

**AKADEMIA GÓRNICZO-HUTNICZA
IM. STANISŁAWA STASZICA W KRAKOWIE**

Wydział Informatyki, Elektroniki i Telekomunikacji
Katedra Informatyki



PROJEKT INŻYNIERSKI

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

**TOMASZ KASPRZYK, DANIEL OGIELA
JAKUB STĘPAK**

OPIEKUN:
dr hab. inż. Piotr Faliszewski

Kraków 2016

OŚWIADCZENIE AUTORA PRACY

OŚWIADCZAM, ŚWIADOMY(-A) ODPOWIEDZIALNOŚCI KARNEJ ZA POŚWIADCZENIE NIEPRAWDY, ŻE NINIEJSZY PROJEKT WYKONAŁEM(-AM) OSOBIŚCIE I SAMODZIELNIE W ZAKRESIE OPISANYM W DALESZEJ CZĘŚCI DOKUMENTU I ŻE NIE KORZYSTAŁEM(-AM) ZE ŹRÓDEŁ INNYCH NIŻ WYMIENIONE W DALESZEJ CZĘŚCI DOKUMENTU.

.....

PODPIS

Spis treści

1	Wizja produktu	4
1.1	Opis problemu	4
1.2	Możliwości	6
1.3	Motywacja do stworzenia projektu	6
1.4	Użytkownicy	7
1.5	Usługi dostarczanie przez system	7
1.5.1	Usługi wymagane	7
1.5.2	Usługi pożądane	7
1.5.3	Usługi dodatkowe	7
2	Studium wykonalności	7
2.1	Opis wymagań	7
2.1.1	Wymagania funkcjonalne	7
2.1.2	Wymagania нефункционалне	10
2.2	Strategia testowania	10
2.3	Aspekt technologiczny	11
2.4	Analiza ryzyka	11

1. Wizja produktu

1.1. Opis problemu

Projekt dotyczy obliczania wyników wyborów. Wybory i sposób wyłaniania zwycięzców są ściśle określone. Wybory można opisać jako parę (C, V) , gdzie $C = \{c_1, c_2, \dots, c_m\}$ stanowi zbiór kandydatów, a $V = \{v_1, v_2, \dots, v_n\}$ to ciąg wyborców. Każdy z wyborców posiada swoje preferencje wyborcze, które są opisane przez ciąg kandydatów ze zbioru C . Długość ciągu kandydatów wynosi m , a kandydaci są uporządkowani od najbardziej preferowanego do najmniej preferowanego. Ponadto określony jest rozmiar k zwycięskiego komitetu, który określa liczbę kandydatów, którzy zwyciężyli w wyborach.

Przykładowe wybory

$$C = \{a, b, c, d, e, \dots\}$$

$$V = (v_1, v_2, v_3, \dots, v_n)$$

$$v_1 : a > b > c > d > e > \dots$$

$$v_2 : c > d > a > b > f > \dots$$

$$v_3 : \dots$$

$$\vdots$$

$$v_n : f > g > e > b > a > \dots$$

$$k = 20$$

Ciąg kandydatów przyporządkowany danemu wyborcy traktowany jest jako głos w wyborach. Każdemu kandydatowi w ciągu stanowiącym głos, przyporządkowane są punkty według punktacji Bordy. Jeżeli przez i zostanie oznaczona pozycja danego kandydata w ciągu stanowiącym głos (jeżeli kandydat jest pierwszy w ciągu, wtedy $i = 1$, jeżeli drugi, wtedy $i = 2$ itd.), to wartość punktowa przyporządkowana według metody Bordy w tym głosie dla tego kandydata wynosi $\beta(i) = m - i$, gdzie m to rozmiar zbioru C (liczba wszystkich kandydatów). Zatem dla danego głosu kolejni uporządkowani kandydaci otrzymują kolejno $m - 1, m - 2, \dots, 1, 0$ punktów.

