

**Uniwersytet Warszawski**  
Wydział Matematyki, Informatyki i Mechaniki

**Jakub Grondziowski**

Nr albumu: 417886

# Obliczanie i wizualizacja wyników metody równych udziałów

Praca magisterska  
na kierunku INFORMATYKA

Praca wykonana pod kierunkiem  
**dra. hab. Piotra Skowrona**  
Instytut Informatyki

Warszawa, Wrzesień 2024



## Streszczenie

Stanford Participatory Budgeting Platform, rozwijany przez Stanford Crowdsourced Democracy Team, to zaawansowany system umożliwiający przeprowadzanie głosowań w ramach budżetu partycypacyjnego. Oprogramowanie to zostało już z sukcesem użyte w wielu amerykańskich miastach. Platforma ta posiada wiele funkcjonalności przydatnych organizatorom podczas przeprowadzania wyborów - ułatwia tworzenie i zarządzanie projektami, udostępnia podstawowe statystyki oddanych głosów, dostarcza wiele sposobów weryfikacji wyborców, a także pozwala na wybór sposobu oddawania głosu. Wśród tych sposobów możemy wyróżnić te najbardziej popularne, czyli głosowanie przez aprobaty oraz głosowanie w skali, jak i te mniej znane, takie jak głosowanie z limitem budżetu czy głosowanie z użyciem rankingu.

W systemie tym brakuje jednak algorytmów do przeliczania głosów i wybierania zwycięskich projektów. W przypadku wyłaniania zwycięzców metodą większościową można posłużyć się udostępnionymi statystykami, jednakże w przypadku chęci użycia bardziej zaawansowanych metod organizatorzy zmuszeni są do eksportu wyników do niewygodnego w użyciu pliku i jego transformację do odpowiedniego formatu rozumianego przez inne narzędzia. Wprowadza to wiele niepotrzebnych kroków które kosztują czas i podczas których może dojść do błędu ludzkiego, co w przypadku głosowań na budżet partycypacyjny jest niedopuszczalne.

Jednym z celów tej pracy jest zaimplementowanie sposobu, który pozwoli organizatorom wyborów na automatyczne wyłonienie zwycięskich projektów z użyciem jednej z dostępnych metod. Rozwiązanie te będzie umożliwiało użycie dowolnej reguły wyłaniania zwycięzców, a także ich rozszerzeń decydujących o rozdysponowaniu pozostałych środków w przypadku nie wykorzystania całego budżetu.

Pabutools [Fal+23], będąca najpopularniejszą biblioteką używaną do obliczeń z użyciem zaawansowanych reguł wyborczych, udostępnia wiele funkcji umożliwiających pracę z instancjami i profilami wyborczymi. Wykorzystywana jest zarówno przez platformy do głosowania jak i naukowców badających własności wszelakich reguł wyborczych. W narzędziu tym brakuje jednak sposobu na wygenerowanie statystyk obrazujących działanie jednej z dostępnych reguł - metody równych udziałów. Statystyki te byłyby szczególnie pomocne osobom nie zaznajomionym z metodą, które chciałyby w prosty sposób zobaczyć jak metoda wybrała zwycięską alokację.

Ponieważ najczęściej mówimy tu o prostych wyborcach nie mających przeważnie do czynienia z rozwiązaniami informatycznymi, konieczne też będzie stworzenie sposobu na wizualizację tychże danych. Surowe statystyki, pomimo posiadania wszystkich koniecznych informacji, często bywają trudne w zrozumieniu nie tylko dla zwykłych ludzi, ale także ekspertów w danej dziedzinie.

Kolejnym więc celem pracy będzie implementacja narzędzi i rozwiązań, które nie tylko pozwolą na generację danych statystycznych opisujących przebieg metody równych udziałów w bibliotece Pabutools, ale także ich łatwą generację i wizualizację przez użytkowników zewnętrznych.

Autor będzie musiał zmierzyć się z kilkoma wyzwaniami:

1. Efektywna implementacja statystyk dotyczących wyznaczonych rozwiązań. Przykładowo, w przypadku bardziej złożonych metod wyborczych nie jest jasne ile głosów brakowało, aby dany projekt został wybrany. Znalezienie minimalnej liczby głosów, która by gwarantowała zwycięstwo jest problemem NP-trudnym. Celem będzie efektywne zaimplementowanie takich metryk lub zaprojektowanie alternatywnych metryk, które dobrze aproksymują odpowiednie kryteria, lub mają podobną intuicyjną interpretację.

2. Efektywna implementacja metody równych udziałów. Aby móc tę metodę obliczać dla dużych instancji wyborczych konieczne może być zaproponowanie nowych rozwiązań algorytmicznych i implementacji.

3. Zrozumienie architektury i zaprojektowanie rozszerzenia istniejącego systemu do głosowania. Wspomniane metody wyborcze i kryteria pochodzą z najnowszych prac publikowanych na takich konferencjach jak EC, NeurIPS i IJCAI.

## **Słowa kluczowe**

budżet partycypacyjny, wybór społeczny, metoda równych udziałów, Stanford Participatory Budgeting Platform, Pabutools, Pabulib, budżet zwolenników, efektywne poparcie

## **Dziedzina pracy (kody wg programu Socrates-Erasmus)**

11.3 Informatyka

## **Klasyfikacja tematyczna**

+ CCS

- Theory of computation
  - Theory and algorithms for application domains
    - \* Algorithmic game theory and mechanism design
      - Algorithmic game theory
- Computing methodologies
  - Artificial intelligence
    - \* Distributed artificial intelligence
      - Multi-agent systems

## **Tytuł pracy w języku angielskim**

Result calculation and visualization of method of equal shares



# Spis treści

|                                                                     |    |
|---------------------------------------------------------------------|----|
| <b>1. Wstęp</b>                                                     | 7  |
| <b>2. Architektura rozwiązania</b>                                  | 15 |
| <b>3. Rozwiązania</b>                                               | 17 |
| 3.1. Obliczanie wyników w Stanford Participatory Budgeting Platform | 17 |
| 3.2. Ekstrakcja danych statystycznych metody równych udziałów       | 18 |
| 3.2.1. Faza ekstrakcji danych detalicznych                          | 18 |
| 3.2.2. Faza wyliczeń danych statystycznych                          | 23 |
| 3.3. Wizualizacja danych statystycznych                             | 29 |
| 3.3.1. Serwer front-endu                                            | 29 |
| 3.3.2. Serwer back-endu                                             | 30 |
| <b>4. Obsługa i instalacja</b>                                      | 31 |
| 4.1. Stanford Participatory Budgeting Platform                      | 31 |
| 4.2. Pabutools                                                      | 32 |
| 4.2.1. Ekstrakcja danych o przebiegu metody równych udziałów        | 32 |
| 4.2.2. Wyliczenie danych statystycznych dla wyników wyborów         | 32 |
| 4.2.3. Wyliczenie efektywnego poparcia projektów                    | 33 |
| 4.2.4. Inne zmiany w Pabutools                                      | 34 |
| 4.3. Wizualizacja danych statystycznych metody równych udziałów     | 34 |
| 4.3.1. Obsługa                                                      | 34 |
| 4.3.2. Instalacja                                                   | 39 |
| <b>5. Lokalizacja kodu źródłowego</b>                               | 41 |
| 5.1. Stanford Participatory Budgeting Platform                      | 41 |
| 5.2. Pabutools                                                      | 41 |
| 5.3. Wizualizacja danych statystycznych metody równych udziałów     | 41 |
| <b>6. Podsumowanie</b>                                              | 43 |



# Rozdział 1

## Wstęp

Budżet partycypacyjny, często nazywany również budżetem obywatelskim, jest narzędziem demokratycznym, umożliwiającym mieszkańcom bezpośredni udział w podejmowaniu decyzji dotyczących przeznaczenia części budżetu publicznego gmin i miast. Jego historia sięga końca lat 80. XX wieku, kiedy to w brazylijskim mieście Porto Alegre, w obliczu trudnych warunków ekonomicznych i społecznych, mieszkańcy miasta wspierani przez lokalne władze, postanowili aktywnie uczestniczyć w alokacji zasobów finansowych miasta [SOU98]. Pomysł ten bardzo przypadł do gustu zarówno w Brazylii, jak i w całej reszcie świata. Wraz z upływem czasu coraz więcej miast zaczęło organizować wybory budżetu partycypacyjnego, w którym to urząd miasta przeznacza wybraną kwotę na realizację projektów, a mieszkańcy miasta i/lub urząd zgłaszają potencjalne projekty oraz ich koszty realizacji (budżet przeznaczony na realizację projektów, jak i same projekty i ich cechy nazywa się instancją wyborczą) a następnie odbywa się głosowanie, w którym ci sami mieszkańcy decydują, które ze zgłoszonych projektów powinny zostać zrealizowane. Rozwiązanie to stało się szczególnie popularne w Europie i obu Amerykach. Coraz częściej do oddania głosów i wyliczania wyników wyborów budżetu partycypacyjnego korzysta się z rozwiązań informatycznych, które pozwalają na zdalne przeprowadzanie wyborów.

W głosowaniach na budżet partycypacyjny mamy do czynienia z wieloma formatami oddawania głosów, które umożliwiają wyrażenie preferencji głosującego. Możemy się spotkać z formatem rankingowym, w którym głosujący jest proszony o posortowanie projektów w kolejności od najbardziej do najmniej przez niego popieranego, czy też formatem żetonowym, w którym głosujący proszony jest o przyznanie skończonej liczby żetonów projektom, które popiera, ofiarując więcej żetonów projektom, które są przez niego bardziej popierane. Najpopularniejszym formatem jest jednak głosowanie przez aprobaty, w której głosujący jest proszony o zaznaczenie dowolnej liczby projektów, które popiera. Ta forma ma też swoje warianty, które ograniczają maksymalną lub minimalną liczbę aprobat, które wyborca może przyznać.

Niezależnie jednak od formatu głosowania, najpopularniejszą metodą stosowaną do wyłaniania zwycięzców na podstawie oddanych głosów jest metoda zachłanna [HN21]. Jest to bardzo prosta metoda, która, otrzymując na wejście oddane przez wyborców głosy (zwane dalej profilem wyborczym), sortuje projekty malejąco po liczbie oddanych na nią głosów (aprobat, żetonów, rankingów), a następnie przechodzi po tak utworzonej liście, wybierając do realizacji projekty, które mogą zostać sfinansowane z dostępnego budżetu. Tak wyłoniony zbiór zwycięzców (zwany często alokacją lub zwycięską alokacją) w wielu przypadkach nie oddaje prawdziwych preferencji grupy wyborczej. O takich alokacjach mówimy, że są nieproporcjonalne [Bri+23].

Wyobraźmy sobie sytuację, w której projekty przypadają do jednej z trzech kategorii: projekty rozwijające infrastrukturę rowerową, projekty rozbudowujące parki i projekty dotyczące poprawienia warunków bezpieczeństwa zwierząt. W grupie wyborców, w której 40% stanowią entuzjaści rozbudowy parków, a po 30% stanowią wyborcy popierający rozbudowę infrastruktury rowerowej i poprawienie warunków bezpieczeństwa zwierząt, metoda zachłanna faworyzować będzie realizację projektów dotyczących parków. Oznacza to, że pozostałe 60% głosów będzie niedoreprezentowana, a w najgorszym przypadku żaden projekt z tych kategorii nie zostanie wybrany, jeśli projekty dotyczące parków pochłoną cały dostępny budżet.

Niedoreprezentowanie można również wyobrazić sobie w kontekście dzielnic miasta. W szczególności w większych miastach, głosujący bardzo często głosować będą na projekty znajdujące się blisko niego, czyli w okolicy swojej dzielnicy. Jeśli któraś z dzielnic ma znacząco większą liczbę mieszkańców, projekty z tej dzielnicy będą faworyzowane przez metodę zachłanną nad pozostałymi projektami (ze względu na większą liczbę głosów oddanych na projekty z liczniejszej dzielnicy). Mamy tu więc do czynienia z dyskryminacją na poziomie dzielnic miasta. Najprostszym rozwiązaniem tego problemu jest podział budżetu na dzielnice (proporcjonalnie do liczby mieszkańców lub powierzchni) i utworzenie osobnych instancji wyborczych na terenie każdej z dzielnic, w których głosować mogą tylko mieszkańcy danej dzielnicy. To rozwiązanie tworzy jednak nowe problemy:

- Mieszkańcy zlokalizowani blisko granicy dzielnic mogą chcieć głosować na projekty z sąsiedniej dzielnicy, ponieważ mają do niej bliżej.
- Mieszkańcy pracujący w innej dzielnicy mogliby chcieć poprzeć projekty z dzielnicy w której pracują, gdyż spędzają w niej dużo czasu.
- Projekty przebiegające przez wiele dzielnic (np. ścieżka rowerowa) są trudne do sklasyfikowania, a w przypadku przypisania do jednej z dzielnic tracą wiele potencjalnych wyborców z innych dzielnic.

Podczas wieloletniego korzystania z metody zachłannej zgłaszający projekty zauważyli również inną jej własność - koszt projektu jest niemalże niezależny od szansy bycia wybranym. Zgłaszający nie muszą zatem dbać o dokładne estymowanie kosztów realizacji projektu. Nic nie stoi na przeszkodzie, aby te koszty przeszacować - w sytuacji w której taki projekt zostanie wybrany, przeszacowane koszty nie trafią na realizację pozostałych projektów, co skutkuje mniejszą ilością wybranych projektów i mniejszym średnim zadowoleniem wszystkich wyborców.

W ostatnich latach starano się znaleźć rozwiązanie, które zniwelowałoby lub wyeliminowałoby problemy tworzone przez zachłanny sposób wyłaniania projektów. W ten sposób powstało wiele nowych metod wyłaniania zwycięskiej alokacji o lepszych własnościach matematycznych. Jedną z nich jest metoda równych udziałów, której głównym celem jest zapewnienie proporcjonalności wybranych projektów względem zarówno liczby oddanych na dany projekt głosów jak i jego kosztu [PPS22]. Zapewnienie proporcjonalności w wielu metrykach daje średnio lepsze zadowolenie wśród wszystkich wyborców. Zadowolenie wyborcy można mierzyć chociażby po liczbie lub sumarycznym koszcie popieranych przez niego projektów, które znalazły się w finalnej alokacji.

Głównym filarem metody równych udziałów jest podział budżetu pomiędzy wszystkich wyborców. Przykładowo, w wyborach z dostępnym budżetem na poziomie 1000000zł, w którym wzięło udział 1000 wyborców, każdy z nich otrzymuje wirtualne 1000zł, które będzie mógł przeznaczyć wraz z innymi głosującymi na popierane przez siebie projekty. Jeśli grupa zwolenników projektu nie będzie mieć sumarycznie wystarczającej kwoty, aby pokryć koszt danego projektu, nie zostanie on wybrany. W konsekwencji projekty, których koszt stanowi znaczący ułamek ogólnodostępnego budżetu, będą musiały uzyskać przynajmniej taki sam ułamek poparcia wśród wszystkich głosujących (dla projektu kosztującego 50% całego budżetu, konieczne będzie zdobycie przynajmniej 50% aprobat wśród głosujących). Ta własność znacząco zrównuje szanse małych i dużych projektów w wyborach - duże projekty są często popularniejsze wśród wyborców, ale ze względu na ich duży koszt muszą przekroczyć pewien próg głosów proporcjonalny do ich kosztu. Małe projekty często są ukierunkowane na pewną grupę wyborców, przez co otrzymują mniej głosów, ale dzięki metodzie równych udziałów nie muszą przejmować się pokonywaniem innych projektów w bezpośrednim starciu. Często bowiem wystarczy, aby zwolennicy projektu byli w stanie pokryć koszt realizacji z pieniędzy, które otrzymali w trakcie podziału budżetu.

