

**Akademia Górniczo-Hutnicza
im. Stanisława Staszica w Krakowie**

Wydział Informatyki, Elektroniki i Telekomunikacji

KATEDRA INFORMATYKI



DOKUMENTACJA UŻYTKOWNIKA

TOMASZ KASPRZYK, DANIEL OGIELA, JAKUB STĘPAK

**SYSTEM OBLICZAJĄCY WYNIKI WYBORÓW DLA
UOGÓLNIENIA SYSTEMU K-BORDA**

PROMOTOR:

dr hab. inż. Piotr Faliszewski

Kraków 2016

Spis treści

1. Wstęp	3
2. Instrukcja uruchomienia systemu	4
2.1. Przygotowanie środowiska	4
2.1.1. Język	4
2.1.2. Repozytorium	4
2.1.3. Biblioteki	4
2.1.4. Baza danych	5
2.2. Uruchomienie serwera	5
2.2.1. Metoda k-Borda	5
2.2.2. Uogólnienie - system ℓ_p Borda	6
2.3. Format danych wejściowych	6
2.4. Szybkość i dokładność wykonywanych obliczeń	6
3. Opis modułów	7

1. Wstęp

Niniejszy podręcznik opisuje sposób użytkowania systemu Election Computing System, powstałego w ramach pracy inżynierskiej realizowanej na Akademii Górniczo-Hutniczej w Krakowie.

System umożliwia obliczenie wyników wyborów w uogólnionym systemie wyborczym k-Borda. Szczegóły na temat tego systemu wyborczego można przeczytać w Przewodniku po projekcie.

2. Instrukcja uruchomienia systemu

Działającą aplikację można przetestować na stronie:

<https://election-computing-system.herokuapp.com>.

Ze względu na ograniczenia na ilość rekordów w bazie danych narzucone przez darmową wersję Heroku, nie będzie można dodać tam zbyt „dużych” wyborów (ograniczona jest liczba głosujących i kandydatów). Aby korzystać z pełnych możliwości aplikacji, należy uruchomić ją na własnym komputerze. Dalsza część tego rozdziału opisuje jak to wykonać.

2.1. Przygotowanie środowiska

2.1.1. Język

System jest aplikacją internetową opartą o framework Django napisaną w języku Python 2.7. Aby uruchomić aplikację, należy uprzednio zainstalować na swoim komputerze interpreter Pythona.

2.1.2. Repozytorium

Repozytorium projektu znajduje się pod adresem:

<https://github.com/jakubste/election-computing-system>. Projekt należy sklonować używając programu Git

```
$ git clone git@github.com:jakubste/election-computing-system.git
```

lub ściągnąć jako ZIP bezpośrednio z GitHuba i rozpakować w wybranym miejscu.

2.1.3. Biblioteki

Do instalacji bibliotek zaleca się używanie mechanizmu `virtualenv`, który separuje środowiska uruchomieniowe dla poszczególnych projektów. Autorzy projektu zalecają też dla wygody wykorzystanie `virtualenvwrapper`'a. Do instalacji bibliotek można użyć programu `pip`. Informacje o wymaganych bibliotekach są zawarte w pliku `requirements.txt`.

```
$ mkvirtualenv inz
```

```
$ pip install -r requirements.txt
```

2.1.4. Baza danych

Aplikacja domyślnie jest skonfigurowana do użycia z bazą danych dostarczaną przez Heroku (PostgreSQL). Dla ułatwienia zostanie przedstawiony sposób konfiguracji z użyciem SQLite3. O konfiguracji dostępu do innych baz danych można przeczytać w dokumentacji Django.

W celu skonfigurowania swojej bazy danych należy w katalogu `ecs` utworzyć plik `local_settings.py` i umieścić tam następujący kod:

```
from settings import *

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),
    }
}
```

Następnie w katalogu głównym programu należy wykonać polecenie:

```
$ ./manage.py migrate
```

2.2. Uruchomienie serwera

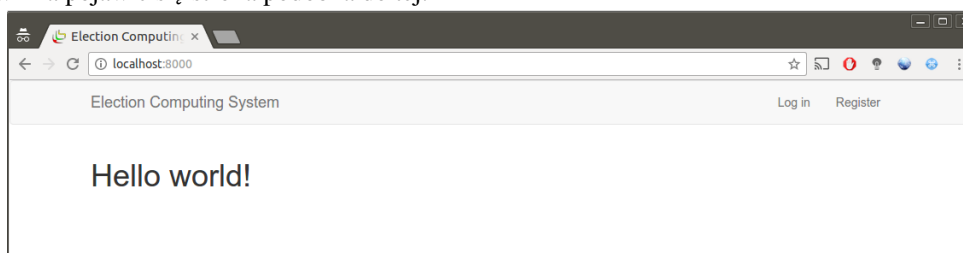
W tym momencie aplikacja powinna być gotowa do działania. W katalogu głównym programu należy wykonać polecenie:

```
$ ./manage.py run_server
```

Po otworzeniu w przeglądarce internetowej adresu:

```
http://localhost:8000/
```

powinna pojawić się strona podobna do tej:



2.2.1. Metoda k-Borda

Rozszerzenie metody Bordy. Wynik, zamiast dla jednego kandydata, obliczany jest dla ciągu kandydatów. f_{kB} -funkcja zadowolenia z komitetu. Ciąg (i_1, \dots, i_k) -ciąg pozycji kandydatów

Przykład $C = c_1, c_2, c_3, c_4$ - zbiór kandydatów, $v = (c_2, c_1, c_4, c_3)$ - głos Niech $k = 2$ (wybory 2 spośród 4) $w = (c_4, c_3)$ Najpierw określamy pozycje kandydatów z komitetu w w v : $pos_v(w) = (3, 4)$, zatem wynik komitetu w dla głosu v wynosi $f_{kB}(3, 4) = (3) + (4) = ||C|| - 3 + (||C|| - 4) = 1 + 0 = 1$

2.2.2. Uogólnienie - system ℓ_p Borda

Zanim wprowadzone zostanie pojęcie uogólnionego systemu k-Borda warto przypomnieć wzór na normę ℓ_p

Norma ℓ_p

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

Wówczas, w uogólnionej wersji metody k-Borda, funkcja zadowolenia f_{kB} zostaje uzależniona również od parametru p z powyższego wzoru. Norma liczona jest z wyników według Bordy, $\beta(i)$. Wzór uogólniony funkcji zadowolenia przyjmuje zatem postać:

$$f_{\ell_p B}(p, (i_1 1, \dots, i_k)) = p[(i_1)]p + [(i_2)]p + \dots + [(i_k)]p$$

Systemy k-Borda i Cahmberlin'a-Courant'a są szczególnymi przypadkami zdefiniowanego powyżej systemu ℓ_p -Borda:

Dla $p = 1, l_1$

$$f_{\ell_p B}(1, (i_1, \dots, i_k)) = \beta(i_1) + \beta(i_2) + \dots + \beta(i_k) = f_{kB}(i_1 1, \dots, i_k)$$

Dla $p = \infty, l_\infty = \max$

$$f_{\ell_p B}(\infty, (i_1, \dots, i_k)) = \lim_{p \rightarrow \infty} \sqrt[p]{\beta[(i_1)]^p + \beta[(i_2)]^p + \dots + [\beta(i_k)]^p} = \max \beta(i_1), \beta(i_2), \dots, \beta(i_k) = \beta(i_1) = f_{CC}$$

2.3. Format danych wejściowych

2.4. Szybkość i dokładność wykonywanych obliczeń

3. Opis modułów