

System obliczający wyniki wyborów dla uogólnienia systemu k-Borda

Tomasz Kasprzyk, Daniel Ogiela, Jakub Stępak

Akademia Górniczo-Hutnicza
Wydział Informatyki, Elektroniki i Telekomunikacji
Katedra Informatyki

Projekt realizowany pod opieką
dr. hab. inż. Piotra Faliszewskiego

26 stycznia 2017

Definicja wyborów

Wybory to para $E = (C, V)$, gdzie $C = \{c_1, c_2, \dots, c_m\}$ to zbiór kandydatów, a $V = (v_1, v_2, \dots, v_n)$ to ciąg wyborców. Każdy wyborca posiada swoje *preferencje*, które są ciągiem kandydatów w porządku od najbardziej preferowanego przez danego wyborcę do najmniej preferowanego. Ponadto dana jest liczba k , będąca wielkością wybieranego komitetu.

Przykładowe wybory

Wybory filmów

$C = \{\text{komedia, horror, film akcji, dramat, science fiction}\}$

$V = (\text{Anna, Jan, Piotr, Paweł})$

Anna: ¹dramat > ²komedia > ³film akcji > ⁴horror > ⁵science fiction

Jan: ¹science fiction > ²komedia > ³dramat > ⁴film akcji > ⁵horror

Piotr: ¹horror > ²dramat > ³film akcji > ⁴komedia > ⁵science fiction

Paweł: ¹science fiction > ²film akcji > ³komedia > ⁴horror > ⁵dramat

$k = 2$

Punktacja Borda

Niech v będzie głosem nad zbiorem kandydatów C . Punkty przyporządkowane każdemu kandydatowi $c \in C$ w v wynoszą $\|C\| - pos_v(c)$, gdzie $pos_v(c)$ to pozycja kandydata c w v .

Funkcja Borda

$$\beta(i) = m - i, \quad \text{gdzie } m = \|C\|$$

Preferencje wyborcy

$$v_1 : \overset{m-1}{c_1} > \overset{m-2}{c_2} > \overset{m-3}{c_3} > \overset{m-4}{c_4} > \dots$$

Ciąg pozycji

Dla wybranego komitetu S i danego wyborcy v definiujemy ciąg $pos_v(S)$ jako posortowany ciąg pozycji, które zajmują kandydaci z S w preferencjach wyborcy v .

Niech $S = \{c_1, c_3, c_5, c_6\}$

Preferencje wyborcy

$$v_1 : \overset{1}{c_5} > \overset{2}{c_3} > \overset{3}{c_1} > \overset{4}{c_2} > \overset{5}{c_6} > \overset{6}{c_4} > \dots$$

$$pos_{v_1}(S) = (1, 2, 3, 5)$$

Oznaczenie wartości funkcji satysfakcji

$$f(i_1, \dots, i_k)$$

Norma ℓ_p

Niech $x_1, x_2, \dots, x_n \in \mathbb{R}, p \in \mathbb{N}$

Norma ℓ_p

$$\ell_p(x_1, x_2, \dots, x_n) = \sqrt[p]{x_1^p + x_2^p + \dots + x_n^p}$$

$$\begin{aligned}\ell_1 &\equiv + \\ \ell_\infty &\equiv \max\end{aligned}$$

System ℓ_p – Borda

Funkcja satysfakcji ℓ_p – Borda

$$f_{\ell_p}(i_1, i_2, \dots, i_k) = \ell_p(\beta(i_1), \beta(i_2), \dots, \beta(i_k))$$

Funkcja satysfakcji k-Borda (gdy $p = 1$)

$$f_{k-Borda}(i_1, \dots, i_k) = \beta(i_1) + \dots + \beta(i_k)$$

Funkcja satysfakcji Chamberlina-Couranta (gdy $p \rightarrow \infty$)

$$f_{CC}(i_1, \dots, i_k) = \beta(i_1)$$

Konto użytkownika

Election Computing System

Your elections

Logged as john.

Log out

Election Computing System

Logged in as john. Click on "Your elections" to start

System for Computing Election Results According to Generalized k-Borda System.

Engineering thesis created on the Faculty of Computer Science, Electronics and Telecommunications, AGH University of Science and Technology in Cracow

Supervisor: Piotr Faliszewski, PhD

Authors:

- Tomasz Kasprzyk
- Daniel Ogiela
- Jakub Stępak

Source code

Source code is available on [Github](#).

[Documentation](#) (in Polish) is available on Github as well.

Metody wprowadzania danych

some_elections

committee size: 8

You should add voters and candidates!