Podział budżetu pomiędzy wszystkich wyborców pozwala również uniknąć problemu tyranii większości, który opisany został wyżej. Przypuśćmy, że mamy do czynienia z tą samą sytuacją tyranii ze względu na kategorię, gdzie 40% głosujących zagłosowało na projekty z tematyki rozbudowy parku, a pozostałe 60% rozdzieliło się równomiernie pomiędzy projekty z kategorii infrastruktury rowerowej i poprawy warunków życia bezpańskich zwierząt. W takiej sytuacji, stosując metodę równych udziałów, zwolennikom projektów dotyczących rozbudowy parków będzie przysługiwać jedynie 40% ogólnodostępnego budżetu. Pozostałe 60% pozostanie w "rękach" pozostałych wyborców, którzy przeznaczą te kwoty na pokrycie projektów z ich kategorii. W ten sam sposób zanika problem tyranii większości spowodowanej nierównolicznymi dzielnicami.

Profile wyborcze są jednak z reguły znacznie bardziej chaotyczne i mniej ustrukturyzowane. Korzystając z metody równych udziałów może zdarzyć się tak, że spora część ogólnodostępnego budżetu nie zostanie wykorzystana. Dzieje się tak, ponieważ u wielu wyborców część wirtualnych pieniędzy, które zostały im przydzielone, nie wystarcza na pokrycie niektórych projektów. Innymi słowy, po kilku rundach, w których zostały wybrane niektóre projekty, na wirtualnych kontach zwolenników pozostałych projektów pozostaje pewna nie wydana kwota, jednak sumarycznie żadna grupa zwolenników nie jest w stanie "złożyć się" na pokrycie tychże projektów. W tej sytuacji stosuje się metody dopełniające. Ich celem jest rozdysponowanie niewykorzystanego przez metodę budżetu pomiędzy projektami nie znajdującymi się w alokacji. Powstało kilka dopełnień metody równych udziałów. Najprostsza z nich jest wybranie pozostałych projektów w sposób zachłanny. Można też starać się wybrać projekty które zmaksymalizują średnie zadowolenie wyborców. Okazuje się jednak, że najlepsze własności zachowuje dopełnienie opierające się na samej metodzie równych udziałów, nazwanej skrótowo ADD-1. Polega ona na iteracyjnym uruchamianiu metody równych udziałów, za każdym razem zwiększając ogólnodostępny budżet o pewną stałą. Dzięki tej zmianie w każdym kolejnym kroku wyborcy otrzymują więcej "wirtualnych" pieniędzy, co pozwala na pokrycie większej ilości projektów. Aby sumaryczny koszt zwycięskiej alokacji nie przekroczył oryginalnie ustalonego budżetu, ADD-1 na koniec każdej iteracji sprawdza wyłonioną alokację względem dwóch warunków stopu. Pierwszy z nich zatrzymuje pętlę w momencie, kiedy wyłoniona alokacja jest wyczerpująca. Alokacja jest wyczerpująca, kiedy różnica oryginalnego budżetu i sumarycznego kosztu alokacji jest mniejsza od kosztu każdego projektu nie znajdującego się w alokacji. Drugi warunek stopu sprawdza, czy alokacja jest niespełnialna, czyli, czy sumaryczny koszt projek-

tów w alokacji jest większy od oryginalnego budżetu. Jeśli tak jest, to zwracana jest alokacja wyprodukowana przez krok poprzedni, która musi być spełnialna.

Najczęściej stosowanym narzędziem, które umożliwia obliczenie wyników wyborów na podstawie ich instancji i profili wyborczych, jest biblioteka [Pabutools](#) [Fal+23]. Napisana przy użyciu języka Python biblioteka skupia w sobie wszelkie funkcje potrzebne do sprawnego wyłonięcia zwycięskich projektów, niezależnie od stopnia skomplikowania parametrów wejściowych (takich jak format oddawania głosów czy sposób rozwiązywania problematycznych remisów). Biblioteka wspiera nie tylko metodę równych udziałów, ale także inne reguły wyboru zwycięzców, między innymi metodę Phragmen’a, użyteczną czy też metodę zachłanną. Pabutools udostępnia również funkcje, które na podstawie instancji, profilu i wyników wyborów są w stanie zwrócić podstawowe dane statystyczne. Wśród nich można wyróżnić między innymi funkcję pozwalającą na obliczenie średniego zadowolenia wyborców (na podstawie różnych metryk), czy też macierz przepływu głosów, która pokazuje, jak dużo wyborców głosujących na dowolny projekt  $A$  zagłosowało również za dowolnym projektem  $B$ .

W tym miejscu warto wspomnieć, w jaki sposób Pabutools pozwala na łatwe wyliczanie wyników wyborów bez potrzeby ręcznego konstruowania instancji i profilu wyborczego z surowych danych. Biblioteka udostępnia funkcje pozwalające na automatyczne generowanie tych dwóch obiektów z pliku w formacie [Pabulib](#) [Fal+23]. Format ten to specjalny standard, stworzony, aby ułatwić udostępnianie danych o wyborach budżetu partycypacyjnego. Pliki w tym formacie, nazywane często plikami .pb, zawierają wszystkie konieczne informacje do stworzenia instancji wyborczej i profilu wyborczego. Plik ten można podzielić na trzy osobne sekcje:

- Metadane, zbierające wszelkie meta informacje o wyborach, takie jak jego nazwa, opis, język, waluta, budżet, liczba głosów i liczba projektów, sposób oddawania głosów i jego ograniczenia (np. minimalna liczba aprobat, które wyborca musiał przyznać)
- Projekty, czyli listę wszystkich projektów w instancji, łącznie z ich nazwami, kosztem oraz liczbą oddanych na nie głosów/zdobytych punktów
- Głosy, czyli lista wszystkich wyborców oraz oddanych przez nich głosów, reprezentowana odpowiednio do sposobu oddawania głosów.

```

META
key;value
description;Municipal PB in Example
country;Poland
instance;2023
budget;1000
vote_type;approval
min_length;2
PROJECTS
project_id;name;cost
1;Budki łęgowe;100
2;Schronisko;300
3;Sprzęt strażacki;400
4;Skatepark;800
5;Parking rowerowy;500
6;Pomoc ubogim;50
VOTES
voter_id;vote
1;5,4,6,2
2;3,6,2
3;5,4,3
4;5,4,3
5;5,2
6;6,4,2,1
7;5,3,2
8;5,4,1
9;5,4,2
10;5,3,1

```

Rysunek 1.1: Przykładowy plik w formacie Pabulib.

Przestudiujemy strukturę tego pliku i zobaczymy, jak możemy z niego zrekonstruować instancję i profil wyborczy. Z pierwszej sekcji o metadanych możemy wywnioskować, że jest to głosowanie o nazwie "Municipal PB In Example", które odbyło się w Polsce w 2023 roku. Budżet tych wyborów wynosił 1000zł, a wyborcy wybierali projekty poprzez aprobaty. Dodatkowo, byli zobowiązani do oddania co najmniej 2 aprobat. W kolejnej sekcji możemy uzupełnić instancję wyborczą informacją o projektach - w tym głosowaniu wyborcy wybierali spośród 6 projektów:

- Budki łęgowe, o koszcie 100zł i identyfikatorze 1.
- Schronisko, o koszcie 300zł i identyfikatorze 2.
- Sprzęt strażacki, o koszcie 400zł i identyfikatorze 3.
- Skatepark, o koszcie 800zł i identyfikatorze 4.
- Parking rowerowy, o koszcie 500zł i identyfikatorze 5.
- Pomoc ubogim, o koszcie 50zł i identyfikatorze 6.

Z ostatniej sekcji możemy zrekonstruować profil wyborczy. W wyborach głos oddało dziesięciu wyborców. Dwóch z nich oddało 4 aprobaty, Jeden z nich oddał 2 aprobaty, a reszta oddała

3 aprobaty. Parując identyfikatory projektów z listą identyfikatorów dostępnej przy każdym z głosujących, możemy odszyfrować poszczególne głosy. Dla przykładu, osoba z identyfikatorem 1 zagłosowała na parking rowerowy, skatepark, pomoc ubogim i schronisko, a osoba z identyfikatorem 10 zagłosowała na parking rowerowy, sprzęt strażacki i budki łęgowe.

Dzięki coraz powszechniejszemu dostępowi do technologii, coraz częściej możemy zaobserwować odbywanie się wyborów za pomocą specjalnych platform do głosowania. Nie inaczej jest też z wyborami na budżet partycypacyjny, gdzie znacząca część miast w Europie i Ameryce Północnej korzysta z takich rozwiązań. Wśród najpopularniejszych można tu wymienić platformę [Decidim](#) [Bar+24], wybieraną chętniej przez Europejczyków, czy [Stanford Participatory Budgeting Platform](#), będąca popularniejsza w Stanach Zjednoczonych i rozwijana przez zespół z Uniwersytetu Stanforda. Stanford Participatory Budgeting Platform umożliwia przeprowadzenie wyborów na budżet partycypacyjny od początku do końca, zaczynając od utworzenia instancji wyborczej, zebrania internetowych głosów i dodania głosów oddanych lokalnie, aż do podsumowania wyborów i wyłonienia zwycięskiej alokacji. Pozwala przy tym administratorom na anonimowy podgląd wyników i ich eksport do prostych plików w formacie CSV. Platforma ta nie umożliwia jednak na zmianę metody z jaką wyłaniani są zwycięzcy. W praktyce więc organizacja wyborów, w której organizatorzy chcieliby skorzystać z bardziej zaawansowanych metod wyznaczania alokacji, jest w tej platformie utrudniona. Jest to ograniczające zarówno dla urzędów miast, które mają mniejszy wybór wśród platform, które obsługują ich metody wyłaniania zwycięzców, jak i samej platformy, która nie może zostać w pełni wykorzystana przez niektóre miasta. Rozwinięcie tego narzędzia pozwoliłoby na łatwiejsze organizowanie wyborów z użyciem zaawansowanych metod wyborczych, jak i przyczyniło się do większej popularyzacji tychże metod i ich zalet.

Metoda równych udziałów, pomimo wielu zalet nad metodą zachłanną, posiada też swoje wady. Główną z nich jest jej skomplikowanie - opisanie wszystkich jej aspektów zwykłego wyborcy może być trudne. Brak zrozumienia sposobu, w jaki wyłaniani są zwycięzcy, może doprowadzić część wyborców do zadania sobie pytania, dlaczego projekt, który uzyskał mniejszą liczbę głosów, został wybrany nad projektem, który uzyskał tych głosów więcej, czy też dlaczego niektóre projekty, których dany wyborca był zwolennikiem, zostały wybrane, a inne nie. Niestety nie istniało rozwiązanie, które w łatwy sposób umożliwiłoby ekstrakcję i wizualizację danych statystycznych, które mogłyby pomóc rozwiązać wątpliwości zwykłych wyborców czy administratorów wyborów budżetu partycypacyjnego. Stworzenie takiego rozwiązania zdecydowanie poprawiłoby świadomość osób zaangażowanych w budżet obywatelski. Pozwoliłoby ono na znacząco łatwiejsze i klarowniejsze przedstawienie działania metody równych udziałów administratorom i wyborcom, a także umożliwiło tworzenie łatwych do zrozumienia raportów, w których każdy zainteresowany mógłby przeczytać, dlaczego dany projekt został lub nie został wybrany do realizacji.

Statystyką, która mogłaby pomóc w zrozumieniu, dlaczego dany projekt został lub nie został wybrany, jest analiza sumarycznego budżetu jego zwolenników podczas działania metody. W wyborach, w których wyborcy mogą poprzeć więcej niż jeden projekt, budżety zwolenników dowolnego projektu  $A$  dynamicznie się zmieniają, ponieważ co rundę pewien inny projekt  $B$  może zostać wybrany, a wtedy wyborcy popierający zarówno projekty  $A$  jak i  $B$  wydają część swego budżetu na pokrycie projektu  $B$  - budżetu, który mógł zostać wykorzystany na pokrycie projektu  $A$ . Śledzenie wydatków wszystkich wyborców i grupowanie ich względem projektów, które są przez nich popierane, pozwala w pełni zobrazować stan budżetu zwolenników dowolnego projektu w każdej rundzie metody równych udziałów. Dokładając do tego informację o budżecie zwolenników projektu  $B_i$  na początku działania metody, daną wzorem

$BS_i = \text{początkowy przydział budżetu jednego wyborcy} * \text{liczba zwolenników projektu } B_i$

oraz informację o budżecie zwolenników danego projektu podczas jego rozważania przez metodę, daną wzorem

$$BE_i = BS_i - \text{suma wszystkich wydatków zwolenników przed rozważeniem projektu}$$

pomagamy uzasadnić kolejność rozważania projektów, a także pokazujemy, jak duży deficyt/zapas budżetu miał projekt niewybrany/wybrany. Przykładowo więc, w wyborach z dostępnym budżetem w wysokości 1000zł i dwoma projektami:  $A$  kosztujący 600zł i  $B$  kosztujący 500zł, oraz profilem w którym 5 wyborców zagłosowało na  $A$ , 2 wyborców zagłosowało na  $B$  a 3 wyborców zagłosowało na oba projekty:

- Początkowy budżet zwolenników  $A$  to 800zł, a  $B$  to 500zł.
- $A$  został wybrany do rozważenia jako pierwszy (większe poparcie względem kosztu).
- $A$  został zrealizowany, każdy z jego zwolenników zapłacił po 75zł.
- $B$  został wybrany do rozważenia jako drugi i nie został zrealizowany, ponieważ 3 jego zwolenników utraciło już 75zł na rzecz projektu  $A$ .
- Budżet zwolenników  $A$  podczas jego rozważenia to dalej 800zł, natomiast dla  $B$  jest to tylko 275zł ( $500\text{zł} - 75\text{zł} * 3$ ).

Ciekawą statystyką jest też efektywne poparcie. Efektywne poparcie projektu pozwala określić, jak blisko realizacji był projekt, który nie został wybrany do zwycięskiej alokacji, lub w przypadku projektów wybranych do realizacji - jak duża była nadwyżka głosów danego projektu. Innymi słowy, efektywne poparcie pozwala przedstawić, ile głosów zabrakło projektowi niezrealizowanemu do realizacji, a projektowi zrealizowanemu - ile głosów mógł stracić i dalej zostać wybranym. Metryka ta jest reprezentowana w procentach, gdzie projekt o efektywnym poparciu na poziomie 100% został wybrany, ale gdyby utracił choć jeden punkt procentowy z budżetu jego zwolenników, nie został by wtedy wybrany przez metodę równych udziałów. Wszystkie projekty, których efektywne poparcie jest mniejsze od 100%, nie są częścią zwycięskiej alokacji. Przeciwnie, projekty o efektywnym poparciu większym lub równym 100% są częścią zwycięskiej alokacji. Przykładowo więc, jeśli projekt w wyborach przez aprobaty uzyskał efektywne poparcie na poziomie 50%, to mógłby zostać wybrany, gdyby zagłosowało na niego dwa razy więcej osób. Jeśli zaś efektywne poparcie projektu wynosi 300%, to projekt ten mógłby utracić  $\frac{2}{3}$  swoich głosów, a nadal zostałby wybrany do realizacji.

Obie z wyżej przedstawionych statystyk nie są w żaden sposób wyliczane przez jakiekolwiek ogólnodostępne narzędzie. W szczególności Pabutools, które jest najbardziej popularnym narzędziem otwartego dostępu w świecie reguł budżetu partycypacyjnego, pomimo dostarczania szeregu metod do wyliczenia danych statystycznych, nie wylicza tych istotnych z punktu widzenia zwykłego wyborcy statystyk. Nawet jeśli takie statystyki byłyby wyliczane, nie istniałby żaden sposób na ich wizualizację w schludny sposób. Finalnie zatem celem było nie tylko umożliwienie wyliczania tychże statystyk, ale też stworzenie narzędzia, które zapewni przyjazny interfejs umożliwiający podgląd wyliczonych danych.