Preferencje wyborcy

$$v_1 : \overset{m-1}{a} > \overset{m-2}{b} > \overset{m-3}{c} > \overset{m-4}{d} > \dots$$

W celu wyłonienia zwycięzców wyborów komitetów definiuje się tzw. funkcję satysfakcji, która bazuje między innymi na punktacji metodą Bordy. Funkcja satysfakcji określa zadowolenie danego wyborcy ze zwycięskiego komitetu. W celu zwięzłego zdefiniowania funkcji celu wprowadza się pojęcie ciągu pozycji, które określa dla danego wyborcy pozycje wszystkich zwycięskich kandydatów z jego preferencji. Ciąg pozycji jest posortowany rosnąco - pozycja najbardziej preferowanego kandydata spośród zwycięzców na początku ciągu, pozycja najmniej preferowanego kandydata spośród zwycięzców na końcu ciągu.

Przykład

Dla następujących preferencji:

Preferencje wyborcy

$$v_3 : \overset{m-1}{g} > \overset{m-2}{d} > \overset{m-3}{e} > \overset{m-4}{b} > \overset{m-5}{f} > \overset{m-6}{a} > \dots$$

komitetu $K = \{a, b, e\}$ i wyborcy v_3 ciąg pozycji oznaczany pos_{v_3} wynosi $(3, 4, 6)$, gdyż pozycja najlepszego kandydata ze zbioru K wynosi 3 (kandydat e), druga najlepsza to pozycja nr 4 (kandydat b), a ostatnią pozycją najmniej preferowanego kandydata jest pozycja nr 6 (kandydat a).

W tym momencie można zdefiniować funkcję satysfakcji, która jako argument przyjmuje zdefiniowany przed chwilą ciąg pozycji, który dla każdego wyborcy i dla każdego komitetu jest określony jednoznacznie. Funkcja satysfakcji określająca zadowolenie danego wyborcy ze zwycięskiego komitetu, będzie więc oznaczana jako $f(i_1, i_2, \dots, i_k)$.

Poprzez różne zdefiniowanie funkcji satysfakcji, definiowane są odmienne systemy wyborcze. Znanymi systemami wyborczymi są system k – Borda oraz system Chamberlin – Courant'a. Funkcje satysfakcji dla tych systemów określone są odpowiednio:

$$f_{k-Borda}(i_1, i_2, \dots, i_k) = \beta(i_1) + \beta(i_2) + \dots + \beta(i_k)$$

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

Wyniki wyborów dla podanych systemów mogą być różne (i zazwyczaj są) pomimo takich samych danych wejściowych. Inaczej rzecz ujmując, zwycięskie komitety mogą składać się z różnych kandydatów po zastosowaniu odmiennych funkcji satysfakcji.

Funkcja satysfakcji, która dotyczy niniejszego projektu inżynierskiego jest oparta na normie ℓ_p . Norma ℓ_p zdefiniowana jest następująco:

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

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

Dla skrajnych przypadków, gdy $p = 1$ i gdy $p \rightarrow \infty$ norma ℓ_p przyjmuje odpowiednio postać sumy:

$$l_1(x_1, x_2, \dots, x_n) = x_1 + x_2 + \dots + x_n$$

oraz postać operatora maksimum:

$$l_\infty(x_1, x_2, \dots, x_n) = \max(x_1, x_2, \dots, x_n)$$

Mając zdefiniowaną normę ℓ_p można określić funkcję satysfakcji będącą tematem pracy.

$$f_{\ell_p-Borda}(i_1, i_2, \dots, i_k) = \ell_p(\beta(i_1), \beta(i_2), \dots, \beta(i_k)) = \sqrt[p]{[\beta(i_1)]^p + [\beta(i_2)]^p + \dots + [\beta(i_k)]^p}$$

Funkcja jest uogólnieniem systemu $k - Borda$. Jeżeli za p przyjęta zostanie wartość 1, zdefiniowana funkcja przyjmuje postać funkcji satysfakcji dla zwykłego systemu $k - Borda$. Dla p dążącego do nieskończoności funkcja przyjmuje postać funkcji satysfakcji dla systemu *Chamberlin - Courant'a*.