Load data from file (.soc PrefLib format)



Generate election from normal distribution



Draw chart

```
1 4
2 1,Dirty Dancing
3 2,Maid in Manhattan
4 3,Shrek (Full-screen)
5 4,Father of the Bride
6 1814,1814,24
7 491,3,4,1,2
8 397,3,1,4,2
9 222,3,4,2,1
10 138,1,3,4,2
11 101,3,1,2,4
12 87,3,2,4,1
13 64,4,3,1,2
14 58,1,3,2,4
15 47,3,2,1,4
16 45,4,3,2,1
17 37,1,4,3,2
18 32,4,1,3,2
19 26,1,4,2,3
20 13,1,2,4,3
21 12,4,1,2,3
22 11,4,2,1,3
23 8,2,3,4,1
24 7,1,2,3,4
25 6,2,3,1,4
26 4,4,2,3,1
27 4,2,1,3,4
28 2,2,1,4,3
29 1,2,4,3,1
30 1,2,4,1,3
31
```

ED-00004-00000196.soc

Generate data from normal distribution

Candidates amount:

Voters amount:

Candidates mean x:

Candidates mean y:

Candidates sigma:

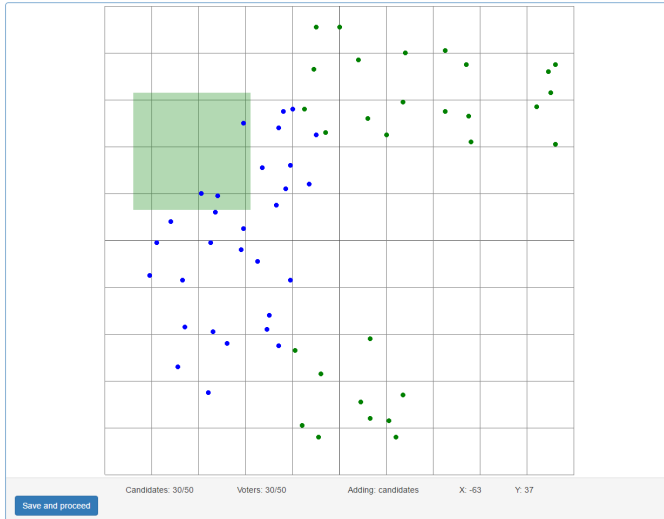
Voters mean x:

Voters mean y:

Voters sigma:

Generate data for election

Zaznaczanie punktów na płaszczyźnie



Dobór parametrów do obliczeń

Calculate new result for election "some_elections"

P parameter:

Algorithm:

Mutation probability:

Crossing probability:

Cycles:

