15/90, 5/90, 1/3],$
$y^* = [1/9, 1/9, 15/90, 5/90, 15/90, 5/90, 1/3].$

Vrednost druge igre v odvisnosti od števila iteracij



Napaka algoritmov $||z^* - z^k||_\infty$ za prvo matrično igro



Analiza konvergence in časovne zahtevnosti za drugo igro

	1. metoda				2. metoda			
	čas	v	x	y	čas	v	x	y
k = 1	0.00365	1.84615	0.01282 0.16667 0.16667 0 0.16667 0.16667 0.32051	0.14286 0.14286 0.14286 0.14286 0.14286 0.14286 0.14286	0.00165	-1	0 0 0.5 0 0 0 0.5	0.5 0 0 0 0 0 0
k = 2	0.00489	0.73931	0.21520 0 0 0.20238 0.36904 0 0.21337	0.14286 0.14286 0.14286 0.14286 0.14286 0.14286 0.14286	0.00270	-0.66667	0 0.33333 0.33333 0 0 0 0.33333	0.33333 0 0 0.33333 0 0 0
k = 10	0.00319	0.53636	0.06687 0.22339 0.24321 0 0.21376 0 0.25274	0.14286 0.14286 0.14286 0.14286 0.14286 0.14286 0.14286	0.00575	-0.18182	0.18182 0.27273 0.09091 0.09091 0.27273 0 0.09091	0.09091 0.27273 0.09091 0.09091 0 0.36364
k = 50	0.01561	0.02724	0.11101 0.10108 0.16020 0.06888 0.16563 0.04690 0.34627	0.10855 0.14050 0.14894 0.05297 0.16978 0.11520 0.26410	0.00418	-0.03922	0.11765 0.09804 0.27451 0.07843 0.07843 0.03922 0.31373	0.11765 0.13725 0.17647 0.05882 0.07843 0.07843 0.33333
k = 100	0.03798	0.00560	0.11286 0.11280 0.16732 0.05262 0.16899 0.05386 0.33153	0.11161 0.13054 0.14982 0.05716 0.16783 0.07269 0.31035	0.00557	-0.01980	0.10891 0.09901 0.15842 0.03960 0.17822 0.08911 0.32673	0.12871 0.11881 0.18812 0.04951 0.15842 0.03960 0.30693
k = 500	0.18201	0.00092	0.111015 0.111365 0.166570 0.055421 0.166630 0.055414 0.333585	0.11136 0.11839 0.16141 0.05547 0.16639 0.05588 0.33111	0.03899	-0.00399	0.09980 0.14371 0.16567 0.05589 0.16567 0.05190 0.31737	0.12575 0.15369 0.13573 0.04591 0.14970 0.06387 0.32335
k = 1000	0.30498	0.00021	0.111174 0.111040 0.166691 0.055660 0.166655 0.055518 0.333259	0.11109 0.11311 0.16518 0.05562 0.16665 0.05559 0.33277	0.04474	-0.00199	0.11289 0.11089 0.17183 0.04695 0.16484 0.05594 0.33666	0.11588 0.11189 0.17083 0.04595 0.17383 0.04895 0.33167
k = 15000	4.83584	3.1e-17	0.11111 0.11111 0.16667 0.05556 0.16667 0.05556 0.33333	0.11111 0.11111 0.16667 0.05556 0.16667 0.05556 0.33333	0.58264	-0.00013	0.11312 0.11112 0.16732 0.05466 0.16299 0.05520 0.33558	0.10752 0.11179 0.16739 0.05786 0.16545 0.05446 0.33544