1.2. Możliwości

Obliczanie wyników wyborów dla systemu $\ell_p - Borda$ metodą typu brute-force jest czasochłonne. Już dla stosunkowo niewielkich rozmiarów danych, algorytm jest nieużyteczny. Dlatego głównym zadaniem projektu jest zaprojektowanie i implementacja algorytmów heurystycznych, które możliwie dokładnie i możliwie szybko obliczają wyniki wyborów. Ponadto do zadań zespołu należy próba oceny działania stworzonych algorytmów heurystycznych.

1.3. Motywacja do stworzenia projektu

Jednym z głównym celów realizacji projektu jest chęć poznania różnych sposobów tworzenia algorytmów heurystycznych oraz zastosowanie ich do rozwiązania problemów w interesującej zespół tematyce, jaką jest sposób wyłaniania zwycięzców w wyborach i związek między tym sposobem a satysfakcją wyborców ze zwycięzców. Ponadto zespół chciał porównać pod różnymi względami różne algorytmy heurystyczne. Interesujące jest zestawienie algorytmów pod względem dokładności obliczanych rozwiązań, czasu działania, czy trudności w projektowaniu i implementacji.

Drugą, istotną motywacją do stworzenia systemu jest możliwość wykorzystania go do wielu bardzo różnych sytuacji. Zdefiniowanie wyborów, których wyniki ma liczyć system, może dotyczyć różnego rodzaju wyborów. Mogą to być wybory ludzi do wszelakiego typu organów władz na różnych szczeblach organizacji państwowych lub organizacji prywatnych. Wybory nie muszą dotyczyć ludzi, a mogą polegać na selekcji innych rzeczy. Można zdefiniować wybory na filmy, które zostaną odtworzone w trakcie podróży samolotem, czy podczas seansu z rodziną i znajomymi. Mogą to być wybory na miejsca wspólnego wypadu znajomych na weekend. Wszędzie, gdzie ma sens zastosowanie preferencji (ułożenie rzeczy od najbardziej do najmniej pożądaney) można wykorzystać stworzony system.

Dzięki stworzonemu systemowi według określonych wymagań, możliwe jest nie tylko szybkie obliczenie wyborów na bazie wprowadzonych do systemu preferencji. Możliwe jest sterowanie sposobem obliczania wyborów według pożądanej reprezentatywności zwycięskich kandydatów. Można decydować czy zwycięzcy wyborów będą tacy, że każdy znajdzie wśród nich takiego, którego bardzo lubi, ale też taki, którego bardzo nie lubi. Czy może zwycięzcy mają być tacy, że większość wyborców co prawda nie znajdzie wśród nich bardzo pożądanego kandydata, ale jednocześnie nie znajdzie takiego, którego bardzo nie lubi. Możliwy jest również wybór pośredni między wskazanymi dwoma rozwiązaniami.

Przykładowo, jeżeli użytkownik chce wybrać kilka filmów do wspólnego seansu z przyjaciółmi i rodziną, to może zdecydować między różnymi opcjami. Jedną z opcji jest taki wybór, aby wśród wybranych filmów każdy ze znajomych znalazł film, który mu bardzo odpowiada, lecz jednocześnie znalazły się takie, które większości znajomych wyraźnie nie odpowiadają. Druga możliwość to taka, w której wybrane zostaną filmy, które większości znajomych ani za bardzo nie odpowiadają, ani za bardzo nie przeszkadzają. Pozostałymi możliwościami są opcje pośrednie między wskazanymi.

1.4. Użytkownicy

Potencjalnymi użytkownikami systemu mogą być osoby i organizacje, które potrzebują systemu wybierania kilku elementów spośród puli wszystkich elementów w taki sposób, aby usatysfakcjonować swoich klientów. System może mieć przykładowe zastosowanie dla przedsiębiorstw realizujących usługi transportu lotniczego czy autokarowego do umilenia czasu podróży poprzez wybór odpowiednich filmów czy potraw. Ponadto system może służyć pojedynczym osobom w celu wyboru odpowiedniej formy spędzania wolnego czasu w grupie znajomych.