## Rozdział 2

# Architektura rozwiązania

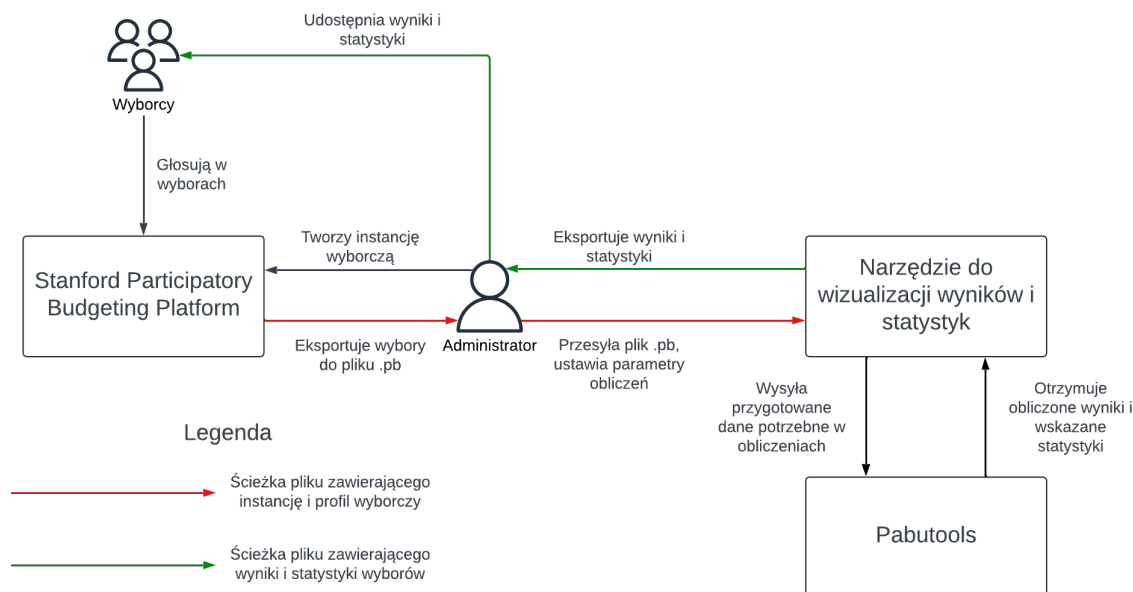
W tym rozdziale skupimy się na architekturze rozwiązania do przedstawionych wcześniej problemów. Rozwiązanie dotyka wielu obszarów całego procesu przeprowadzania wyborów i wyliczania ich wyników.

Całość rozpoczyna się w Stanford Participatory Budgeting Platform, w którym administrator tworzy instancję wyborczą i ustawia sposób wyrażania poparcia na konkretne projekty. Po przeprowadzeniu wyborów i uzupełnieniu profilu wyborczego, Stanford Participatory Budgeting Platform powinien udostępnić możliwość wyliczenia wyników korzystając z bardziej zaawansowanej metody wyborczej. Z uwagi na różnicę w stosie technologicznym platformy do głosowania (Ruby on Rails) i biblioteki obsługującej wyliczanie rezultatów dla bardziej zaawansowanych metod wyborczych (Python), łatwe podpięcie biblioteki do rozwiązania nie było możliwe. Aby uniknąć rozpraszania kodu metod wyborczych poprzez ich bezpośrednią implementację w platformie lub oddzielnej bibliotece dla języka Ruby, najlepszą decyzją architektoniczną jest implementacja eksportu wyników do pliku w formacie Pabulib. Dzięki takiemu rozwiązaniu administratorzy mogą z łatwością wyeksportować wyniki dowolnych wyborów i skorzystać z interesujących ich metod dostępnych w bibliotece Pabutools, czy też w rozwiązaniach dostępnych w internecie. Takie podejście łączy w sobie zalety łatwej obsługi dla osób korzystających z platformy jak i scentralizowania metod wyborczych w jednym narzędziu.

Przygotowany plik .pb posłuży administratorowi do obliczenia wyników i wizualizacji statystyk wyborów. Do tego celu użyje on specjalnie przygotowanego narzędzia do wizualizacji wyników i statystyk. Zgodnie z myślą centralizowania rozwiązań, narzędzie to korzysta z biblioteki Pabutools, która to odpowiedzialna jest za wyliczenie wyników i wcześniej opisanych statystyk. Wyliczone dane są w narzędziu wyświetlane dynamicznie i w przyjazny dla użytkownika sposób. Aby zapewnić łatwy i przejrzysty interfejs użytkownikowi końcowemu jak i możliwość połączenia się z biblioteką Pabutools, narzędzie to jest aplikacją internetową, podzieloną na serwer back-endu, pełniącego rolę REST API importującego kod z biblioteki Pabutools, oraz serwer front-endu, odpytującego back-end o dane na podstawie przesłanego pliku w formacie Pabulib i tworzącego z uzyskanych danych prostą do zrozumienia tabelę, którą można wyeksportować do statycznego pliku HTML i CSS. Serwer back-endu został napisany w języku Python, korzystając z zrzębu Django, aby umożliwić proste połączenie się z biblioteką Pabutools. Serwer front-endu jest aplikacją napisaną przy użyciu zrzębu React w języku TypeScript, dzięki czemu aplikacja może bazować na małych i typowanych komponentach. W tym kontekście komponenty nie tylko pozwalają na dynamiczne ładowanie danych, ale także zachowują czystość i strukturalność kodu. Całość została spięta z użyciem narzędzia Docker, który pozwala na łatwą instalację osobom zainteresowanym narzędziem, niezależnie od ich systemu operacyjnego.

Główna logika odpowiedzialna za wyliczanie statystyk została zaimplementowana w bibliotece Pabutools. To tam rozwinięta została metoda równych udziałów tak, aby możliwa była ekstrakcja danych statystycznych opisywanych w poprzednim rozdziale. Pomysłem architektonicznym rozwiązania mogła być bezpośrednia ekstrakcja, czyli zbieranie i wyliczanie danych statystycznych w momencie przebiegu metody równych udziałów. Taka bezpośredniość wymagała jednak implementacji sporej ilości kodu, który wiązał się z ryzykiem pogorszenia zarówno czystości, jak i efektywności kodu składającego się na poszczególne funkcje. Finalnie więc rozwiązanie zostało podzielone na dwie zależne od siebie fazy. Pierwsza z nich skupia się na odzyskaniu danych o przebiegu metody równych udziałów (nazywanych dalej danymi detalicznymi), które umożliwią pełne odwzorowanie wszystkich iteracji wykonanych przez funkcję reprezentującą metodę równych udziałów. Druga z nich to faza obliczeń, która, korzystając z danych detalicznych, wylicza interesujące nas dane statystyczne, bez potrzeby wywoływania funkcji odpowiedzialnych za wyliczanie wyników wyborów. Używając takiego podziału, w istniejącym już kodzie funkcji odpowiedzialnej za metodę równych udziałów wystarczy zaimplementować tylko pierwszą fazę, która w poszczególnych iteracjach zapisuje dane o jej przebiegu, bez konieczności przeprowadzania skomplikowanych i czasochłonnych obliczeń. One realizowane są w drugiej fazie, zaimplementowanej w osobnym pliku i odbywającej się po obliczeniu wyników wyborów. Taki podział gwarantuje nam zachowanie efektywności funkcji do obliczania wyników jak i utrzymanie czystości kodu, zarówno w istniejących jak i nowo utworzonych funkcjach.

Na poniższym rysunku przedstawiona została pełna architektura rozwiązania w formie diagramu. Prostokąty na diagramie reprezentują opisane wyżej części systemu, natomiast strzałki reprezentują akcje zainicjowane przez użytkowników lub części systemu. Dodatkowo, kolorowymi strzałkami przedstawione zostały akcje powiązane z plikami w formacie Pabulib i wyeksportowanym plikiem z wynikami i statystykami.



Rysunek 2.1: Architektura pełnego rozwiązania

## Rozdział 3

# Rozwiązania

W tym rozdziale skupimy się na przedstawieniu rozwiązań wyżej wymienionych problemów oraz podejścia do ich implementacji.

### 3.1. Obliczanie wyników w Stanford Participatory Budgeting Platform

Celem rozwinięcia tej platformy było umożliwienie osobom zarządzającym łatwe i intuicyjne wyliczenie wyników wyborów, korzystając z różnych metod wyborczych, w tym z metody równych udziałów. Zgodnie z rozwiązaniem architektonicznym przedstawionym wcześniej, implementacja skupiała się na umożliwieniu eksportu danych do pliku w formacie .pb. Ponieważ sama platforma pozwala na przeprowadzenie wyborów, korzystając z różnych sposobów oddawania głosów, w tym: głosowanie przez aprobaty, głosowanie plecakowe, głosowanie rankingowe i głosowanie żetonowe, konieczna była implementacja eksportu dla każdego z tych sposobów. Implementacja dla każdego z nich stworzyła nowy przycisk w interfejsie oraz nową funkcję, która wywoływana jest po kliknięciu tego przycisku. Każda z tych funkcji działa z tym samym schematem - z bazy danych wyciąga potrzebne jej informacje, transformuje je, aby spełniały warunki narzucone przez format .pb, następnie generuje informację o metadanych wyborów, dostępnych projektach i oddanych głosach. Z uwagi na duże podobieństwa podczas generacji sekcji metadanych i sekcji projektów, ta część została uwspólniona w jednej prywatnej funkcji. Aby zapewnić lepszą ochronę danych, głosy wyborców w pliku są wstawiane losowo. Prześledźmy, czym różnią się poszczególne funkcje w etapach transformacji i generacji danych o oddanych głosach.

- Głosowanie przez aprobaty
  - W metadanych dodawana jest informacja o minimalnej i maksymalnej liczbie aprobat, które wyborca mógł przyznać.
  - W sekcji projektów, dla kolumny o głosach, sumowana jest liczba aprobat przyznana danemu projektowi.
  - W sekcji głosów, każdego głosującego reprezentuje lista identyfikatorów projektu, na które oddał głos.
- Głosowanie plecakowe
  - W sekcji projektów, dla kolumny o głosach, sumowana jest liczba głosów przyznana danemu projektowi.

- W sekcji głosów, każdego głosującego reprezentuje lista identyfikatorów projektu, na które oddał głos.
- Głosowanie rankingowe
  - W metadanych dodawana jest informacja o minimalnej i maksymalnej liczbie projektów, które wyborca mógł poddać rankingowi. Maksymalna liczba projektów reprezentuje największą liczbę punktów jaka mogła zostać przyznana projektowi.
  - W sekcji projektów, dla kolumny o głosach, sumowana jest punktacja z rankingów wszystkich wyborców.
  - W sekcji głosów, każdego głosującego reprezentuje lista identyfikatorów projektu, na które oddał głos, posortowana w kolejności od najbardziej do najmniej preferowanego.
- Głosowanie żetonowe
  - W metadanych dodawana jest informacja o minimalnej i maksymalnej liczbie żetonów, które wyborca mógł przyznać projektom.
  - W sekcji projektów, dla kolumny o głosach sumowana, jest ilość żetonów przyznana przez wszystkich wyborców.
  - W sekcji głosów, każdego głosującego reprezentują dwie listy o tej samej długości, które parują się indeksami: lista identyfikatorów projektu na które oddał głos oraz lista żetonów, które zostały przyznane temu projektowi.

## 3.2. Ekstrakcja danych statystycznych metody równych udziałów

Ekstrakcja dodatkowych danych z metody równych udziałów jest kluczowym elementem pozwalającym na sporządzenie raportów i statystyk o poszczególnych wyborach, które pomagają zrozumieć sposób działania metody. Zgodnie z myślą scentralizowania rozwiązań, implementacja kodu umożliwiająca taką ekstrakcję odbyła się w publicznej bibliotece Pabutools. Biblioteka ta posiada funkcję pozwalającą na wyliczenie wyników wyborów korzystając z metody równych udziałów, zarówno w wariantcie zwykłym jak i wariantcie wyczerpującym ADD-1. To z tych funkcji należało zwrócić dodatkowe dane statystyczne. Tak jak zostało to wspomniane w rozdziale opisującym architekturę rozwiązania, mamy tu do czynienia z dwoma następującymi po sobie fazami. W kolejnych podrozdziałach oddzielnie przeanalizujemy rozwiązania użyte w tych fazach, aby lepiej zobrazować pełne rozwiązanie ekstrakcji danych statystycznych.

### 3.2.1. Faza ekstrakcji danych detalicznych

Pierwszym krokiem do wprowadzenia możliwości ekstrakcji danych detalicznych z metod wyborczych jest umożliwienie wszelakim funkcjom reprezentującym te metody na zwrócenia takich danych. W poprzedniej wersji biblioteki wynikiem wszystkich funkcji reprezentujących metody wyborcze była lista projektów, która reprezentowała wybrane przez metodę projekty do realizacji. Ponieważ metody wyborcze to jedna z najważniejszych funkcjonalności tej biblioteki, chciałem zachować jak największą kompatybilność wsteczną i nie zmieniać typu zwracanych danych. Stworzyłem więc nową klasę, która dziedziczyła po liście. Klasa ta w swojej liście przechowuje zwycięskie projekty, ale oprócz tego posiada również atrybut poświęconym detalom przebiegu metody. Dzięki użyciu dziedziczenia po liście rozwiązanie zapewnia kompatybilność wsteczną.

Dane o przebiegu metody są reprezentowane przez klasę, w której najważniejszym atrybutem jest lista iteracji. Iteracje te również mają oddzielną klasę i ich zadaniem jest jak najlepsze odwzorowanie faktycznych iteracji które odbywają się podczas biegu metody. Pojedyncza iteracja zawiera następujące atrybuty:

- Listę projektów dostępnych w tej iteracji (które nie zostały jeszcze wybrane ani odrzucone).
- Projekt wybrany do realizacji w tej iteracji (może być pusty w przypadku ostatniej iteracji, jeśli ta iteracja odrzuciła wszystkie pozostałe projekty).
- Dostępny budżet wszystkich wyborców przed wybraniem projektu.
- Dostępny budżet wszystkich wyborców po wyborze projektu.

Klasę reprezentującą projekty w liście jest specjalna klasa dziedzicząca po głównej klasie projektu. W klasie dziedziczącej zdefiniowana jest flaga, pozwalająca oznaczać projekty jako odrzucone. Umożliwia to filtrację projektów odrzuconych w konkretnej iteracji. Flaga ta jest ustawiana przez metodę równych udziałów podczas odrzucenia danego projektu. Projekt wybrany do realizacji jest reprezentowany zwykłą klasą projektu i jest zapisywany na końcu danej iteracji, podobnie jak dostępny budżet wszystkich wyborców po wyborze projektu. Dostępny budżet wszystkich wyborców przed wybraniem projektu jest ustawiany na początku danej iteracji.

Z racji na to, że dane detaliczne nie są konieczne w każdej sytuacji, do argumentów metody równych udziałów dodana została flaga, która decyduje, czy dany przebieg powinien zapisywać swoje dane detaliczne czy też nie.

Dzięki tak zaprojektowanemu rozwiązaniu nie wymagane są żadne zmiany w metodzie wyczerpującej ADD-1. Jako, że metoda ADD-1 operuje na funkcji reprezentującej metodę równych udziałów i zwraca ostatnią spełnialną alokację lub pierwszą wyczerpującą alokację wyprodukowaną przez metodę równych udziałów, sama metoda ADD-1 dodatkowo zwróci też dane detaliczne, o ile wyżej wspomniana flaga zostanie ustawiona.

Aby lepiej zobrazować działanie ekstrakcji danych detalicznych, zwięźle opiszemy jego działanie na przykładzie.

### Przykład ekstrakcji danych detalicznych

Budżet: 1000zł

| Projekt          | Koszt | Głos  | Głos  | Głos  | Głos  | Głos  | Głos  | Głos  | Głos  | Głos  | Głos  |
|------------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| Parking rowerowy | 500zł | X     |       | X     | X     | X     |       | X     | X     | X     | X     |
| Skatepark        | 800zł | X     |       | X     | X     |       | X     |       | X     | X     |       |
| Schronisko       | 300zł | X     | X     |       |       | X     | X     | X     |       | X     |       |
| Budki łęgowe     | 100zł |       |       |       |       |       | X     |       | X     |       | X     |
| Pomoc ubogim     | 50zł  | X     | X     |       |       |       | X     |       |       |       |       |
| Sprzęt strażacki | 400zł |       | X     | X     | X     |       |       | X     |       |       | X     |
| Pozostały budżet | -     | 100zł | 100zł | 100zł | 100zł | 100zł | 100zł | 100zł | 100zł | 100zł | 100zł |

Tabela 3.1: Przykładowe wybory budżetu partycypacyjnego.

