Najlepszy agent

Wybór najlepszego z agentów na podstawie metody porównywania parami (PC) i problemu Max-SAT

aplikacja webowa

Wykonał:
Dawid Talaga
EAIiIB, V rok
Kraków, 2018

I. Cel projektu

Celem projektu jest utworzenie aplikacji webowej, która wybierze najlepszego agenta (dowolny obiekt o określonych parametrach). Dokona tego wykorzystując dwie różne metody
i prezentując wyniki każdej z nich:

1. Metoda porównywania parami (PC).
2. Problem maksymalnej spełnialności (Max-SAT).

Aplikacja ma być narzędziem łatwym i przyjemnym w obsłudze, którego uruchomienie i używanie nie powinno sprawiać większych problemów, a wręcz zachęcać do zagłębienia się w tematykę.

II. Metody wykorzystywane w aplikacji

1. Metoda porównywania parami (The Pairwise Comparison Method)

Metoda porównywania parami jest pozwala na wybór najlepszej z alternatyw, kiedy do podjęcia jest określona decyzja. Jest jedną z metod podejmowania decyzji. Opiera się na założeniu, że zamiast porównywać wszystkie opcje (alternatywy) od razu, łatwiej jest porównać je parami, a następnie zebrać wszystkie wyniki. Takie podejście nie tylko upraszcza podjęcie decyzji, ale daje także bardziej wiarygodne rezultaty.

Porównywanie składa się z dwóch etapów:

1. Porównywanie alternatyw - w tym etapie użytkownik określa jak ważny jest dla niego każdy parametr, mówiąc potocznie - jak bardzo zależy mu na każdym z parametrów.
2. Porównywanie agentów - odbywa się poprzez porównanie każdej wartości danego parametru z każdą inną, np. jeśli agent posiada parametr *kolor*, mamy 3 agentów, których kolor jest określony jako *niebieski, czerwony, zielony*, to użytkownik musi dokonać trzech porównań: *niebieski* vs *czerwony*, *niebieski* vs *zielony*, *niebieski* vs *czerwony*.

Każde porównanie to przypisanie wartości liczbowej określającej, która wartość jest lepsza i jak bardzo. W prostych aplikacjach często wykorzystuje się suwak, którego położenie określa wybór użytkownika.

Po dokonaniu porównań i wywołaniu metody, użytkownik otrzymuje odpowiedź, która przedstawia wartość każdego agenta (im wyższa, tym agent jest *lepszy*). Wartości te sumują się do 1 lub do 100% po przeskalowaniu.

Metoda porównywania parami posiada silne podłoże matematyczne i opiera się na obliczeniach algebraicznych. W celu poszerzenia wiedzy o metodzie odsyłam do mojej pracy inżynierskiej, dostępnej pod adresem: https://github.com/katal24/PC_engineer.

2. Problem maksymalnej spełnialności (Maximum satisfiability problem)

Problem spełnialności (SAT) to zagadnienie rachunku zdań, które określa, czy dla danej formuły logicznej istnieje takie wartościowanie (podstawienie) zmiennych, żeby formuła była prawdziwa. Stwierdzenie, czy zadana formuła logiczna jest spełnialna, to zagadnienie teorii złożoności obliczeniowej.
Mówiąc wprost: zadajemy formułę logiczną i pytamy, czy jest możliwe, aby była ona spełniona.

W aplikacji posłużymy się odmianą SAT'u, jakim jest problem maksymalnej spełnialności (Max-SAT). Rozszerza on możliwości SAT'u. Na wejściu otrzymuje formułę logiczną w koniunkcyjnej postaci normalnej (CNF). Kiedy formuła logiczna nie jest spełniona, poszukuje takiego wartościowania zmiennych, które zmaksymalizuje liczbę spełnionych klauzul.

Do obliczenia problemu maksymalnej spełnialności wykorzystuje się specjalne prover'y.