1.5. Usługi dostarczanie przez system

1.5.1. Usługi wymagane

- możliwość wprowadzenia wyborów do systemu
- obliczenie wyników wyborów
- wyświetlanie wyników wyborów

1.5.2. Usługi pożądane

- możliwość zarządzania swoimi wyborami (dodawanie nowych wyborów, nowych wyników, porównywanie różnych rezultatów)
- intuicyjna nawigacja po systemie

1.5.3. Usługi dodatkowe

- wygodny sposób definiowania wyborów - zadanie projektowe koncentruje się na projektowaniu i tworzeniu algorytmów obliczających wyniki wyborów. System musi wczytywać pewien format pliku wejściowego definiującego wybory, który niekoniecznie jest prosty i intuicyjny w tworzeniu. Jeżeli ta dodatkowa usługa zostałaby zrealizowana, ułatwiłaby użytkowanie systemu i poprawiła zadowolenie użytkowników z korzystania z systemu

2. Studium wykonalności

2.1. Opis wymagań

2.1.1. Wymagania funkcjonalne

Poniżej zamieszczona jest lista najważniejszych wymagań w postaci kolejnych tabel. Tabela składa się z trzech pól opisujących dane wymaganie: *Nazwa*, *Opis* i *Uzasadnienie*. Pole *Nazwa* jest nazwą usługi dostarczanej przez system, pole *Opis* szczegółowo opisuje w jaki sposób jest realizowane dane wymaganie, a pole *Uzasadnienie* podaje powód dlaczego w taki, a nie inny sposób wymaganie zostało zrealizowane i postawione.

Nazwa	Zdefiniowanie wyborów
Opis	<p>System oferuje różne sposoby zdefiniowania i wprowadzenia wyborów do systemu. Pierwszym i podstawowym jest możliwość określania wyborów w pliku formatu <i>.soc</i>. Sposób określania w nim kandydatów oraz głosów wyborców jest ściśle określony. W pierwszej linii pliku znajduje się liczba kandydatów do zwycięskiego komitetu. W kolejnych liniach znajdują się numery i nazwy kandydatów. Następnie jest linia z informacją o liczbie głosujących, liczbie policzonych głosów oraz liczbie głosów unikalnych. W ostatnim bloku pliku znajdują się linie z informacją o preferencjach głosów unikalnych. Każda linia składa się z liczby powtórzeń danego głosu oraz preferencji określonej dla tego głosu, która ma postać ciągu numerów kandydatów od najbardziej do najmniej pożądanego. Po stworzeniu opisanego pliku system umożliwia wskazanie pliku z systemu plików.</p> <p>Drugim sposobem definiowania wyborów oferowanym przez system jest generowanie wyborów z rozkładu normalnego. W tym celu kandydaci i wyborcy są reprezentowani jako punkty płaszczyźnie. Preferencje wyborców obliczane są na bazie odległości euklidesowych do poszczególnych kandydatów na płaszczyźnie. Aby wygenerować wybory użytkownik podaje w formularzu parametry potrzebne do generacji wyborów zgodnie z rozkładem normalnym. Uzupełnia pola dotyczące: liczby kandydatów, średniej wartości współrzędnej x dla wyborców, średniej wartości współrzędnej x dla kandydatów, średniej wartości współrzędnej y dla wyborców, średniej wartości współrzędnej y dla kandydatów, odchylenia standardowego wyborców oraz odchylenia standardowego kandydatów.</p>
Uzasadnienie	<p>Definicja wyborów za pomocą pliku formatu <i>.soc</i> zapewnia spełnienie podstawowej usługi oferowanej przez system - wprowadzenia wyborów do systemu. Uzasadnieniem dla konieczności generacji wyborów z rozkładu normalnego jest możliwość atrakcyjnej i klarownej wizualizacji wyborów i wyników wyborów. Wizualizacja w ten sposób wyników wyborów umożliwia między innymi ocenienie jakości stworzonych algorytmów i porównanie ich.</p>