Metoda równych udziałów w pierwszej iteracji zauważy, że projekt 'Skatepark' ma za mało głosów, aby zostać zrealizowanym, natomiast jako zwycięzcę wybierze projekt 'Parking rowerowy'. W danych detalicznych zapisze więc 'Parking rowerowy' jako wybrany projekt. Do listy budżetu wyborców przed wyborem projektu dopisze wartość 100zł dla każdego wyborcy, natomiast do listy budżetu wyborców po wyborze projektu każdej osobie, która zagłosowała na 'Parking rowerowy', zostanie odjęte  $500\text{zł}/8 = 62.5\text{zł}$ . Na liście pozostałych projektów 'Skatepark' zostanie oznaczony jako odrzucony.

Budżet: 1000zł

| Projekt          | Koszt | Głos    | Głos  | Głos    | Głos    | Głos    | Głos  | Głos    | Głos    | Głos    | Głos  |
|------------------|-------|---------|-------|---------|---------|---------|-------|---------|---------|---------|-------|
| Schronisko       | 300zł | X       | X     |         |         | X       | X     | X       |         | X       |       |
| Budki łęgowe     | 100zł |         |       |         |         |         | X     |         | X       |         | X     |
| Pomoc ubogim     | 50zł  | X       | X     |         |         |         | X     |         |         |         |       |
| Sprzęt strażacki | 400zł |         | X     | X       | X       |         |       | X       |         |         | X     |
| Pozostały budżet | -     | 37.50zł | 100zł | 37.50zł | 37.50zł | 37.50zł | 100zł | 37.50zł | 37.50zł | 37.50zł | 100zł |

Tabela 3.2: Stan obliczeń po 1 iteracji.

| Numer iteracji | Wybrany projekt  | Odrzucone projekty |
|----------------|------------------|--------------------|
| 1              | Parking rowerowy | Skatepark          |

| Iteracja | Rodzaj listy         | Wyb 1   | Wyb 2 | Wyb 3   | Wyb 4   | Wyb 5   | Wyb 6 | Wyb 7   | Wyb 8   | Wyb 9   | Wyb 10 |
|----------|----------------------|---------|-------|---------|---------|---------|-------|---------|---------|---------|--------|
| 1        | Budżet przed wyborem | 100zł   | 100zł | 100zł   | 100zł   | 100zł   | 100zł | 100zł   | 100zł   | 100zł   | 100zł  |
| 1        | Budżet po wyborze    | 37.50zł | 100zł | 37.50zł | 37.50zł | 37.50zł | 100zł | 37.50zł | 37.50zł | 37.50zł | 100zł  |

Tabela 3.3: Dane detaliczne po 1 iteracji.

W kolejnej iteracji, metoda wybierze 'Schronisko' jako projekt zwycięski. Oprócz tego, metoda zauważy, że projekt 'Sprzęt strażacki' jest już nierealizowalny z aktualnym budżetem jego zwolenników. Budżet początkowy tej iteracji będzie budżetem końcowym poprzedniej iteracji. Aby otrzymać budżet po wybraniu projektów, należy od każdego zwolennika projektu 'Schronisko' odjąć kwotę, jaką na ten projekt przeznaczył.

Budżet: 1000zł

| Projekt          | Koszt | Głos | Głos | Głos    | Głos    | Głos | Głos | Głos | Głos    | Głos | Głos    |
|------------------|-------|------|------|---------|---------|------|------|------|---------|------|---------|
| Budki łęgowe     | 100zł |      |      |         |         |      | X    |      | X       |      | X       |
| Pomoc ubogim     | 50zł  | X    | X    |         |         |      | X    |      |         |      |         |
| Pozostały budżet | -     | 0zł  | 25zł | 37.50zł | 37.50zł | 0zł  | 25zł | 0zł  | 37.50zł | 0zł  | 37.50zł |

Tabela 3.4: Stan obliczeń po 2 iteracji.

| Numer iteracji | Wybrany projekt  | Odrzucone projekty |
|----------------|------------------|--------------------|
| 1              | Parking rowerowy | Skatepark          |
| 2              | Schronisko       | Sprzęt strażacki   |

| Iteracja | Rodzaj listy         | Wyb 1   | Wyb 2 | Wyb 3   | Wyb 4   | Wyb 5   | Wyb 6 | Wyb 7   | Wyb 8   | Wyb 9   | Wyb 10  |
|----------|----------------------|---------|-------|---------|---------|---------|-------|---------|---------|---------|---------|
| 1        | Budżet przed wyborem | 100zł   | 100zł | 100zł   | 100zł   | 100zł   | 100zł | 100zł   | 100zł   | 100zł   | 100zł   |
| 1        | Budżet po wyborze    | 37.50zł | 100zł | 37.50zł | 37.50zł | 37.50zł | 100zł | 37.50zł | 37.50zł | 37.50zł | 100zł   |
| 2        | Budżet przed wyborem | 37.50zł | 100zł | 37.50zł | 37.50zł | 37.50zł | 100zł | 37.50zł | 37.50zł | 37.50zł | 100zł   |
| 2        | Budżet po wyborze    | 0zł     | 25zł  | 37.50zł | 37.50zł | 0zł     | 25zł  | 0zł     | 37.50zł | 0zł     | 37.50zł |

Tabela 3.5: Dane detaliczne po 2 iteracji.

W kolejnej iteracji, z racji na wyższą efektywną liczbę głosów, zostanie wybrany projekt 'Budki łęgowe'. Pozostały projekt 'Pomoc ubogim' nie zostanie jeszcze odrzucony. Ponownie budżet wyborców przed wyborem projektu jest równy budżetowi wyborców po wyborze projektu z poprzedniej iteracji, a żeby otrzymać budżet wyborców po wybraniu projektu odejmowana jest kwota, jaką każdy ze zwolenników przeznaczył na wybrany projekt.

Budżet: 1000zł

| Projekt          | Koszt | Głos | Głos | Głos    | Głos    | Głos | Głos | Głos | Głos | Głos | Głos |
|------------------|-------|------|------|---------|---------|------|------|------|------|------|------|
| Pomoc ubogim     | 50zł  | X    | X    |         |         |      | X    |      |      |      |      |
| Pozostały budżet | -     | 0zł  | 25zł | 37.50zł | 37.50zł | 0zł  | 0zł  | 0zł  | 0zł  | 0zł  | 0zł  |

Tabela 3.6: Stan obliczeń po 3 iteracji.

| Numer iteracji | Wybrany projekt  | Odrzucone projekty |
|----------------|------------------|--------------------|
| 1              | Parking rowerowy | Skatepark          |
| 2              | Schronisko       | Sprzęt strażacki   |
| 3              | Budki łęgowe     | -                  |

| Iteracja | Rodzaj listy         | Wyb 1   | Wyb 2 | Wyb 3   | Wyb 4   | Wyb 5   | Wyb 6 | Wyb 7   | Wyb 8   | Wyb 9   | Wyb 10  |
|----------|----------------------|---------|-------|---------|---------|---------|-------|---------|---------|---------|---------|
| 1        | Budżet przed wyborem | 100zł   | 100zł | 100zł   | 100zł   | 100zł   | 100zł | 100zł   | 100zł   | 100zł   | 100zł   |
| 1        | Budżet po wyborze    | 37.50zł | 100zł | 37.50zł | 37.50zł | 37.50zł | 100zł | 37.50zł | 37.50zł | 37.50zł | 100zł   |
| 2        | Budżet przed wyborem | 37.50zł | 100zł | 37.50zł | 37.50zł | 37.50zł | 100zł | 37.50zł | 37.50zł | 37.50zł | 100zł   |
| 2        | Budżet po wyborze    | 0zł     | 25zł  | 37.50zł | 37.50zł | 0zł     | 25zł  | 0zł     | 37.50zł | 0zł     | 37.50zł |
| 3        | Budżet przed wyborem | 0zł     | 25zł  | 37.50zł | 37.50zł | 0zł     | 25zł  | 0zł     | 37.50zł | 0zł     | 37.50zł |
| 3        | Budżet po wyborze    | 0zł     | 25zł  | 37.50zł | 37.50zł | 0zł     | 0zł   | 0zł     | 0zł     | 0zł     | 0zł     |

Tabela 3.7: Dane detaliczne po 3 iteracji.

W ostatniej iteracji metoda zauważyła, że zwolennikom projektu 'Pomoc ubogim' brakuje budżetu na jego pokrycie, dlatego odrzuca ten projekt. Nie zostaje wybrany żaden inny projekt, a budżety wyborców pozostają niezmienione.

| Numer iteracji | Wybrany projekt  | Odrzucone projekty |
|----------------|------------------|--------------------|
| 1              | Parking rowerowy | Skatepark          |
| 2              | Schronisko       | Sprzęt strażacki   |
| 3              | Budki legowe     | -                  |
| 4              | -                | Pomoc ubogim       |

| Iteracja | Rodzaj listy         | Wyb 1   | Wyb 2 | Wyb 3   | Wyb 4   | Wyb 5   | Wyb 6 | Wyb 7   | Wyb 8   | Wyb 9   | Wyb 10  |
|----------|----------------------|---------|-------|---------|---------|---------|-------|---------|---------|---------|---------|
| 1        | Budżet przed wyborem | 100zł   | 100zł | 100zł   | 100zł   | 100zł   | 100zł | 100zł   | 100zł   | 100zł   | 100zł   |
| 1        | Budżet po wyborze    | 37.50zł | 100zł | 37.50zł | 37.50zł | 37.50zł | 100zł | 37.50zł | 37.50zł | 37.50zł | 100zł   |
| 2        | Budżet przed wyborem | 37.50zł | 100zł | 37.50zł | 37.50zł | 37.50zł | 100zł | 37.50zł | 37.50zł | 37.50zł | 100zł   |
| 2        | Budżet po wyborze    | 0zł     | 25zł  | 37.50zł | 37.50zł | 0zł     | 25zł  | 0zł     | 37.50zł | 0zł     | 37.50zł |
| 3        | Budżet przed wyborem | 0zł     | 25zł  | 37.50zł | 37.50zł | 0zł     | 25zł  | 0zł     | 37.50zł | 0zł     | 37.50zł |
| 3        | Budżet po wyborze    | 0zł     | 25zł  | 37.50zł | 37.50zł | 0zł     | 0zł   | 0zł     | 0zł     | 0zł     | 0zł     |
| 4        | Budżet przed wyborem | 0zł     | 25zł  | 37.50zł | 37.50zł | 0zł     | 0zł   | 0zł     | 0zł     | 0zł     | 0zł     |
| 4        | Budżet po wyborze    | 0zł     | 25zł  | 37.50zł | 37.50zł | 0zł     | 0zł   | 0zł     | 0zł     | 0zł     | 0zł     |

Tabela 3.8: Dane detaliczne po 4 iteracji.

### 3.2.2. Faza wyliczeń danych statystycznych

Na podstawie danych detalicznych możemy wyliczyć wszelkiego rodzaju dane statystyczne. Są one zaimplementowane jako oddzielne funkcje, których głównym argumentem są zwrócone przez metodę równych udziałów dane detaliczne. W ramach tej pracy zaimplementowane zostały dwie funkcje:

- Funkcja odpowiedzialna za wyliczenie statystyk skoncentrowanych wokół budżetów zwolenników każdego z projektów.
- Funkcja odpowiedzialna za wyliczenie efektywnego poparcia każdego z projektów.

Przedstawię teraz rozwiązania użyte w obu tych funkcjach.

#### Funkcja do wyliczenia statystyk dotyczących budżetów zwolenników

Zaimplementowana funkcja korzysta jedynie z wyliczonych wcześniej danych detalicznych i to z nich wylicza wszelkie statystyki. Wśród tych statystyk znajdują się:

- Budżet zwolenników projektu w trakcie jego rozważania przez metodę.
- Lista wszystkich projektów wybranych przed rozważaniem danego projektu  $A$  oraz informacja, ile każdy z tych projektów zabrał z budżetu zwolenników projektu  $A$ .
- Suma całego straconego budżetu na rzecz wcześniej wybranych projektów, wyliczana za pomocą wyżej wspomnianej listy.

Te wartości są zagregowane w pojedynczej klasie i są obliczane dla wszystkich projektów. Funkcja zwraca listę klas ze statystykami wspomnianymi wyżej i będzie je wyliczać przechodząc po iteracjach zapisanych w danych detalicznych.

Pierwszym krokiem tej funkcji jest zainicjalizowanie tabeli śledzącej wydatki wszystkich głosujących. Kiedy głosujący pokryje koszt jakiegoś projektu, do tabeli śledzącej wydatki pod jego indeksem zostanie dodana para (projekt, ilość przeznaczonych pieniędzy na ten projekt). Oprócz tej tabeli istotna jest również informacja o indeksach zwolenników dla każdego projektu. Jest to informacja przechowywana w poszczególnych projektach. Po inicjalizacji przechodzimy do pętli. W każdej kroku pętli wykonywane są następujące czynności:

- Obliczane są statystyki zwycięskiego projektu, które dodawane są do zwracanej listy.
- Aktualizowana jest tabela wydatków. Ponieważ w iteracji zapisany jest budżet każdego wyborcy przed i po wybraniu projektu, wystarczy policzyć różnicę między nimi i dla każdego głosującego dodać odpowiedni rekord z wybranym projektem, jeśli różnica jest większa od 0.
- Przechodzimy po liście wszystkich pozostałych projektów. Jeśli projekt został odrzucony w danej iteracji, obliczane są jego statystyki które następnie są dodawane do zwracanej listy.

Proces wyliczania statystyk dla projektu  $A$  korzysta z aktualnego budżetu wyborców oraz listy wydatków. Aby obliczyć budżet zwolenników projektu  $A$  w trakcie jego rozważania, obliczana jest suma aktualnego budżetu wszystkich zwolenników. Do obliczenia listy projektów wybranych przed projektem  $A$ , sumujemy wydatki każdego zwolennika projektu  $A$ , grupując wydatki po wszystkich wcześniej wybranych projektach. W ten sposób, dla każdego wcześniej wybranego projektu  $B$  otrzymamy ilość pieniędzy, którą zwolennicy rozważanego projektu

A przeznaczyli na realizację projektu *B*. Do wyliczeń dla zwycięskiego projektu aktualnym budżetem jest budżet przed opłaceniem projektu. Do wyliczeń dla projektu odrzuconego aktualnym budżetem jest budżet po opłaceniu zwycięskiego projektu.

Opiszemy teraz tę procedurę na przykładzie. Skorzystamy z danych detalicznych, które wyliczyliśmy w poprzednim przykładzie. Dla ułatwienia, do każdego z projektów w pierwszej tabeli danych detalicznych dodana została lista indeksów wyborców, którzy poparli dany projekt.

#### Przykład wyliczenia statystyk dotyczących budżetów zwolenników

| Numer iteracji | Wybrany projekt                     | Odrzucone projekty            |
|----------------|-------------------------------------|-------------------------------|
| 1              | Parking rowerowy (1,3,4,5,7,8,9,10) | Skatepark (1,3,4,6,8,9)       |
| 2              | Schronisko (1,2,5,6,7,9)            | Sprzęt strażacki (2,3,4,7,10) |
| 3              | Budki łęgowe (6,8,10)               | -                             |
| 4              | -                                   | Pomoc ubogim (1,2,6)          |

| Iteracja | Rodzaj listy         | Wyb 1   | Wyb 2 | Wyb 3   | Wyb 4   | Wyb 5   | Wyb 6 | Wyb 7   | Wyb 8   | Wyb 9   | Wyb 10  |
|----------|----------------------|---------|-------|---------|---------|---------|-------|---------|---------|---------|---------|
| 1        | Budżet przed wyborem | 100zł   | 100zł | 100zł   | 100zł   | 100zł   | 100zł | 100zł   | 100zł   | 100zł   | 100zł   |
| 1        | Budżet po wyborze    | 37.50zł | 100zł | 37.50zł | 37.50zł | 37.50zł | 100zł | 37.50zł | 37.50zł | 37.50zł | 100zł   |
| 2        | Budżet przed wyborem | 37.50zł | 100zł | 37.50zł | 37.50zł | 37.50zł | 100zł | 37.50zł | 37.50zł | 37.50zł | 100zł   |
| 2        | Budżet po wyborze    | 0zł     | 25zł  | 37.50zł | 37.50zł | 0zł     | 25zł  | 0zł     | 37.50zł | 0zł     | 37.50zł |
| 3        | Budżet przed wyborem | 0zł     | 25zł  | 37.50zł | 37.50zł | 0zł     | 25zł  | 0zł     | 37.50zł | 0zł     | 37.50zł |
| 3        | Budżet po wyborze    | 0zł     | 25zł  | 37.50zł | 37.50zł | 0zł     | 0zł   | 0zł     | 0zł     | 0zł     | 0zł     |
| 4        | Budżet przed wyborem | 0zł     | 25zł  | 37.50zł | 37.50zł | 0zł     | 0zł   | 0zł     | 0zł     | 0zł     | 0zł     |
| 4        | Budżet po wyborze    | 0zł     | 25zł  | 37.50zł | 37.50zł | 0zł     | 0zł   | 0zł     | 0zł     | 0zł     | 0zł     |

Tabela 3.9: Dane detaliczne zwrócone przez metodę równych udziałów. Obok projektów w nawiasach znajdują się indeksy wyborców popierających dany projekt.