III. Schemat działania aplikacji

Poniżej omówię każdy z etapów działania aplikacji.

IV. Etapy działania aplikacji

1. Określenie parametrów agentów

Aplikacja służy do wyboru najlepszego agenta z całej populacji agentów. Populacja może mieć dowolną wielkość.

Pierwszym krokiem jest określenie jakie cechy (atrybuty) posiada każdy agent. To elementy, które charakteryzują agentów i na podstawie których aplikacja wybierze najlepszego. Mogą to być wartości numeryczne (np. długość) lub inne (np. kolor).

Po określeniu atrybutów, użytkownik charakteryzuje każdego z agentów. Czyni to poprzez podanie wartości dla każdego atrybutu, np. dla atrybutu *długość* podaje wartość *100*, a dla atrybutu *kolor* podaje wartość *zielony*. Dla każdego agenta musi zostać określony każda z cech.

2. Metoda porównywania parami

2a. Ustalenie szczegółów porównywania

Zanim użytkownik przystąpi do porównywania, aplikacja daje możliwość uproszczenia tego procesu. Często zdarza się, że wartości parametrów są numeryczne, np. gdy parametr określa długość, szerokość, szybkość, czy inną wielkość. W takim przypadku porównywanie może okazać się żmudne. Aplikacja pozwala więc zdecydować użytkownikowi, żeby takie wartości porównane zostały same, a wartością porównania stanie się stosunek jednej wartości do drugiej. Jeśli więc pierwszy agent w polu *długość* ma wartość *500*, drugi *1000*, to wartość porównania będzie ½, co oznacza, że drugi agent jest pod tym względem dwa razy lepszy. Użytkownik może także zdecydować, że mniejsza wartość jest korzystniejsza, wtedy porównanie uzyska wartość 2.

Do określenia, które wartości mają być porównywane ręcznie przez użytkownika, a które zostaną automatycznie przypisane, służy tabela. Warto zaznaczyć, że automatyczne porównania mają sens tylko dla parametrów określanych przez wartości numeryczne.

2b. Wykonanie porównań

Kolejnym krokiem jest dokonanie porównań. Służy do tego suwak, którego pozycja określa która z dwóch wartości jest lepsza (według zdania użytkownika) oraz o ile w porównaniu do drugiej wartości.

Jeżeli suwak pozostanie na środku, to obie wartości są równoważne.
Jeżeli suwak zostanie przesunięty w którąś stronę, to wartość, która jest po danej stronie staje się preferowana. Im ważniejsza jest dana wartość, tym dalej od środka powinien znaleźć się suwak. Jeśli użytkownik przesunie go do końca, oznacza to *zdecydowanie lepszy*, jeśli przesunie go do ¾ suwaka, oznacza to *dużo lepszy*, a jeśli przesunie tylko trochę, oznacza to *trochę lepszy*.

Najpierw porównywane są parametry (nie ma możliwości automatycznego porównania parametrów), następnie poszczególne wartości w ramach danych parametrów.

2c. Uruchomienie metody PC

Po wykonaniu wszystkich porównań, użytkownik uruchamia metodę PC, szczegóły działania serwera, który wykonuje obliczenia zostały przedstawione w kolejnym rozdziale.

3. Metoda wykorzystująca problem maksymalnej spełnialności

3a. Zdefiniowane słownika

W celu wykorzystania MaxSAT'u, konieczne jest określenie wspólnego słownictwa, tak, aby równe wartości zawsze określane były w ten sam sposób. Jest to konieczne w celu dobrego porównywania, w przeciwnym razie użytkownik określając, np. *długość* może wpisać *1000* lub 1 000, a dla algorytmu te dwie wartości będą różne. W tym celu stosujemy słownik, jest to zbiór wszystkich wartości, które mogą określać agentów.

Uwaga! Każda wartość, którą wpisujemy, aby określić agentów (opisane w kroku 1.) musi być zdefiniowana w słowniku. Jeśli więc użytkownik zakłada wykorzystanie tej metody wyboru najlepszego agenta, warto rozpocząć pracę z aplikacją właśnie od zdefiniowania słownika.