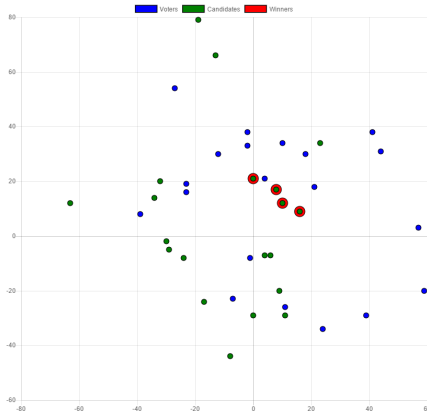
Add new result

Rozmieszczenie kandydatów, wyborców i zwycięzców

committee size: 4



Result for algorithm: Greedy with p: 2



Winners

Collapse/expand winners listing

Rodolfo

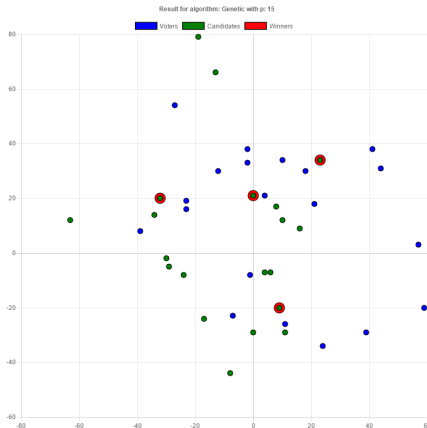
Erin

Cheryl

Esther

Rozmieszczenie kandydatów, wyborców i zwycięzców

committee size: 4



Winners

Collapse/expand winners listing

























Loren

Rodolfo

Latisha

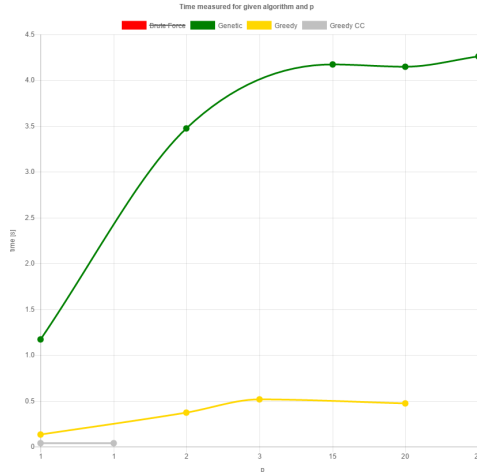
Kristen

Porównanie wyników dla różnych algorytmów

p param	Algorithm	Time	Committee score ⓘ	Algorithm parameters	Actions
1	Brute force	4.41 s	1099.00		 
1	Brute force	4.39 s	1099.00		 
1	Greedy Algorithm	0.13 s	1099.00		 
1	Greedy CC	0.04 s	921.00		 
1	Greedy CC	0.04 s	921.00		 
1	Genetic	1.17 s	1099.00	C 50 P 10 Q 20	 
2	Greedy Algorithm	0.37 s	555.63		 
2	Genetic	3.48 s	555.63	C 50 P 10 Q 20	 
3	Greedy Algorithm	0.52 s	447.83		 
15	Genetic	4.17 s	366.18	C 50 P 10 Q 20	 
20	Brute force	19.98 s	365.06		 
20	Greedy Algorithm	0.47 s	362.52		 
20	Genetic	4.15 s	365.06	C 50 P 10 Q 20	 
20	Genetic	4.26 s	365.06	C 50 P 10 Q 20	 

time [s]

Porównanie wyników dla różnych algorytmów



Algorytm brute-force

Data: K - zbiór wszystkich możliwych komitetów w danych wyborach

Result: $REZULTAT$ - zwycięski zbiór k kandydatów

```
 $REZULTAT \leftarrow \emptyset$   
 $najlepszy\_wynik \leftarrow 0$   
for  $k \in K$  do  
    if  $wynik(k) > najlepszy\_wynik$  then  
         $REZUTLAT \leftarrow k$   
         $najlepszy\_wynik \leftarrow wynik(k)$   
    end  
end  
return  $REZULTAT$ 
```


Algorytm zachłanny zależny od parametru p

```
for  $i \leftarrow 1$  to  $k$  do  
  | for  $c \in C \setminus REZULTAT$  do  
  |   | for  $v \in V$  do  
  |   |   |  $dodaj\_zadowolenie\_wyborcy(v, REZULTAT \cup c);$   
  |   | end  
  |   | if  $badany\_kandydat\_najlepszy(c)$  then  
  |   |   |  $uaktualnij\_lidera\_iteracji(c);$   
  |   | end  
  | end  
  |  $REZULTAT \leftarrow REZULTAT \cup zwyciezca\_iteracji;$   
end  
return  $REZULTAT$ 
```

Algorytm zachłanny wg zasady *Chamberlina* – *Couranta*

- niezależny od parametru p
- aproksymacja wg funkcji satysfakcji *Chamberlina* – *Couranta*
- schemat działania identyczny jak wcześniejszego algorytmu
- zdecydowanie szybszy od głównego algorytmu zachłannego
- dobra aproksymacja systemu ℓ_p – *Borda* dla dużych wartości parametru p

Osobnik

Jako początkową populację osobników przyjmujemy 50 losowo
wybranych komitetów.

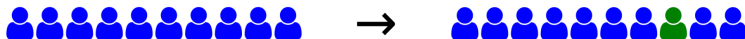


Krzyżowanie i mutacja

Podczas krzyżowania dwa osobniki wymieniają się losowo kandydatami. Tworzony jest nowy osobnik, zawierający k kandydatów którzy poprzednio byli w co najmniej jednym z osobników przystępujących do krzyżowania.



Podczas mutacji jeden z kandydatów w osobniku jest wymieniany na innego, niebędącego dotąd w danym osobniku.



Cykl

Na jeden cykl algorytmu składa się:

- wybór par osobników do krzyżowania się
- krzyżowanie osobników
- mutacja osobników zgodnie z zadaniem prawdopodobieństwem
- wybór N osobników do przejścia do kolejnego cyklu

Liczba N określa wielkość puli osobników i jest wybierana na początku działania algorytmu. W naszej implementacji przyjęto stałą liczbę 50 osobników.

Parametry algorytmu

System pozwala wybrać:

- liczbę cykli działania algorytmu
- część puli jaka jest poddawana krzyżowaniu (osobniki są z niej losowo wybierane do krzyżowania)
- prawdopodobieństwo wystąpienia mutacji w pojedynczym osobniku

Dziękujemy za uwagę