W pierwszej iteracji zwyciężył projekt 'Parking rowerowy' i to dla niego w pierwszej kolejności wyliczamy dane statystyczne. Budżet jego zwolenników wyliczamy z listy budżetu wyborców przed wybraniem projektu zawartej w danych detalicznych 1 iteracji. Ponieważ jest to pierwszy projekt który rozważamy, jego zwolennicy nie mieli jeszcze okazji wydać pieniędzy na inne projekty. Po obliczeniu danych statystycznych dla tego projektu, do tabeli spisującej wydatki wyborców dopisujemy zwycięski projekt 'Parking rowerowy' oraz kwotę na niego przeznaczoną przez każdego z jego zwolenników. Wydatki możemy policzyć z różnicy budżetów wyborców przed i po realizacji projektu.

| Projekt          | Wyb 1   | Wyb 2 | Wyb 3   | Wyb 4   | Wyb 5   | Wyb 6 | Wyb 7   | Wyb 8   | Wyb 9   | Wyb 10  |
|------------------|---------|-------|---------|---------|---------|-------|---------|---------|---------|---------|
| Parking rowerowy | 62.50zł | -     | 62.50zł | 62.50zł | 62.50zł | -     | 62.50zł | 62.50zł | 62.50zł | 62.50zł |

Tabela 3.10: Tabela wydatków wyborców po wybraniu 'Parking rowerowy'.

W tej samej iteracji odrzucony został projekt 'Skatepark', dla którego wyliczymy teraz dane statystyczne. Budżet jego zwolenników wyliczymy z listy budżetów wyborców po wybraniu projektu, co w naszym przypadku będzie sprowadzało się do wyliczenia  $37.50\text{zł}$  (wyborcy 1, 3, 4, 8, 9) \*  $5 + 100\text{zł}$  (wyborca 6) =  $287.50\text{zł}$ . Do wyliczenia listy projektów wybranych przed rozważeniem projektu 'Skatepark', skorzystamy z tabeli wydatków, sumując dla każdego wpisanego tam projektu kwotę, jaką zwolennicy projektu 'Skatepark' przeznaczyci na te projekty. Tutaj zwolennicy (wyborcy 1, 3, 4, 8, 9) przeznaczyci łącznie  $312.50\text{zł}$ .

| Projekt   | Budżet zwolenników | Stracony budżet             |
|-----------|--------------------|-----------------------------|
| Siłownia  | 800zł              | -                           |
| Skatepark | 287.50zł           | Parking rowerowy - 312.50zł |

Tabela 3.11: Wynik działania metody po analizie pierwszej iteracji.

W kolejnej iteracji mamy podobną sytuację, co w iteracji pierwszej. Wybrany został projekt 'Schronisko'. Do wynikowej tabeli dodajemy więc ten projekt, ponownie sumując budżet jego zwolenników (wyborcy 1, 2, 5, 6, 7, 9) z listy budżetów wyborców przed wyborem projektu. Jego budżet utracony na rzecz innych projektów wyliczamy z niezmienionej od końca pierwszej iteracji tabeli wydatków, patrząc tylko na wydatki zwolenników projektu 'Schronisko'. Po wyliczeniu danych statystycznych, do tabeli wydatków dodajemy wydatki zwolenników projektu.

| Projekt          | Wyb 1   | Wyb 2 | Wyb 3   | Wyb 4   | Wyb 5   | Wyb 6 | Wyb 7   | Wyb 8   | Wyb 9   | Wyb 10  |
|------------------|---------|-------|---------|---------|---------|-------|---------|---------|---------|---------|
| Parking rowerowy | 62.50zł | -     | 62.50zł | 62.50zł | 62.50zł | -     | 62.50zł | 62.50zł | 62.50zł | 62.50zł |
| Schronisko       | 37.50zł | 75zł  | -       | -       | 37.50zł | 75zł  | 37.50zł | -       | 37.50zł | -       |

Tabela 3.12: Tabela wydatków wyborców po wybraniu 'Schronisko'.

Oprócz tego, w drugiej iteracji odrzucony został projekt 'Sprzęt strażacki', dla którego wyliczymy teraz dane statystyczne. Budżet jego zwolenników (2, 3, 4, 7, 10) wyliczymy z listy budżetów wyborców po wybraniu projektu. W ten sam sposób, co dla projektu 'Skatepark', wyliczymy też listę projektów wybranych przed rozważeniem projektu 'Sprzęt strażacki', korzystając z tabeli wydatków. W tym przypadku jednak zostaną wylistowane projekty wybrane zarówno w iteracji pierwszej, jak i drugiej.

| Projekt          | Budżet zwolenników | Stracony budżet                                 |
|------------------|--------------------|-------------------------------------------------|
| Siłownia         | 800zł              | -                                               |
| Skatepark        | 287.50zł           | Parking rowerowy - 312.50zł                     |
| Schronisko       | 350zł              | Parking rowerowy - 250zł                        |
| Sprzęt strażacki | 137.50zł           | Parking rowerowy - 250zł, Schronisko - 112.50zł |

Tabela 3.13: Wynik działania metody po analizie drugiej iteracji.

W trzeciej iteracji metoda wybrała jeden projekt i nie odrzuciła żadnego innego. W tej sytuacji funkcja dla wybranego projektu 'Budki łęgowe' wyliczy dane statystyczne i zapisze wydatki jego zwolenników w ten sam sposób co w poprzednich iteracjach.

| Projekt          | Wyb 1   | Wyb 2 | Wyb 3   | Wyb 4   | Wyb 5   | Wyb 6 | Wyb 7   | Wyb 8   | Wyb 9   | Wyb 10  |
|------------------|---------|-------|---------|---------|---------|-------|---------|---------|---------|---------|
| Parking rowerowy | 62.50zł | -     | 62.50zł | 62.50zł | 62.50zł | -     | 62.50zł | 62.50zł | 62.50zł | 62.50zł |
| Schronisko       | 37.50zł | 75zł  | -       | -       | 37.50zł | 75zł  | 37.50zł | -       | 37.50zł | -       |
| Budki łęgowe     | -       | -     | -       | -       | -       | 25zł  | -       | 37.50zł | -       | 37.50zł |

Tabela 3.14: Tabela wydatków wyborców po wybraniu 'Budki łęgowe'.

| Projekt          | Budżet zwolenników | Stracony budżet                                 |
|------------------|--------------------|-------------------------------------------------|
| Siłownia         | 800zł              | -                                               |
| Skatepark        | 287.50zł           | Parking rowerowy - 312.50zł                     |
| Schronisko       | 350zł              | Parking rowerowy - 250zł                        |
| Sprzęt strażacki | 137.50zł           | Parking rowerowy - 250zł, Schronisko - 112.50zł |
| Budki łęgowe     | 100zł              | Parking rowerowy - 125zł, Schronisko - 75zł     |

Tabela 3.15: Wynik działania metody po analizie trzeciej iteracji.

W ostatniej iteracji nie został wybrany żaden projekt, dlatego do tabeli wydatków nie zostanie dodany żaden wpis. Obliczymy za to dane statystyczne ostatniego projektu 'Pomoc ubogim', korzystając z wyliczonej dotychczas tabeli wydatków i listy budżetów wyborców po wybraniu projektu (w tym wypadku lista ta jest równa liście budżetów wyborców przed wybraniem projektu).

| Projekt          | Budżet zwolenników | Stracony budżet                                                        |
|------------------|--------------------|------------------------------------------------------------------------|
| Siłownia         | 800zł              | -                                                                      |
| Skatepark        | 287.50zł           | Parking rowerowy - 312.50zł                                            |
| Schronisko       | 350zł              | Parking rowerowy - 250zł                                               |
| Sprzęt strażacki | 137.50zł           | Parking rowerowy - 250zł, Schronisko - 112.50zł                        |
| Budki łęgowe     | 100zł              | Parking rowerowy - 125zł, Schronisko - 75zł                            |
| Pomoc ubogim     | 25zł               | Parking rowerowy - 62.50zł, Schronisko - 187.50zł, Budki łęgowe - 25zł |

Tabela 3.16: Wynik działania metody po analizie czwartej iteracji.

### Funkcja do wyliczenia efektywnego poparcia projektów

Ta funkcja, oprócz danych detalicznych, korzysta również z instancji wyborczej i profilu wyborczego. Wynikiem tej funkcji jest lista projektów z przypisanymi wartościami całkowito-liczbowymi, które reprezentują procentowy wskaźnik efektywnego poparcia. Warto zwrócić uwagę na to, że sztuczne zwiększanie lub zmniejszanie budżetów zwolenników dla każdego projektu w celu obliczenia ich efektywnego poparcia jest podejściem błędnym, gdyż zmodyfikowanie budżetu zwolenników drastycznie wpływa na przebieg metody równych udziałów, między innymi na kolejność w której rozważane są projekty. Aby poprawnie wyliczyć wartości efektywnego poparcia, zdecydowałem się skorzystać z pomocy metody równych udziałów. Dzięki niej nie musimy sztucznie zmieniać budżetu zwolenników danego projektu, a zamiast tego możemy w każdej iteracji obliczyć wskaźnik efektywnego poparcia korzystając z wcześniej wyliczonego współczynnika  $\rho$  i aktualnego budżetu wyborców.

Współczynnik  $\rho$  to specjalna wartość obliczana przez metodę równych udziałów dla każdego projektu w każdym kroku iteracyjnym. Współczynnik  $\rho$  projektu  $A$  mówi, jaką część inicjalnego budżetu musi przeznaczyć każdy ze zwolenników, aby pokryć koszt realizacji projektu. Jeśli wyborca nie posiada odpowiedniej części inicjalnego budżetu (bo wydał wcześniej pieniądze na pokrycie innych projektów), to na realizację projektu  $A$  przeznacza cały swój pozostały budżet. W takich sytuacjach współczynnik  $\rho$  jest z reguły większy, aby bardziej 'zamożni' wyborcy mogli pokryć większą część realizowanego projektu  $A$ . Przykładowo więc, jeśli koszt projektu  $A$  wynosi 200zł, a jego jedynych czterech zwolenników posiada w pewnej iteracji budżety w wysokości 30zł, 100zł, 90zł, 50zł (przy inicjalnym budżecie 100zł), to aby pokryć koszt projektu  $A$  współczynnik  $\rho$  musi wynieść 0.6 - Taka wartość spowoduje, że pierwszy i czwarty wyborca wydadzą pozostały swój budżet, natomiast drugi i trzeci wyborca podzielą się po równo kosztem 60zł, co da łącznie koszt projektu 200zł (30zł + 50zł + 60zł + 60zł = 200zł). Co ważne, współczynnik  $\rho$  jest dokładnie odwrotnością innego współczynnika - efektywnej liczby

głosów projektów. Na podstawie efektywnej liczby głosów w każdej iteracji wybierane są projekty które mają zostać zrealizowane. Wybierane są projekty z największą efektywną liczbą głosów, a to równorzędnie oznacza, że są to także projekty z najmniejszym współczynnikiem  $\rho$ .

Aby w pełni zrozumieć działanie metody, konieczne jest też wprowadzenie konceptu projektu pominiętego. Projekt pominięty, w kontekście metody równych udziałów, to projekt który znajduje się w jego instancji wyborczej, ale nie jest on rozważany przez metodę w żadnym momencie jego działania. Taki projekt nie może zatem zostać wybrany, a co za tym idzie nie wpływa on na zmiany w budżecie. Jest to szczególnie istotne w przypadku projektów, które oryginalnie zostały wybrane przez metodę równych udziałów.

Do wyliczenia efektywnego poparcia dowolnego projektu  $A$  potrzebne będzie połączenie tych dwóch idei, wartości współczynnika  $\rho$  oraz konceptu projektu pominiętego. Jeśli w danej instancji wyborczej oznaczymy projekt  $A$  jako pominięty, a następnie uruchomimy metodę równych udziałów na takiej instancji, będziemy mieli możliwość porównania pominiętego projektu (a dokładniej, jego kosztu i aktualnych budżetów jego zwolenników) z najmniejszym (czyli najlepszym) w danej iteracji współczynnikiem  $\rho_B$  innego projektu  $B$ . W każdej iteracji, po wybraniu jakiegoś projektu  $B$  innego od  $A$ , będziemy mogli policzyć pokrycie projektu  $A$ , czyli jak dużo pieniędzy byłoby w stanie zebrać zwolennicy projektu  $A$  przy współczynniku  $\rho_B$ .

$$cover = \sum_{v \in A.supporters} \min(v.availableBudget, \rho_B * v.initialBudget)$$

Ta wartość natomiast jest już bezpośrednio wykorzystywana do policzenia efektywnego poparcia projektu  $A$  - wystarczy bowiem sprawdzić, jaką część kosztu projektu  $A$  reprezentuje wyliczone pokrycie i sprowadzić wynik do wartości procentowej:

$$Iteration\ effective\ support_A = \frac{cover}{A.cost} * 100$$

Dla każdej iteracji metody równych udziałów przeliczamy efektywne poparcie projektu  $A$ . W każdej iteracji ta wartość będzie inna, gdyż zmianie ulega nie tylko budżet zwolenników projektu  $A$ , ale także współczynnik  $\rho_B$ , w każdej iteracji bowiem do realizacji wybierany jest inny projekt  $B$ . Dodatkowo, aby obsłużyć przypadek w którym projekt pominięty  $A$  jest również ostatnim z wybranych oryginalnie projektów, dodamy wyliczenie efektywnego poparcia na podstawie budżetu który pozostał zwolennikom projektu  $A$  w momencie zakończenia metody. Jeśli zwolennikom zostanie wystarczająco pieniędzy aby pokryć taki projekt, efektywne poparcie przekroczy 100%. Jako finalną wartość efektywnego poparcia wybieramy największą wyliczoną wartość na przestrzeni iteracji.