3b. Zdefiniowanie idealnego agenta

Kolejnym krokiem jest określenie cech idealnego agenta. Jego cechy zostaną porównane z populacją agentów i na tej podstawie MaxSAT określi, który posiada najwięcej cech wspólnych z idealnym agentem.

Do zdefiniowania idealnego agenta służy specjalna tabela. Użytkownik musi w niej określić każdy z parametrów. Każda wartość może zostać wyspecyfikowana poprzez podanie dokładnej wartości (np. *kolor* równy *zielony)* lub określenie maksymalnej pożądanej wartości (np. *długość* mniejsza niż *50*), lub określenie minimalnej pożądanej wartości (np. *waga* większa niż *200*).

3c. Uruchomienie prover'a MaxSAT

Po określeniu wszystkich cech idealnego agenta, użytkownik uruchamia prover MaxSAT.

Szczegóły działania prover'a, który wykonuje obliczenia zostały przedstawione w jednym z kolejnych rozdziałów.

4. Przeglądanie wyników

Wyniki prezentowane są osobno dla każdej z metod. Daje to możliwość ich szybkiego porównania oraz przeglądania wyników, gdy zastosowano tylko jedną metodę.
Wynikiem aplikacji dla każdej z metod jest lista agentów posortowana od *najlepszego* do *najgorszego*. W różny sposób prezentowane są jednak wartości:

1. Dla metody PC - procentowe wartości, które mówią jak dobry jest dany agent. Wartości sumują się do *100%*.
2. Dla metody MaxSAT - ilość parametrów, które pokrywają się z idealnym agentem. Maksymalną wartością jest ilość parametrów agenta.

V. Implementacja metod wyboru najlepszego agenta

Operacje matematyczne, które odpowiadają za wyznaczenie najlepszego agenta wykonywane są po stronie backendowej, która pełni rolę serwera. Odbiera on dane od części frontendowej, wykonuje odpowiednie obliczenia, a następnie zwraca wyniki do frontendu. Poniżej przedstawiam implementację obu metod.

1. Implementacja metody Porównywania Parami (PC)

Metoda zaimplementowana jest w języku Java, z wykorzystaniem biblioteki *PairwiseComparisons*. Udostępnia ona wiele metod (około 50), które są związane z PC. W aplikacji wykorzystywana jest metoda *ahp*, która przyjmuje dane o agentach i zwraca gotowy ranking agentów. O blibliotece szczegółowo można poczytać pod adresem: https://github.com/katal24/Talaga_praca_inz.

UWAGA! Biblioteka wykorzystuje język R, aby więc uruchomić część backendową aplikacji należy zainstalować na serwerze pakiet tego języka. Jest on darmowy i dostępny pod adresem: http://cran.us.r-project.org.

2. Zastosowanie MaxSAT

Pierwszym krokiem jest zbudowanie formuły logicznej, która zostanie wysłana do prover'a.
Prover MaxSAT przyjmuje formułę złożoną z liczb całkowitych i operatorów. Wartości parametrów agentów (również idealnego agenta) są więc przekształcane na liczby całkowite. Tworzona jest formuła postaci ...

Problem MaxSAT jest rozwiązywany z pomocą prover'a SAT4J. Aplikacja załącza bibliotekę, która obsługuje odpowiednie metody powiązane z MaxSAT'em. W aplikcaji wykorzystywana jest metoda *findModel* udostępniana przez klasę *WeightedMaxSatDecorator*.
O szczegółach biblioteki można poczytać pod adresem: http://www.sat4j.org.

VI. Budowa aplikacji

6. Budowa aplikacji

Aplikacja składa się z dwóch części: backend i frontend. Szczegóły ich budowy zostały przedstawione poniżej.