Tabela 1: Zdefiniowanie wyborów

Nazwa	Obliczanie wyników wyborów
Opis	<p>System oferuje kilka algorytmów heurystycznych obliczających wyniki wyborów. Algorytmy bazują na różnym paradygmacie tworzenia ich. Jeden z paradygmatów to programowanie zachłanne, a drugi to programowanie genetyczne. Algorytmy dają przybliżone rozwiązania problemu pod względem jego dokładności. Użytkownik przy tworzeniu nowego wyniku dla zdefiniowanych wyborów może w prosty sposób wybrać jeden z algorytmów.</p>
Uzasadnienie	<p>Kilka stworzonych algorytmów i różne podejście do tworzenia ich, umożliwia ocenę jakości algorytmów na bazie zestawienia wyników obliczonych wyborów. Ponadto stworzenie różnych algorytmów pozwoli na dogłębsze poznanie sposobów tworzenia algorytmów heurystycznych, jak również samej tematyki pracy, jaką były różne sposoby wyboru zwycięskiego komitetu.</p>

Tabela 2: Obliczanie wyników wyborów

Nazwa	Wersja algorytmu zachłannego <i>Greedy</i> – <i>CC</i>
Opis	Implementacja algorytmu zachłannego dla szczególnego przypadku (parametr $p \rightarrow \infty$)
Uzasadnienie	Dla stosunkowo "dużych" wartości parametru p wykonywanie operacji pierwiastkowania i potęgowania stanowi niepotrzebny narzut czasowy. W tym przypadku operację obliczania normy ℓ_p można zastąpić wyznaczaniem maksimum dla pojedynczych wyników kandydatów z komitetu, dla którego liczono by normę ℓ_p . Korzyści jakie płyną z takiej modyfikacji to nie tylko skrócenie czasu obliczeń, ale także ustalenie od jakiej wartości parametru p komitet wygrywający jest zgodny z wynikiem otrzymanym dla systemu <i>Chamberlin</i> – <i>Courant'a</i> .

Tabela 3: Wersja algorytmu zachłannego *Greedy* – *CC*

Nazwa	Prezentacja wyników wyborów
Opis	System oferuje dwa sposoby prezentacji wyników wyborów. Pierwszy z nich, to proste wskazanie kandydatów i wypisanie nazw kandydatów w widocznym panelu. Ta forma prezentacji dotyczy wyników wyborów, które zdefiniowane zostały za pomocą pliku formatu <i>.soc</i> . Druga forma prezentacji wyników wyborów dotyczy wyborów wygenerowanych z rozkładu normalnego. Wizualizacja polega na narysowaniu wykresu $2D$, na którym punkty reprezentujące kandydatów ze zwycięskiego komitetu są wyraźnie wyróżnione na tle kandydatów, którzy nie dostali się do zwycięskiego komitetu. Dla wyborów generowanych z rozkładu normalnego system również oferuje wypisanie nazw zwycięzców w dobrze widocznym panelu.
Uzasadnienie	Prezentacja wyników wyborów w postaci wykresu $2D$ z wyraźnie zaznaczonymi zwycięzcami i przegranymi, którzy są reprezentowani jako punkty, pozwala na porównanie w prosty sposób wyników wyborów dla różnych wartości parametru p . Taka forma wizualizacji jest jednocześnie wizualnie atrakcyjna. Prezentacja wyników dla wyborów zdefiniowanych z pliku formatu <i>.soc</i> ogranicza się jedynie do wypisania zwycięzców, ponieważ kandydaci nie są reprezentowani jako punkty i tym samym nie mają współrzędnych w tej odmianie wyborów.

Tabela 4: Prezentacja wyników wyborów

2.1.2. Wymagania niefunkcjonalne

Wymagania niefunkcjonalne podzielono na wymagania produktowe i wymagania organizacyjne. Poniżej zamieszczono tabele analogiczne do tych z wymagań funkcjonalnych dla odpowiednich typów wymagań niefunkcjonalnych.