Zobaczmy, jak zachowuje się ten algorytm przy różnych rodzajach projektów. W przypadku projektu  $A_w$ , który w finalnej alokacji został wybrany do realizacji, spodziewamy się efektywnego poparcia co najmniej równego 100%. Prześledźmy iteracyjne wyliczanie efektywnego poparcia w momencie, w którym oznaczymy projekt  $A_w$  jako projekt pominięty i uruchomimy metodę równych udziałów. W pierwszej kolejności rozważane zostaną projekty które w oryginalnej alokacji zostały wybrane przed projektem  $A_w$ . Skoro zostały wcześniej rozważane, to ich współczynnik  $\rho$  był niższy od współczynnika  $\rho_{A_w}$ , a co za tym idzie wartość pokrycia projektu  $A_w$  nigdy nie przekroczy jego kosztu, (żeby mógł to zrobić, wymagane by było aby rozważane  $\rho > \rho_{A_w}$ ) co z kolei powoduje, że wartość iteracyjnego efektywnego poparcia nie przekroczy 100%. Ciekawie robi się w momencie osiągnięcia iteracji, w której projekt  $A_w$  oryginalnie został wybrany do realizacji. Ponieważ teraz jest on oznaczony jako pominięty, zamiast niego zostanie wybrany inny projekt  $B$ . Projekt ten będzie mieć współczynnik  $\rho_B$  nie

niższy od  $\rho_{A_w}$  (gdyby było inaczej, to w oryginalnym biegu to projekt  $B$  zostałby wybrany do realizacji nad projektem  $A$ ). W obliczeniach pokrycia  $A_w$  wyjdzie nam zatem co najmniej koszt projektu (A bardzo często nawet i więcej) co przełoży się na wartość iteracyjnego efektywnego poparcia przekraczającego 100%. W kolejnych iteracjach, w zależności od skomplikowanych powiązań pomiędzy grupami wyborców, wartość iteracyjnego efektywnego poparcia może być zarówno mniejsza jak i większa od 100%. W rzadkich przypadkach może nawet przekroczyć dotychczasowe obliczone maksimum, jeśli wartość  $\rho_B$  rozważanego projektu  $B$  będzie wysoka, a budżet zwolenników  $A_w$  nie uległ znacznemu pomniejszeniu. Tak jak zostało wspomniane wcześniej, na koniec działania metody podliczane jest pokrycie projektu i efektywne poparcie  $A_w$  na podstawie pozostałego budżetu zwolenników. Z reguły dla projektów nie wybieranych na końcu obliczenie to nic nie zmienia, ponieważ budżet zwolenników jest już za mały, aby pokryć koszt  $A_w$  (a więc i efektywne poparcie jest mniejsze od 100%). Obliczenie to ma znaczenie, gdy projekt  $A_w$  oryginalnie został wybrany jako ostatni.

Przypuśćmy zatem, że  $A_w$  faktycznie jest oryginalnie ostatnim wybranym projektem. U uruchamiamy metodę równych udziałów z  $A_w$  jako projekt pominięty. W każdej iteracji wybrane zostały projekty z niższym  $\rho$ , więc iteracyjne wartości efektywnego poparcia projektu  $A_w$  nie przekroczą 100%. Jeśli w ostatniej iteracji zamiast  $A_w$  wybrany zostanie inny projekt, to tłumaczenie z poprzedniego akapitu dalej jest w mocy. Jeśli natomiast żaden inny projekt nie został wybrany do realizacji (bo ich koszty przekraczałyby dostępny budżet ich zwolenników), wtedy obliczenie prawdziwego efektywnego poparcia przypada na wyliczenia, które odbywają się na końcu działania metody. Ponieważ wiemy, że w oryginalnej ostatniej iteracji projekt  $A_w$  został wybrany, to wiemy również, że na koniec działania metody z projektem pominiętym  $A_w$  zwolennicy posiadają wystarczającą ilość pieniędzy aby pokryć koszt projektu  $A_w$ . To przekłada się na iteracyjne efektywne poparcie przekraczające 100%, które następnie jest wybierane jako maksymalne znalezione efektywne poparcie.

Rozważenie projektu  $A_l$ , który w oryginalnej alokacji nie został wybrany do realizacji, jest koncepcyjnie prostsze. Ponieważ projekt ten nie został zrealizowany, ustawienie go jako projekt pominięty nie zmieni sposobu, w jaki metoda równych udziałów wybiera projekty do rozważenia. Uruchomienie metody równych udziałów z pominiętym  $A_l$  to tak naprawdę sprawdzenie, w której iteracji współczynnik  $\rho_{A_l}$  i budżet zwolenników  $A_l$  były najbliżej wybrania do realizacji. Ponieważ projekt  $A_l$  oryginalnie nie został wybrany, wiemy, że w każdej takiej iteracji współczynnik  $\rho_{A_l}$  był większy od współczynnika  $\rho$  wybranego w tej iteracji projektu (co powoduje, że współczynnik wybranego projektu  $\rho$  nie wystarcza na pokrycie projektu  $A_l$ ) lub budżet zwolenników był nie wystarczający na pokrycie kosztu projektu  $A_l$  (taka sytuacja ma miejsce w iteracjach, w których oryginalnie  $A_l$ ) został już oznaczony jako projektu odrzucony.

Wracając do implementacji w Pabutools, do metody równych udziałów został zatem dodany argument pozwalający określić, który z projektów dostępnych w instancji ma być projektem pominiętym. Projekt pominięty nie będzie brać udziału w procesie wybierania zwycięskiej alokacji, ale jego koszt i aktualny budżet jego zwolenników, wraz ze współczynnikiem  $\rho$  w danej iteracji, posłużą do obliczenia wartości efektywnego poparcia w tej iteracji metody. Finalną wartością efektywnego poparcia będzie największe efektywne poparcie na przestrzeni wszystkich wykonanych iteracji. Oprócz tego, w detalach zwracanych przez tę metodę został dodany atrybut, w którym znajdziemy wyliczoną wartość efektywnego poparcia dla pominiętego projektu.

Funkcja do wyliczenia efektywnego poparcia każdego z projektów przechodzi po wszystkich projektach dostępnych w instancji wyborczej. W kroku pętli uruchamiana jest metoda równych udziałów z ustawioną flagą na wyliczenie danych statystycznych i aktualnym projektem jako projektem pominiętym. Finalny wynik jest wyciągany z detali zwróconych przez metodę i przypisywany do projektu pominiętego. Tak spreparowana para jest zapisana do zwracanej mapy projektów i ich efektywnych poparców.

### 3.3. Wizualizacja danych statystycznych

Celem stworzenia tego narzędzia było umożliwienie administratorom (bądź też zwykłym użytkownikom) łatwą generację i wizualizację wyników oraz danych statystycznych dowolnych wyborów budżetu partycypacyjnego. Dzięki niemu użytkownik nie musi mieć styczności z surowymi danymi, które mogłyby utrudnić interpretację ich znaczenia. Tak jak zostało wspomniane w rozdziale opisującym architekturę rozwiązania, narzędzie to jest aplikacją internetową podzieloną na serwery front-endu i back-endu, których działanie i cechy szczególne opiszemy w następnych podrozdziałach.

#### 3.3.1. Serwer front-endu

Serwer front-endu odpowiedzialny jest za wystawienie aplikacji, będącej interfejsem użytkownika. W interfejsie tym zbierane są dane oraz parametry potrzebne do wyliczenia interesujących użytkownika danych statystycznych. Odbywa się to za pomocą prostego formularza, który daje możliwość przesłania pliku w formacie Pabulib lub wybrania jednego z predefiniowanych wyborów. Dostępne parametry pozwalają użytkownikowi zdecydować, czy chce skorzystać z wyczerpującej wersji metody równych udziałów (ADD-1), ustawić warunek stopu metody wyczerpującej oraz czy algorytm ma przeliczyć efektywne poparcie projektów. Po zaakceptowaniu formularza dane te są przesyłane za pomocą protokołu HTTP do odpowiednich punktów końcowych serwera back-endu, który zwraca do interfejsu dane statystyczne w formacie JSON. Otrzymane dane są przez serwer front-endu pokazywane użytkownikowi w formie tabeli. Aby umożliwić szersze i łatwiejsze udostępnianie wyników zwykłym wyborcom, serwer front-endu oferuje możliwość eksportu danych do statycznego pliku typu HTML+CSS. Taki plik może być wystawiony na oficjalnej stronie wyborów lub rozesłany bezpośrednio do zainteresowanych wyborców.

### 3.3.2. Serwer back-endu

Serwer back-endu jest w pełni niezależną aplikacją będącą serwerem REST API. Jego jedynym zadaniem jest odbieranie zapytań od serwera front-endu, odpowiednie przygotowanie parametrów wywołania metody równych udziałów i funkcji wyliczenia danych statystycznych, wywołanie wyżej wspomnianej metody i funkcji oraz zwrócenie wyników w formacie JSON. Aby obsłużyć dane z predefiniowanych wyborów jak i wyborów przesłanych za pomocą plików w formacie .pb, serwer back-endu udostępnia dwa punkty końcowe. Oba punkty końcowe przyjmują te same argumenty obliczeń i robią to korzystając z opcjonalnych parametrów dostępnych w adresie URL.

Wśród akceptowanych parametrów opcjonalnych mamy więc:

- `exhaust` - jeśli parametr jest obecny, skorzystaj z wyczerpującego wariantu metody.
- `feasible-stop` - jeśli parametr jest obecny wraz z parametrem `exhaust`, skorzystaj z wyczerpującego wariantu metody który zatrzymuje się dopiero przy znalezieniu niespełnialnej alokacji.
- `eff-support` - jeśli parametr jest obecny, oblicz efektywne poparcie projektów.

Punkty końcowe różni natomiast sposób w jaki przesyłane są dane o wyborach. W przypadku plików predefiniowanych, punkt końcowy jest typu GET. Nazwa predefiniowanej elekcji zapisana jest w ścieżce URL. Identyfikuje ona plik .pb zawierający informacje o danej elekcji. Plik ten zapisany jest w wewnętrznych zasobach serwera back-endu. Przykładowy adres URL zapytania do tego punktu końcowego może więc wyglądać tak:

**`localhost:8000/mes/wieliczka2023?exhaust&eff-support`**

To zapytanie obliczy dane statystyczne dla predefiniowanej elekcji z Wieliczki 2023, korzystając z wariantu wyczerpującego metody, a w dodatku policzy efektywne poparcie projektów.

Drugi punkt końcowy, pozwalający na przesłanie własnego pliku .pb, jest typu POST. Umożliwia on przesłanie danych zawartych w pliku korzystając z ciała zapytania. Przykładowy adres URL zapytania do tego punktu końcowego może wyglądać tak:

**`localhost:8000/mes/upload?exhaust&feasible-stop`**

W tym wypadku nazwa adresu nie zawiera nazwy wyborów, ponieważ wszystkie potrzebne dane zawarte są w ciele metody. Takie zapytanie sparsuje przesłany plik i w przypadku braku błędów składniowych policzy jego dane statystyczne, wywołując wyczerpujący wariant metody, który zatrzyma się dopiero po natrafieniu na niespełnialną alokację. W przypadku błędu parsowania użytkownik zostanie poinformowany o przyczynie błędu za pomocą powiadomienia.

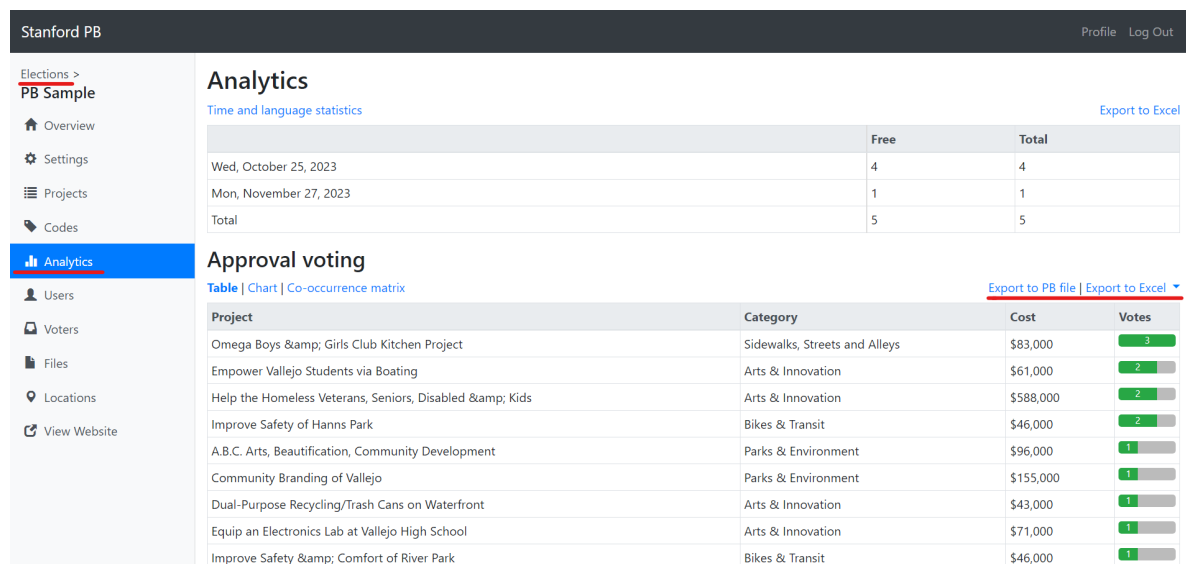
## Rozdział 4

# Obsługa i instalacja

Ten rozdział omówi w jaki sposób korzystać z zaimplementowanych rozwiązań i narzędzi.

### 4.1. Stanford Participatory Budgeting Platform

Do rezultatów wyborów na tej platformie dostęp mają jedynie administratorzy, którzy są również odpowiedzialni za tworzenie, edycję i zarządzanie wyborami. Po utworzeniu i przeprowadzeniu odpowiednich wyborów administrator ma możliwość eksportu wyników do pliku .csv lub pliku .pb. Aby to zrobić, należy zalogować się do panelu administratora, wejść w zarządzanie dowolnych wyborów, a następnie przejść do zakładki "Analytics". To w niej znajdziemy oba przyciski służące do eksportu wyników interesujących nas wyborów. Pobrany plik .pb nie wymaga dodatkowych edycji i jest już gotowy do wykorzystania w zewnętrznych narzędziach, takich jak Pabutools lub internetowy kalkulator wyników dla wyborów korzystających z metody równych udziałów. Na poniższym obrazku czerwoną linią podkreślone są omówione kroki.



Rysunek 4.1: Graficzne przedstawienie omówionej funkcjonalności w Stanford Participatory Budgeting Platform.

Z przedstawionego powyżej rozwiązania można korzystać w [oficjalnej wersji narzędzia](#) utrzymywanego przez zespół z uniwersytetu Stanforda lub lokalnego, stabilnego [odgałęzienia tego repozytorium](#). Instalacja narzędzia jest dokładnie opisana w pliku README.md projektu.

## 4.2. Pabutools

Tak jak zostało wspomniane w poprzednim rozdziale, rozwinięcia, które zostały zaimplementowane w bibliotece Pabutools, można podzielić na dwie oddzielne części, które użyte razem pozwalają na zwrócenie danych statystycznych o wynikach metody równych udziałów. Obsługę omówimy zatem z poszanowaniem tego naturalnego podziału.

### 4.2.1. Ekstrakcja danych o przebiegu metody równych udziałów

Pierwszym krokiem jest umożliwienie zbierania danych o przebiegu metody równych udziałów. Robi się to z użyciem flagi "analytics", która domyślnie jest ustawiona na False. Przekazując wartość True w argumencie umożliwiamy zbieranie dodatkowych danych, które pozwalają na szybsze odtwarzanie kolejnych iteracji metody równych udziałów. Ustawienie tej flagi w metodach wyczerpujących również przyniesie zamierzony efekt, zwracając dodatkowe dane o ostatnim udanym biegu metody. Obecnie zbieranie danych o przebiegu metody równych udziałów jest możliwe tylko w rozwiązaniach rezolutnych, które w przypadku remisów zwracają jeden wynik.

Aby umożliwić przechowywanie danych o przebiegu metody, wartość zwracana wszystkich reguł została zmieniona ze standardowej listy na specjalną klasę reprezentującą alokację budżetu o nazwie "BudgetAllocation", która dziedziczy po klasie listy. Umożliwia to na dostęp do wyników metody w taki sam sposób jak dotychczas. Klasa ta została rozszerzona o dodatkowy atrybut "details", w którym zlokalizowane są zebrane dane o przebiegu metody, o ile flaga "analytics" została włączona podczas uruchomienia metody. To właśnie te dane będą kluczowe w drugim etapie odzyskiwania danych statystycznych metody równych udziałów. Struktura obiektu detali nie jest aż tak istotna z punktu widzenia użytkownika. W przypadku metody równych udziałów składa się ona z listy mnogości głosujących wykorzystanej w profilu oraz listy iteracji reprezentowanej klasą "MESIteration". Klasa ta przechowuje informacje o danej iteracji z metody równych udziałów, w tym informacje o wybranym projekcie i projektach odrzuconych, a także liście dostępnych budżetów wyborców przed i po wyborze zwycięskiego projektu.