6.1. Backend

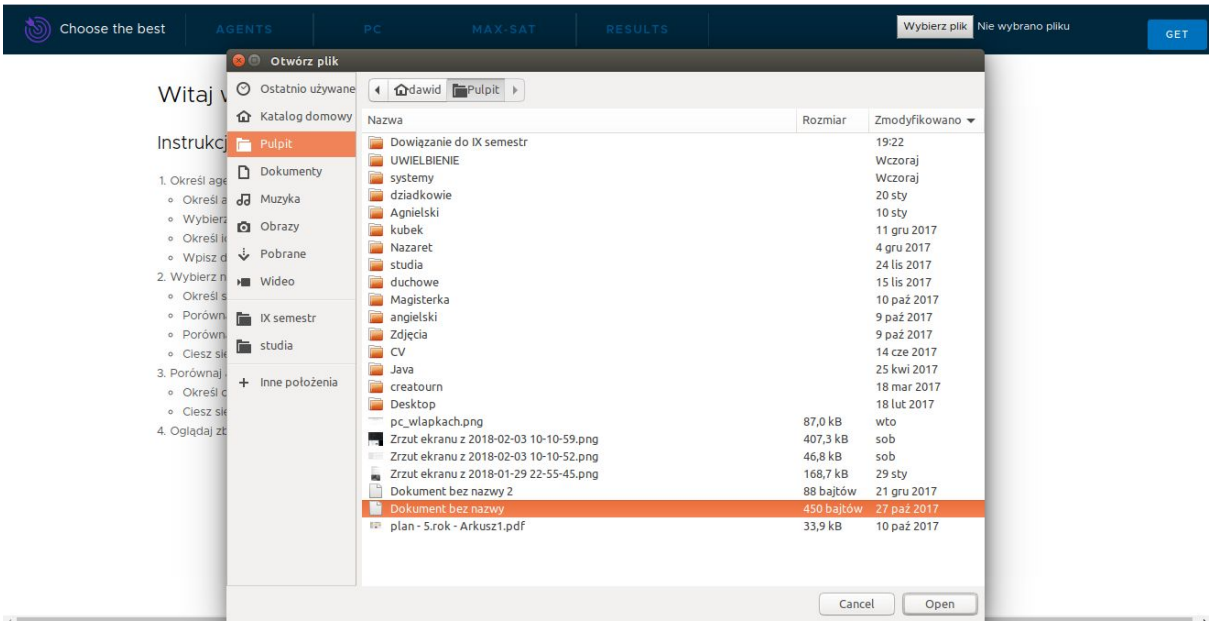
TODO

6.2. Frontend

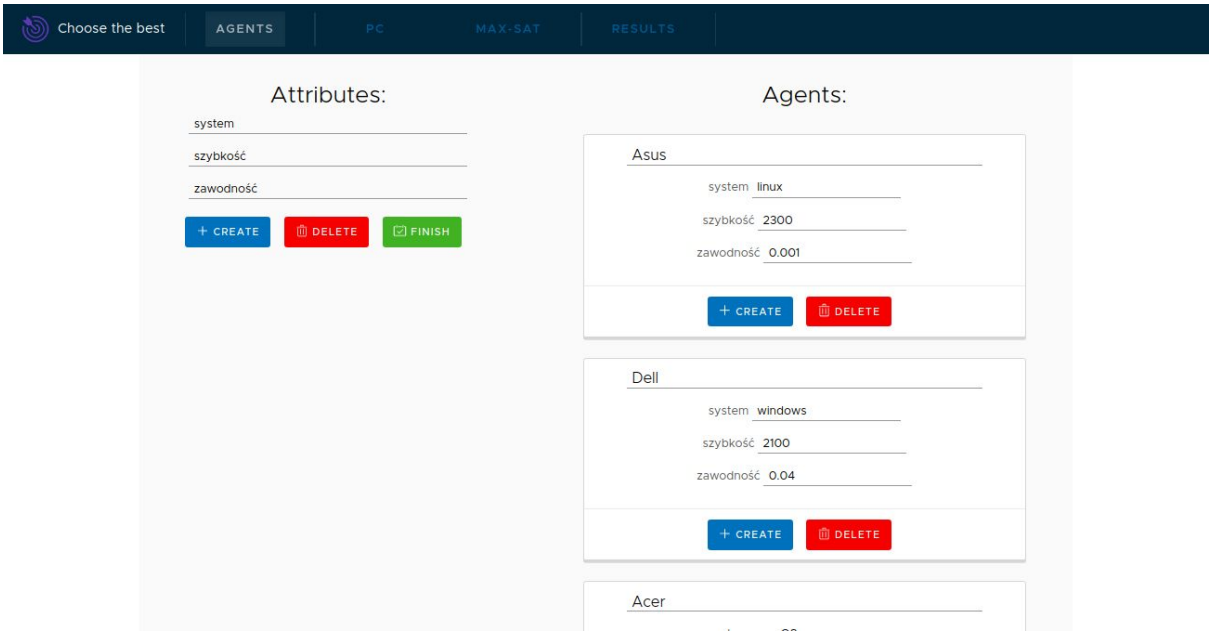
TODO