Wymagania produktowe

Nazwa	Obliczanie wyników wyborów w satysfakcjonującym czasie
Opis	Ponieważ dla już stosunkowo małych danych metoda brute-force rozwiązywania opisanego problemu nie daje rezultatów w satysfakcjonującym czasie, więc to wymaganie niefunkcjonalne jest kluczowe w tym projekcie. Zastosowanie algorytmów heurystycznych wychodzi naprzeciw temu wymaganiu. Algorytmy powinny dawać rozwiązania w zadowalającym czasie dla możliwie dużych danych.
Uzasadnienie	Czas działania algorytmu powinien się mieścić w możliwym do zaakceptowania przedziale czasu ze względu na konieczność jego użyteczności.

Tabela 5: Obliczanie wyników wyborów w satysfakcjonującym czasie

Wymagania organizacyjne

Nazwa	Dotrzymanie terminów na przedstawienie poszczególnych elementów pracy
Opis	Kolejne elementy pracy, które należało przedstawiać w umówionych terminach to: wizja produktu, studium wykonalności, prototyp oraz tuż przed końcem projektu: produkt końcowy wraz z dokumentacją.
Uzasadnienie	Terminy są dostosowane do trybu wykonywania pracy inżynierskiej, który jest ustalany przez władze uczelni.

Tabela 6: Obliczanie wyników wyborów w satysfakcjonującym czasie

2.2. Strategia testowania

Podstawowym rodzajem testów pisanych w trakcie tworzenia systemu były testy jednostkowe. Były one dodawane na bieżąco od razu przy dodawaniu kolejnych funkcjonalności systemu.

Ponadto w celu zagwarantowania poprawności działania wcześniej dodanych funkcjonalności po dodaniu nowych skorzystano z ciągłej integracji. Do tego celu wykorzystano serwis Travis CI skonfigurowany do repozytorium kodu. Przy każdej zmianie kodu źródłowego w repozytorium dokonywane było automatyczne włączenie testów jednostkowych.

Do oceny poprawności obliczanych wyników wyborów przez stworzone algorytmu, przeprowadzono testy porównawcze. Zestawiano wyniki wyborów obliczonych przez różne algorytmy dla tych samych danych wejściowych. Ponadto porównywano wykresy wyników wyborów z oczekiwanymi rezultatami.

2.3. Aspekt technologiczny

Do stworzenia systemu wykorzystano język *Python2.7*. Produkt postanowiono wykonać w postaci aplikacji webowej. Aplikację internetową oparto na frameworku pythonowym *Django1.9*. Do tworzenia stron internetowych użyto frameworku *Bootstrap*. Do wdrożenia systemu wykorzystano platformę *Heroku*.

Wybrano taki stos technologiczny ze względu na doświadczenie części zespołu pracy z nim. Wiedza na temat tych narzędzi przekonywała zespół o możliwości zrealizowania projektu w tej technologii.

2.4. Analiza ryzyka

Głównym zagrożeniem dla realizacji produktu mogły być problemy ze stworzeniem wystarczająco dobrego algorytmu heurystycznego dla opisanego problemu. Ze względu na niewielkie doświadczenie zespołu w tej dziedzinie taka sytuacja mogła wystąpić. Głównym zabezpieczeniem na wypadek takiego scenariusza było stosunkowo szybkie stworzenie bazy całego systemu i możliwość wczesnego skupienia się na algorytmach heurystycznych.

Innym, dużo mniejszym zagrożeniem dla projektu mógł być brak skoordynowania prac poszczególnych członków zespołu. Ponieważ dla członków zespołu projekt inżynierski był tylko jednym z wielu zajęć w czasie trwania projektu, mogła wystąpić sytuacja kiedy ze względu na natężenie innych obowiązków, dany członek zespołu nie mógłby wykonać koniecznej pracy w odpowiednim czasie.