Omówione zmiany znajdują się w module "pabutools.rules.mes", odpowiednio w klasach mes\_rule.py i mes\_details.py.

Więcej informacji o ekstrakcji danych z przebiegu metody równych udziałów, w tym sygnaturach funkcji i klas można znaleźć w oficjalnej dokumentacji Pabutools na [stronie poświęconej regułom wyborczym](#), w sekcji Method of Equal Shares.

### 4.2.2. Wyliczenie danych statystycznych dla wyników wyborów

Na podstawie danych detalicznych z klasy reprezentującej alokację budżetu możemy wyliczyć interesujące nas dane statystyczne. Służą do tego nowo utworzone metody do obliczenia konkretnych statystyk, dotyczących zarówno budżetu zwolenników, jak i wyliczenia efektywnego poparcia dla wszystkich projektów z danej instancji wyborczej.

## Obliczenie statystyk dotyczących budżetu zwolenników projektów

Do obliczenia statystyk wokół budżetu zwolenników projektów należy użyć funkcji `"calculate_project_loss"` z modułu `"pabutools.analytics.mesanalytics"`. Jedyńm wymagany argumentem dla tej funkcji jest klasa `"AllocationDetails"`, która powinna zawierać dane o kolejnych iteracjach z analizowanego przez nas biegu metody równych udziałów dla konkretnej instancji. Funkcja ta zadziała tylko w przypadku detali mających atrybut `"iterations"`, która obecnie występuje tylko w detalach metody równych udziałów. Ze względu na iteracyjną naturę metody Phragmena jest też możliwość rozszerzenia jej o zbieranie danych z jej przebiegu i wykorzystanie ich w opisywanej funkcji. Oprócz obowiązkowego argumentu, funkcja `"calculate_project_loss"` umożliwia włączenie flagi `"verbose"` odpowiedzialnej za zwracanie na standardowe wyjście informacji pomagające w zrozumieniu działania tej funkcji.

Wynikiem tej funkcji jest lista obiektów typu `"ProjectLoss"`. Klasa ta dziedziczy po klasie `"Project"` i zawiera atrybuty opisujące dane statystyczne o wybranym projekcie. Oprócz atrybutu `"project"`, który zawiera wszelkie metainformacje o danym projekcie, klasa `"ProjectLoss"` posiada również atrybut `"supporters_budget"`, przechowujący inicjalną ilość pieniędzy, która przysługiwała zwolennikom tego projektu przed rozpoczęciem biegu metody równych udziałów, jak i atrybut `"budget_lost"`, który przechowuje mapę wszystkich projektów wybranych przed rozważeniem danego projektu oraz ile pieniędzy zwolennicy rozważanego projektu wydali na wcześniej wybrane projekty. W mapie tej kluczami są projekty, a wartościami - liczby dziesiętne reprezentujące stracony budżet zwolenników na rzecz projektu pod kluczem. Klasa `"ProjectLoss"` zawiera również metodę `"total_budget_lost"`, która sumuje wszystkie wartości w wyżej opisanej mapie.

### 4.2.3. Wyliczenie efektywnego poparcia projektów

Aby obliczyć efektywne poparcie dla wszystkich projektów, należy skorzystać z funkcji `"calculate_effective_supports"` z modułu `"pabutools.analytics.mesanalytics"`. Metoda ta wymaga trzech argumentów: instancji wyborczej, profilu wyborczego oraz wybranej alokacji budżetu z dostępnymi detalami przebiegu metody. Aby wyniki były spójne, przesyłana instancja i profil wyborczy powinny być takie same, co te użyte do wyliczenia przesłanej alokacji budżetu. Opcjonalnie możemy dodać jeszcze dwa argumenty: parametry wywołania metody równych udziałów `"mes_params"` (w formie mapy) oraz finalny budżet wyborów `"final_budget"`, który pozwala na symulację wyczerpującego wariantu metody równych udziałów. Wszystkie dodatkowe parametry metody zostaną zachowane, z wyłączeniem instancji, profilu, flagi `"analytics"`, oraz parametru `"skipped_project"`, które są niezbędne do prawidłowych wyliczeń. Odnośnie do argumentu `"final_budget"`, aby poprawnie z niego skorzystać, należy przekazać wartość finalnego budżetu użytego w ostatnim udanym przebiegu metody równych udziałów w wariantcie wyczerpującym. Wartość tę można wyliczyć korzystając z metody `"get_final_budget"` dostępnej w detalach alokacji. Funkcja do wyliczania efektywnego poparcia zwraca mapę z projektami jako klucze i procentową wartością efektywnego poparcia jako wartości.

Sama funkcja iteruje po projektach i korzysta z funkcji `"calculate_effective_support"`. Funkcja ta jest również upubliczniona i można ją wykorzystywać do szybszego wyliczania efektywnego poparcia dla pojedynczego projektu. Działa ona w ten sam sposób, ale zamiast argumentu alokacji budżetu wymaga przekazanie odpowiedniego projektu oraz informacji, czy podany projekt zwyciężył (czyli, czy był obecny w alokacji budżetu).

Więcej informacji o tych funkcjach można znaleźć w oficjalnej dokumentacji Pabutools na [stronie poświęconej analityce](#), w sekcji Properties Specific to the Method of Equal Shares.

#### 4.2.4. Inne zmiany w Pabutools

Do Pabutools została dodana możliwość kontrolowania warunku stopu dla metody wyczerpującej. Do tej funkcji został dodany opcjonalny argument "exhaustive\_stop", który w domyśle jest ustawiony na True. Oznacza to, że metoda wyczerpująca zakończy działanie, jeśli po raz pierwszy napotka na niespełnialny wynik (zwracając poprzednio wyliczoną alokację) lub po raz pierwszy natrafi na wynik wyczerpujący budżet przydzielony na wybór projektów (zwracając pierwszą wyczerpującą alokację). Ustawiając "exhaustive\_stop" na False, pozbywamy się drugiego warunku i pozwalamy metodzie na wykonywanie kolejnych iteracji, aż do momentu napotkania niespełnialnej alokacji. Dodanie tej opcji pozwala na dokładne odzwierciedlenie sposobu w jaki zostały wybrane projekty podczas wyborów budżetu obywatelskiego w Wieliczce i Świeciu w 2023 roku.

Więcej informacji o tej zmianie można znaleźć w oficjalnej dokumentacji Pabutools, [na stronie poświęconej regułom wyborczym](#), w sekcji "Exhaustion methods".

### 4.3. Wizualizacja danych statystycznych metody równych udziałów

W tym podrozdziale omówimy jak korzystać i w jaki sposób lokalnie uruchomić narzędzie do wizualizacji danych statystycznych.

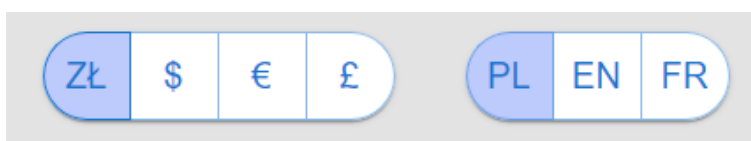
#### 4.3.1. Obsługa

Narzędzie można podzielić na 3 odrębne sekcje:

- Sekcję opcji, w której użytkownik może zmienić język i walutę prezentowaną na stronie.
- Sekcję formularzową, w której użytkownik podaje dane konieczne do obliczenia interesujących go danych statystycznych.
- Sekcję wynikową, gdzie wyświetlają się wyniki obliczeń w formie podsumowania i tabeli. Sekcja wynikowa pojawia się dopiero po złożeniu formularza.

#### Sekcja opcji

W sekcji opcji dostępne są dwa zbiory przycisków, jeden zawierający symbole reprezentujące walutę która dodawana jest jako jednostka do każdej wartości monetarnej, a drugi ze skrótami nazw państw według standardu ISO 3166-1, które reprezentują język wyświetlany na stronie. Użytkownik może dynamicznie zmieniać prezentowany język i walutę bez konieczności przeładowywania strony.



Rysunek 4.2: Przyciski reprezentujące kolejno waluty i języki dostępne na stronie.

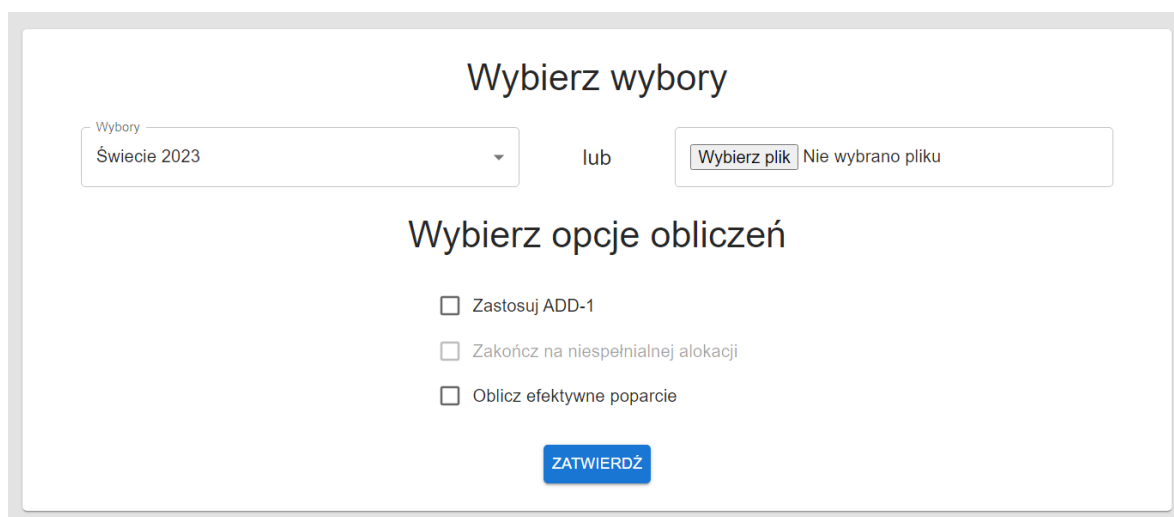
## Sekcja formularzowa

W sekcji formularzowej użytkownik podaje wszelkie informacje konieczne do poprawnego wyliczenia interesujących go danych statystycznych. W pierwszym kroku konieczne jest podanie wyborów na których powinny zostać wykonane obliczenia. Użytkownik ma tutaj dwie wykluczające się nawzajem opcje:

- Użytkownik może wybrać jedną z predefiniowanych elekcji. Na ten moment obsługiwane są elekcje ze Świecia 2023 i Wieliczki 2023.
- Użytkownik może również podać plik w formacie Pabulib.

W obu przypadkach dane statystyczne zostaną wyliczone przy użyciu biblioteki Pabutools. W następnym kroku użytkownik ma możliwość ustawienia opcji przebiegu wyliczeń. Wśród dostępnych opcji mamy:

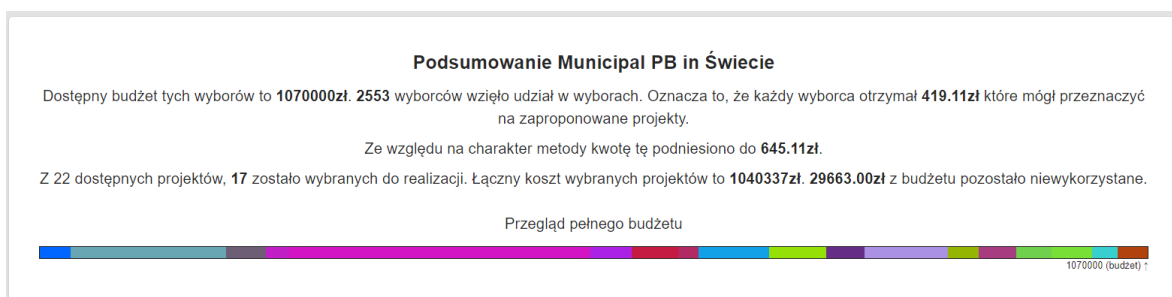
- Wyliczenie wyborów z użyciem wyczerpującej wersji metody równych udziałów ADD-1.
- Obliczanie metody aż do momentu uzyskania niespełnialnej alokacji, bez zatrzymywania się w momencie osiągnięcia alokacji wyczerpującej budżet. Ta opcja dostępna jest tylko przy włączonej opcji ADD-1.
- Dodatkowe policzenie efektywnego poparcia dla wszystkich projektów.



Rysunek 4.3: Sekcja formularzowa narzędzia.

## Sekcja wynikowa

Sekcja wynikowa podsumowuje i prezentuje wszystkie wyliczenia jakie miały miejsce po złożeniu formularza. Pierwsza jej część podsumowuje wszystkie informacje o wynikach wyborów. Możemy znaleźć tam informację o liczbie wybranych projektów i liczbie wszystkich projektów, budżecie wyborów, ilości budżetu przeznaczanego na realizację projektów oraz ilości niewykorzystanego budżetu. W przypadku wybrania wariantu wyczerpującego pokazuje się również informacja o ile budżet został zwiększony w ramach finalnej alokacji. W podsumowaniu znajduje się również przekrój obrazujący wydatki całego budżetu danych wyborów. Po najechaniu na odpowiedni pasek myszką ukaże nam się dymek z nazwą projektu oraz jego koszt.



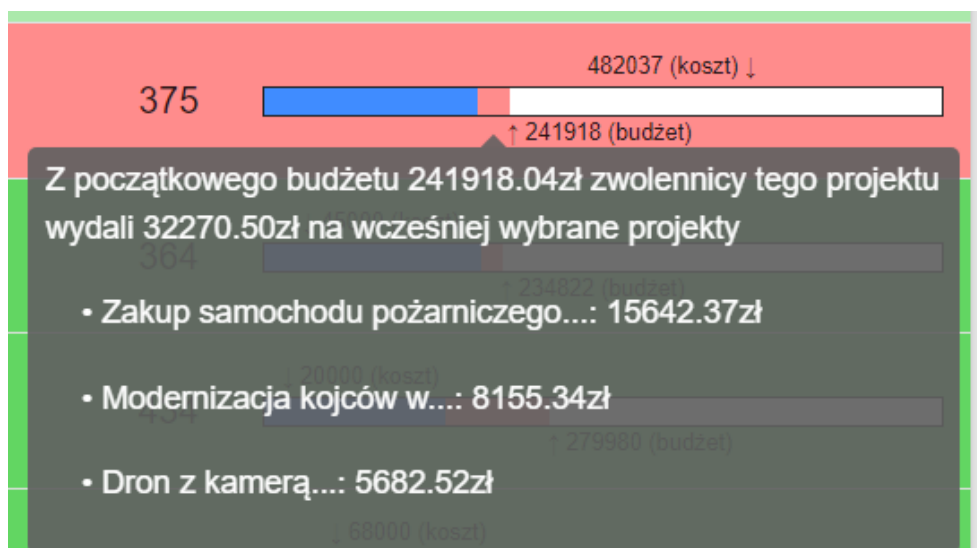
Rysunek 4.4: Przykładowa sekcja podsumowująca wyniki wyborów.