VII. Działanie aplikacji - przykład

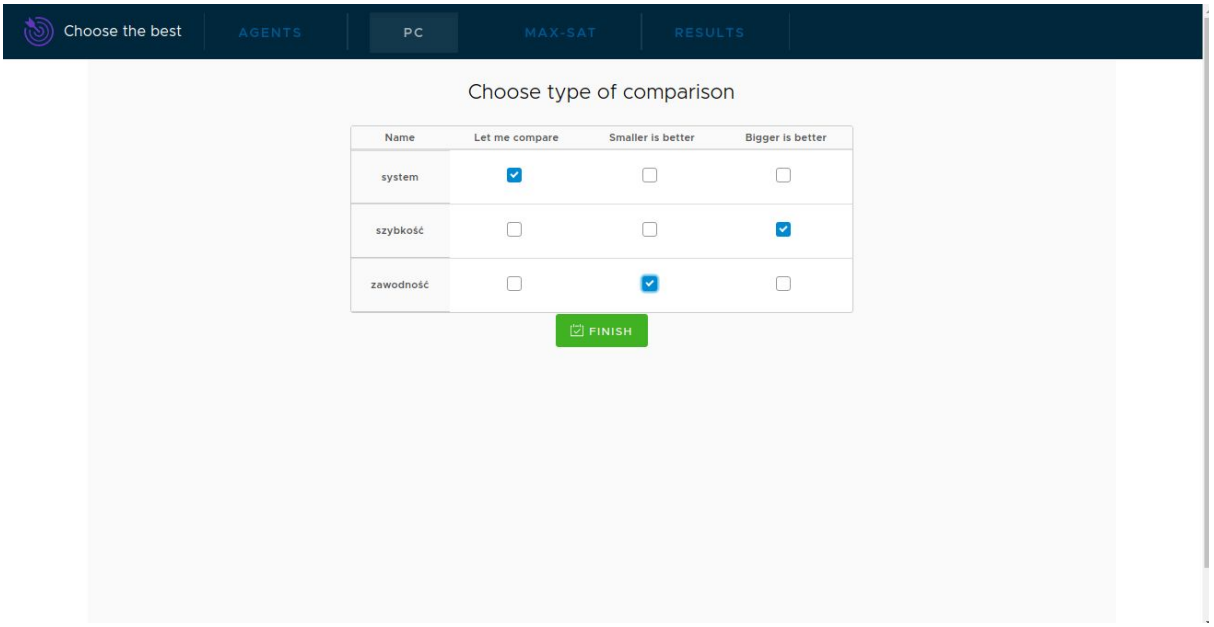
1. Wybór pliku ze słownikiem



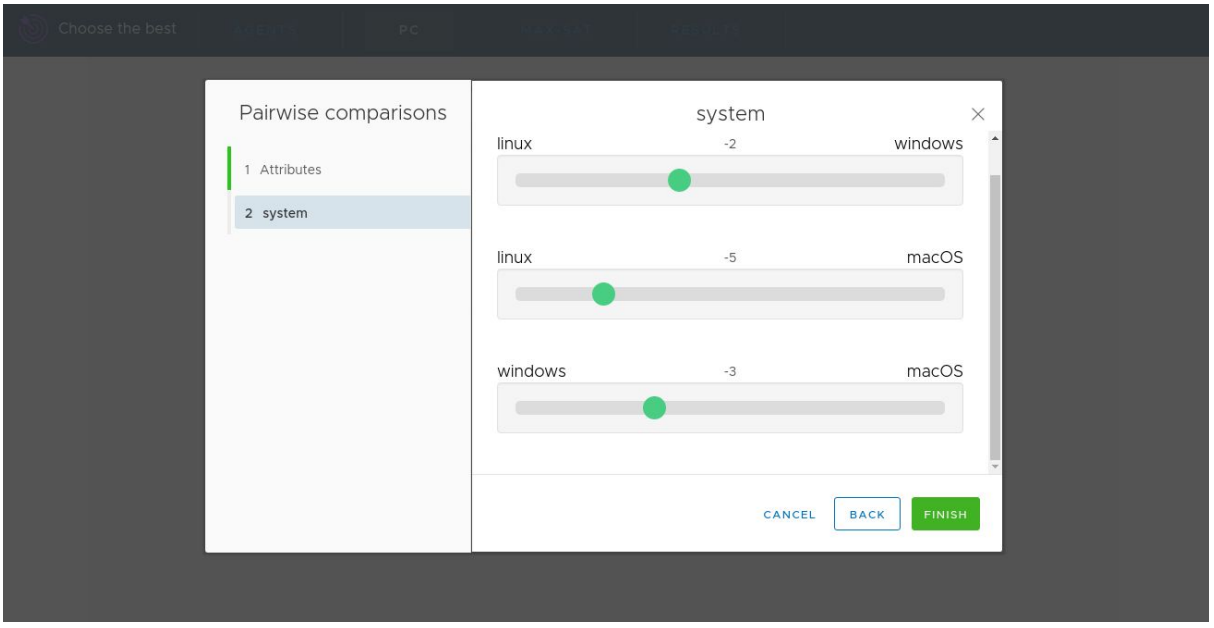