Druga część sekcji wynikowej to tabela z informacjami o każdym projekcie z instancji wyborczej. Każdy wiersz tabeli reprezentuje jeden projekt. W tabeli możemy znaleźć takie informacje jak: nazwa projektu, runda w której został rozważany, koszt projektu, liczba zdobytych głosów, efektywne poparcie (jeśli zostało wybrane do wyliczenia w formularzu) oraz wykres podsumowujący budżet przypadający zwolennikom tego projektu. Tabela daje możliwość sortowania wartości po kolumnach.

| Runda ↑ | Nazwa                                                                                                         | Koszt    | Liczba głosów | Efektywne poparcie | Przegląd budżetu                                                      |
|---------|---------------------------------------------------------------------------------------------------------------|----------|---------------|--------------------|-----------------------------------------------------------------------|
| 1       | <a href="#">Modernizacja kojców w schronisku dla zwierząt w celu poprawy warunków psów tam przebywających</a> | 150000zł | 1030          | 300%               | <div> <div>↓ 150000 (koszt)</div> <div>664468 (budżet) ↑</div> </div> |
| 2       | <a href="#">Naprawa i modernizacja pomostu nad Jeziorem Deczno</a>                                            | 413000zł | 623           | 86%                | <div> <div>413000 (koszt) ↓</div> <div>401907 (budżet) ↑</div> </div> |
| 3       | <a href="#">Sterylizacja suk z terenów wiejskich</a>                                                          | 39000zł  | 610           | 599%               | <div> <div>↓ 39000 (koszt)</div> <div>393520 (budżet) ↑</div> </div>  |
| 4       | <a href="#">Zakup sprzętu szkoleniowego dla OSP w Świeciu- Przechowo</a>                                      | 21900zł  | 531           | 422%               | <div> <div>↓ 21900 (koszt)</div> <div>342656 (budżet) ↑</div> </div>  |

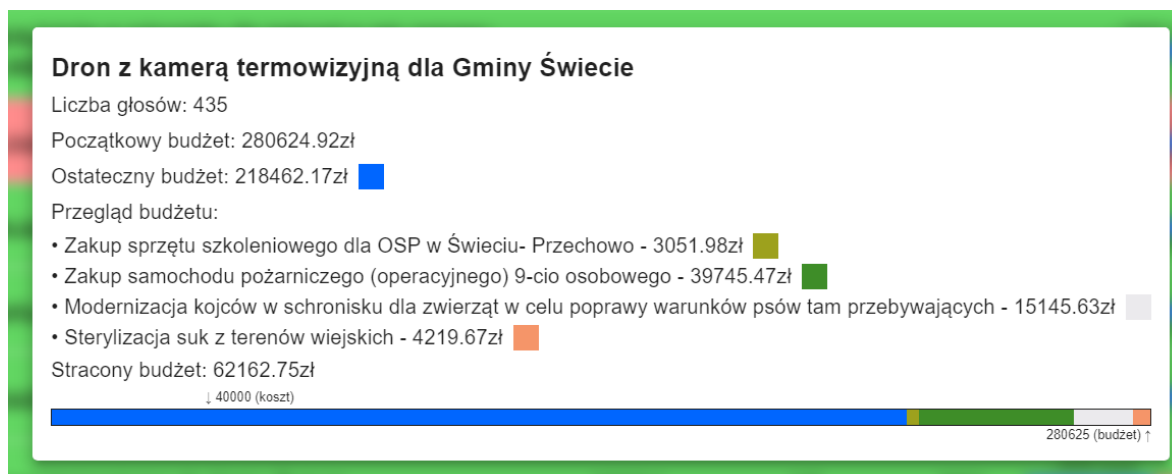
Rysunek 4.5: Fragment tabeli wynikowej z danymi o projektach.

Na wspomnianym wyżej wykresie znajduje się informacja o koszcie projektu (oznaczona strzałką nad wykresem) i o początkowym budżecie zwolenników danego projektu (oznaczona strzałką pod wykresem). Po najechaniu na niebieski pas na wykresie w dymku pojawi się informacja o finalnym budżecie i koszcie projektu. Po najechaniu na czerwony pas na wykresie w dymku pojawi się informacja o 3 projektach na które zwolennicy danego projektu wydali najwięcej pieniędzy przed rozważeniem danego projektu.



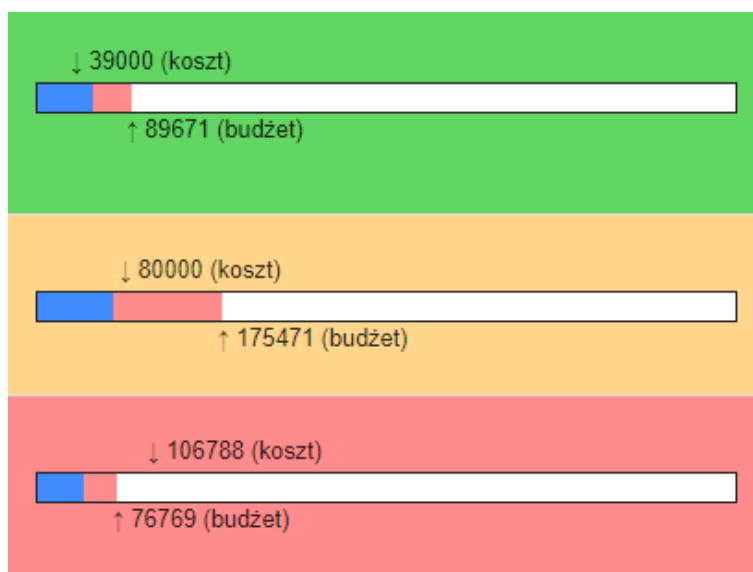
Rysunek 4.6: Przykładowe informacje pojawiające się po najechaniu na czerwony pas projektu.

Po kliknięciu w nazwę interesującego nas projektu, w oknie wyświetli się podgląd dokładniejszej wersji tego wykresu. Dokładniejsza wersja zawiera listę wszystkich projektów wybranych przed rozważeniem klikniętego projektu wraz z kwotą, która została na nie przeznaczona przez zwolenników tegoż projektu. Dodatkowo, każdy wcześniej wybrany projekt ma swój dedykowany pas na wykresie, zamiast jednolitego czerwonego pasa reprezentującego cały stracony budżet danego projektu.



Rysunek 4.7: Dokładne informacje o przykładowym projekcie z rozkładem straconego budżetu.

Każdy wiersz tabeli reprezentujący jeden projekt ma przypisany jeden z trzech kolorów. Kolor zielony reprezentuje projekty, które zostały wybrane do realizacji, a zatem te, których budżet zwolenników był wystarczający w czasie rozważania tego projektu na pokrycie jego kosztu (inaczej, ostateczny budżet zwolenników  $>$  koszt projektu). Kolorem czerwonym zostały oznaczone projekty, których koszt był większy niż początkowa wartość budżetu jego zwolenników. Są to projekty, które nie miały szans na zostanie wybranym już od samego początku działania metody. Kolorem żółtym zostały oznaczone projekty, których początkowy budżet zwolenników był wystarczający, aby pokryć jego koszt, ale przez wcześniejsze wybory innych projektów, budżet jego zwolenników spadł poniżej kosztu wymaganego do jego realizacji.



Rysunek 4.8: Kolory rzędów bazujące na przykładowych budżetach zwolenników projektów.

## Eksport wyników do statycznego pliku

Aby umożliwić łatwe udostępnianie wyników dla osób zaangażowanych w budżet obywatelski, do narzędzia został dodatkowo wprowadzony przycisk umożliwiający eksport wyników do statycznego pliku HTML wraz ze stylami CSS. Przycisk ten pojawia się pod formularzem, zaraz po jego zatwierdzeniu. Z uwagi na naturę statycznych plików, w wyeksportowanej wersji nie ma możliwości wykonywania dynamicznych akcji, takich jak sortowanie tabeli.

## Moduł translacyjny i18n

Przedstawione narzędzie jest zintegrowane z modułem translacyjnym, który pozwala na łatwą zmianę języka wyświetlanego tekstu. Tłumaczenia przechowywane są w słownikach na oddzielnym serwerze i18n. Aby aplikacja mogła skorzystać z tych słowników, należy podać dedykowany klucz dostępu, nadpisując zmienną "apiKey" w pliku "i18n.ts". Do aplikacji można dodawać kolejne słowniki z nowymi językami. Aby to zrobić należy wejść do panelu administracyjnego zarządzania słownikami dostępnymi na stronie [i18nexus](#). Po dodaniu nowego słownika i przetłumaczeniu wymaganych fraz, do listy obsługiwanych języków pod zmienną "supportedLngs" w pliku "i18n.ts" można dodać akronim nowo dodanego języka.

Dostęp do dedykowanego klucza, jak i panelu administracyjnego został przekazany osobom utrzymującym aplikację.

### 4.3.2. Instalacja

Do lokalnego postawienia serwera back-endu i front-endu zostało wykorzystane narzędzie Docker. Dzięki temu możliwe jest zbudowanie obrazów i uruchomienie serwera w specjalnych kontenerach, co zapewnia niezawodność narzędzia, niezależnie od systemu, na którym jest uruchamiany.

Aby uruchomić narzędzie, należy najpierw pobrać i skonfigurować narzędzie Docker. Po poprawnym skonfigurowaniu wystarczy wejść do folderu głównego projektu i w konsoli komend wpisać komendę:

```
docker-compose build --no-cache
```

To zbuduje nam potrzebne obrazy front-endu i back-endu. Aby uruchomić kontenery z aplikacją, należy wpisać komendę:

```
docker-compose up
```

Po wpisaniu należy poczekać, aż uruchomią się wszystkie konieczne serwisy. Aplikacja internetowa znajduje się pod adresem localhost:3000. Serwer back-endu jest aplikacją typu REST API i można go odpytywać pod adresem localhost:8000.

Możliwe, choć niezalecane, jest też uruchomienie aplikacji bez użycia narzędzia Docker. Należy wtedy osobno uruchomić serwery back-endu i front-endu. Serwer back-end wymaga Pythona w wersji 3. W folderze głównym back-endu (o nazwie mes\_analytics) należy zainstalować konieczne zależności komendą:

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

a następnie uruchomić serwer komendą:

```
python3 manage.py runserver
```

Serwer front-endu można uruchomić mając zainstalowany menedżer paczek npm. Po instalacji należy pobrać wykorzystywane paczki i uruchomić serwer komendą

```
npm install && npm start
```

Tak jak w poprzednim przypadku, strona internetowa będzie dostępna pod adresem localhost:3000, a serwer back-endu pod adresem localhost:8000.



## Rozdział 5

# Lokalizacja kodu źródłowego

### 5.1. Stanford Participatory Budgeting Platform

Kod został wprowadzony do oficjalnego repozytorium platformy. Wprowadzone zmiany można prześledzić w [tej paczce zmian](#).

### 5.2. Pabutools

Biblioteka została w pełni rozszerzona o wszystkie wymienione wyżej funkcjonalności. Zmiany te dostępne są od wersji 1.1.7 biblioteki. Zmiany można prześledzić w tym zestawie zmian:

- [Implementacja alokacji budżetu z detalami i wyliczanie analizy budżetu zwolenników](#)
- [Dodanie wsparcia dla multi-profiles](#)
- [Dostosowanie zmian do nowej architektury](#)
- [Implementacja funkcji do wyliczania efektywnego poparcia](#)
- [Dodanie opcji kontrolowania warunku stopu metody wyczerpującej](#)

### 5.3. Wizualizacja danych statystycznych metody równych udziałów

Kod został upubliczniony na moim repozytorium i przekazany pod opiekę promotora. Repozytorium z kodem jest dostępne [tutaj](#).



## Rozdział 6

# Podsumowanie

Celem pracy było stworzenie nowych i rozwój istniejących narzędzi tak, aby usprawnić proces wyłaniania zwycięzców z użyciem bardziej zaawansowanych metod wyborczych oraz obliczania danych statystycznych wynikających z przebiegu jednej z tych metod - metody równych udziałów. Cel ten udało się zrealizować, wprowadzając do świata reguł wyborczych budżetu partycypacyjnego nowe rozwiązania, wśród nich:

- Dodanie możliwości eksportu do formatu Pabulib instancji i profili wyborczych utworzonych w platformie do głosowania "Stanford Participatory Budgeting Platform".
- Implementacja funkcjonalności umożliwiających obliczenie zaawansowanych danych statystycznych obrazujących sposób działania metody równych udziałów.
- Stworzenie narzędzia pozwalającego na generację i wizualizację wyliczonych danych statystycznych.

Wszystkie wyżej przedstawione punkty były tworzone z myślą centralizowania rozwiązań, klarownej obsługi przez użytkowników końcowych i umożliwienia łatwego rozwijania zaimplementowanych kawałków kodu. Rozwiązania te zostały zaimplementowane z użyciem języków Ruby, Python i TypeScript oraz zrębów Django i React.

Dzięki realizacji tej pracy udało się ułatwić życie administratorów zarządzających elektronicznymi głosowaniami na budżet partycypacyjny z wykorzystaniem platformy Stanford Participatory Budgeting Platform. Umożliwiając eksport danych o głosowaniu do zunifikowanego pliku w formacie Pabulib, pozwalamy im na wykorzystanie innych narzędzi skoncentrowanych wokół budżetu partycypacyjnego. Takim narzędziem jest między innymi Pabutools, w którym od teraz dostępne są funkcje wyliczające dane statystyczne dla metody równych udziałów, których głównym celem jest łatwe przedstawienie przebiegu tejże metody wyborcom i innym zainteresowanym osobom. Dzięki wprowadzeniu dedykowanego narzędzia do generacji i wizualizacji wyżej wspomnianych statystyk, każda osoba może w kilka minut utworzyć stronę ukazującą te statystyki z użyciem pliku w formacie Pabulib, bez potrzeby zrozumienia działania biblioteki i zaimplementowanego w ramach tej pracy kodu. Dodatkowo możliwość eksportu zwizualizowanych statystyk do statycznego pliku HTML+CSS pozwala na ich łatwiejsze udostępnianie.

Przedstawione narzędzia mają potencjał zostać wykorzystane w niedalekiej przyszłości - co roku bowiem organizowane są wybory budżetu partycypacyjnego, a w przeróżnych miastach i gminach, które zaadaptowały już metodę równych udziałów, możliwe będzie łatwe udostępnianie statystyk przedstawionych w schludnej formie. To z kolei nie tylko przyczyni się do zwiększenia świadomości wyborców głosujących w wyborach, ale także zwiększy okoliczną rozpoznawalność metody równych udziałów - co jest korzystne zarówno dla samej metody, jak i konceptu budżetu partycypacyjnego, który od lat walczy z niską frekwencją w wyborach.

Dzięki przemyślanej architekturze pełnego rozwiązania, jak i poszczególnych jego komponentów, możliwy jest dalszy rozwój opisanych narzędzi. Można skupić się na rozwijaniu narzędzia do generacji i wizualizacji statystyk w oparciu o obliczenia wykonywane w bibliotece Pabutools. Pozwoli to na jeszcze klarowniejsze przedstawienie sposobu działania metody równych udziałów. Oprócz tego, rozwinąć można też inne platformy do zarządzania głosowaniami na budżet partycypacyjny, takie jak Decidim czy open-active-voting, aby umożliwić eksport ich instancji i profili wyborczych do pliku w formacie Pabulib.

# Bibliografia

- [SOU98] de SOUSA SANTOS B. *Participatory Budgeting in Porto Alegre: Toward a Redistributive Democracy. Politics & Society*. 1998. URL: <https://doi.org/10.1177/0032329298026004003>.
- [HN21] Aziz Haris i Shah Nisarg. *Participatory Budgeting: Models and Approaches*. Springer International Publishing, 2021. URL: [https://doi.org/10.1007/978-3-030-54936-7\\_10](https://doi.org/10.1007/978-3-030-54936-7_10).
- [Bar+24] Xabier E. Barandiaran, Antonio Calleja-López, Arnau Monterde i Carol Romero. *Decidim, a Technopolitical Network for Participatory Democracy*. Springer Nature Switzerland, 2024. URL: [https://doi.org/10.1007/978-3-031-50784-7\\_1](https://doi.org/10.1007/978-3-031-50784-7_1).
- [PPS22] Dominik Peters, Grzegorz Pierczyński i Piotr Skowron. *Proportional Participatory Budgeting with Additive Utilities*. 2022. arXiv: [2008.13276](https://arxiv.org/abs/2008.13276) [cs.GT]. URL: <https://arxiv.org/abs/2008.13276>.
- [Bri+23] Markus Brill, Stefan Forster, Martin Lackner, Jan Maly i Jannik Peters. *Proportionality in Approval-Based Participatory Budgeting*. 2023. arXiv: [2302.03672](https://arxiv.org/abs/2302.03672) [cs.GT]. URL: <https://arxiv.org/abs/2302.03672>.
- [Fal+23] Piotr Faliszewski, Jarosław Flis, Dominik Peters, Grzegorz Pierczyński, Piotr Skowron, Dariusz Stoliczki, Stanisław Szufa i Nimrod Talmon. *Participatory Budgeting: Data, Tools, and Analysis*. 2023. arXiv: [2305.11035](https://arxiv.org/abs/2305.11035) [cs.GT]. URL: <https://arxiv.org/abs/2305.11035>.