2. Utworzenie agentów



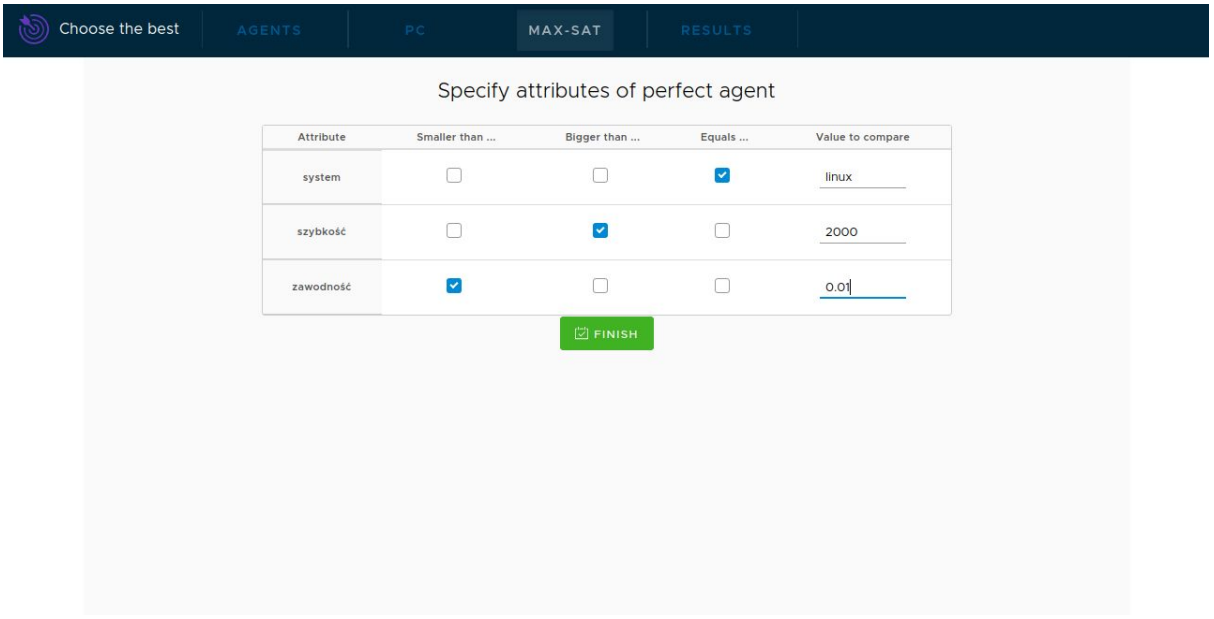
3. Ustalenie szczegółów porównania



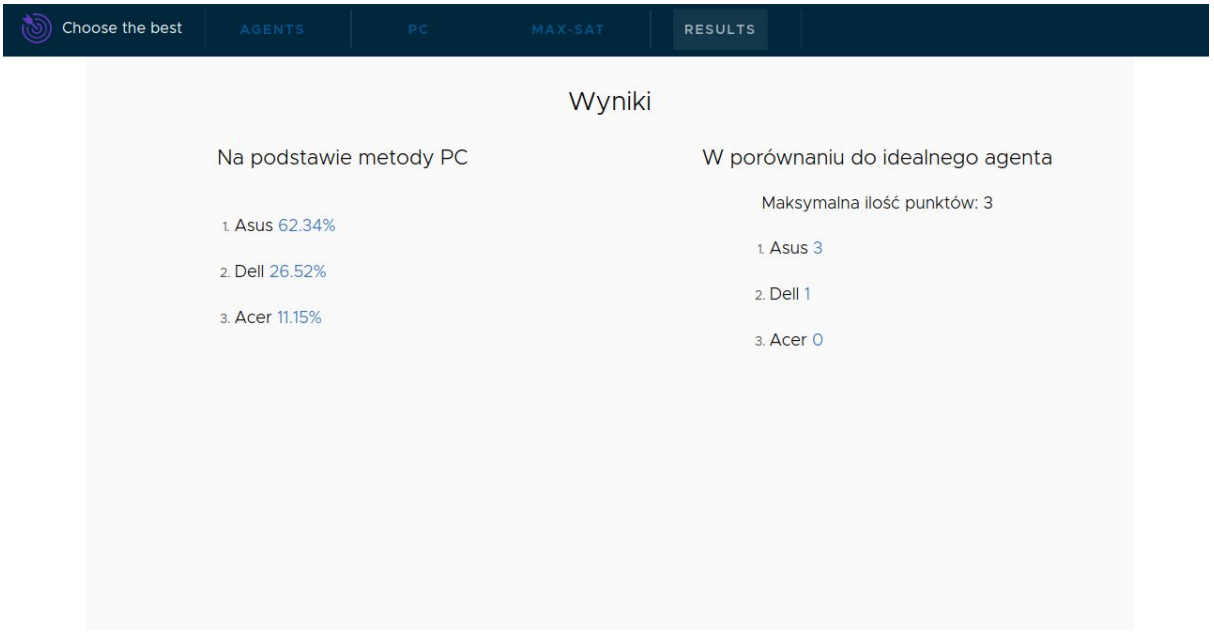
4. Wykonanie porównań



5. Zdefiniowanie idealnego agenta



6. Przeglądanie wyników



VIII. Uwagi dotyczące używania aplikacji

TODO