



POLITECHNIKA ŚLĄSKA
WYDZIAŁ AUTOMATYKI, ELEKTRONIKI I INFORMATYKI
KIERUNEK INFORMATYKA

Praca dyplomowa magisterska

Podejmowanie decyzji w grach komputerowych
w oparciu o analizę danych historycznych

Autor: Denis Wychowalek

Kierujący pracą: dr Ewa Lach

Gliwice, październik 2016

Spis treści

| | |
|---|-----------|
| 1. Wstęp..... | 5 |
| 2. Wprowadzenie | 6 |
| 2.1. Elementy systemu podejmowania decyzji i klasyfikacja problemów decyzyjnych..... | 6 |
| 2.2. Teoria decyzji w informatyce i grach komputerowych | 7 |
| 3. Przegląd metod i struktur wspomagania decyzji..... | 10 |
| 3.1. Drzewa decyzyjne | 10 |
| 3.1.1. Budowa drzewa decyzyjnego | 11 |
| 3.1.2. Algorytmy do budowy drzewa | 14 |
| 3.2. Logika rozmyta..... | 15 |
| 3.2.1. Rozmywanie | 16 |
| 3.2.2. Wnioskowanie..... | 17 |
| 3.2.3. Wyostrzanie | 18 |
| 3.3. Sztuczne sieci neuronowe..... | 18 |
| 3.3.1. Budowa sztucznej sieci neuronowej | 19 |
| 3.3.2. Metody uczenia sztucznej sieci neuronowej | 20 |
| 3.4. Sieci bayesowskie..... | 21 |
| 4. Środowisko Badawcze: gra strategiczna | 24 |
| 4.1. Przygotowanie środowiska badawczego | 24 |
| 4.1.1. Sformułowanie problemu decyzyjnego | 24 |
| 4.1.2. Wymagania funkcjonalne i нефункционалне Środowiska Badawczego | 25 |
| 4.2. Funkcje dostępne w aplikacji..... | 26 |
| 4.3. Obsługa..... | 29 |
| 4.4. Specyfikacja wewnętrzna | 30 |

| | |
|---|-----------|
| 4.4.1. Organizacja danych | 30 |
| 4.4.2. Klasy | 31 |
| 4.5. Implementacja wybranych metod | 32 |
| 4.5.1. Drzewa decyzyjne | 32 |
| 4.5.2. Logika rozmyta | 34 |
| 5. Testowanie i analiza wyników | 36 |
| 5.1. Pomiar czasu wykonania algorytmów | 36 |
| 5.2. Test jakości otrzymanych wyników | 39 |
| 6. Podsumowanie | 44 |
| Bibliografia | 46 |
| Spis rysunków | 48 |
| Spis tabel | 49 |
| Dodatek A | 50 |

1. Wstęp

Twórcy chcąc dostarczyć graczom coraz bardziej realistyczny świat w grach komputerowych są zmuszeni do zmagania się z problemem podejmowania decyzji na różnych etapach gry. Aby poradzić sobie z tym zagadnieniem korzystają z różnych metod wspomagania procesu decyzyjnego.

Celem niniejszej pracy dyplomowej jest dokonanie przeglądu dostępnych metod rozwiązywania problemu podejmowania decyzji, ich zastosowania w grach komputerowych oraz zaimplementowanie wybranych algorytmów i porównanie ich.

Dalsza część pracy wygląda następująco: rozdział drugi jest wprowadzeniem do tematyki związanej z teorią decyzji oraz jej zastosowaniem. Opisane są elementy, z których powinien się składać system podejmowania decyzji i klasyfikacja problemów decyzyjnych. W trzecim rozdziale znajduje się analiza teoretyczna znanych metod wspomagania decyzji. Rozdział czwarty zawiera opis środowiska badawczego, jego funkcjonalności, informacje o implementacji wybranych metod oraz analiza otrzymanych wyników. Na końcu autor podsumowuje przebieg pracy, napotkane problemy i otrzymane wyniki.

2. Wprowadzenie

Podejmowanie decyzji w życiu człowieka odgrywa bardzo dużą rolę. Problem pojawia się w momencie, gdy istnieje kilka opcji do wyboru i poprzez swoje preferencje lub obliczenia, człowiek musi dokonać wyboru optymalnej decyzji licząc się z jej możliwymi konsekwencjami. Z problemem można się spotkać pracując z pewnym zestawem danych, bądź też wykonując codzienne obowiązki. Czasami gdy problem decyzyjny przekracza ludzkie możliwości, część zadań zostaje przekazana maszynom lub systemom, które w oparciu o zbiór reguł i zbiór danych musi utworzyć kilka dostępnych wariantów decyzyjnych, dokonać ich analizy i wybrać jedną opcję spośród wszystkich dostępnych, która będzie optymalna lub gdy nie można dokonać decyzji w przewidywanym czasie wybrać tą, która jest możliwie najlepsza w danym momencie. Ponieważ problem decyzyjny dotyczy wielu aspektów życia człowieka zajmuje się tym problemem oddzielna dziedzina nauki. Teoria decyzji znajduje swoje zastosowanie i jest rozwijana przez wiele dziedzin nauki przez m.in. kognitywistykę, matematykę, informatykę, zarządzanie, ekonomię, medycynę, socjologię i filozofię. Wiedza z każdej z nich znajduje zastosowanie w pewnym stopniu w innych dziedzinach nauki [1].

2.1. Elementy systemu podejmowania decyzji i klasyfikacja problemów decyzyjnych

System podejmowania decyzji powinien składać się z kilku elementów. Na początku warto zacząć od sformułowania problemu decyzyjnego, który ma zostać poddany analizie. Należy określić jakiej dziedziny ma on dotyczyć i jaki cel ma zostać osiągnięty, jakie rozwiązanie ma zostać uznane za optymalne lub możliwie jak najlepsze. Następnie powinno sformułować się reguły, które będą dokonywać selekcji wariantów postępowania, by wykluczyć te, które nie będą należały do dziedziny rozwiązań. Dla stworzonych różnych decyzji należy przygotować zestaw obliczeń, które dokonają oceny jakościowej lub ilościowej każdej decyzji. W przypadku poruszania problemu ekonomicznego, może być to ocena ryzyka inwestycyjnego na giełdzie, gdzie należy obliczyć jaka może być możliwa strata w

przypadku złej inwestycji i jaki będzie zysk w przypadku sukcesu. Ostatnim etapem jest wybór najlepszej decyzji spośród wszystkich dostępnych. W przypadku bardziej skomplikowanych problemów należy liczyć się z tym, że dokonanie jednej decyzji będzie powodowało powstawanie kolejnych problemów decyzyjnych i czas oczekiwania na ostateczną decyzję może się wydłużyć lub będzie konieczne przeznaczenie większych zasobów do niezbędnych obliczeń [2].

Problemy decyzyjne można podzielić na kilka grup, które w zależności od rodzaju samego problemu, zestawu dostarczonych danych oraz pewności przy podejmowaniu decyzji wymagają specyficznego dla danego typu podejścia. Ze względu na warunek pewności można je podzielić na:

- Pewne – czyli takie których możliwe konsekwencje zostały jasno określone.
- Ryzykowne – w których należy określić możliwe korzyści i negatywne konsekwencje. Może być to związane z określeniem prawdopodobieństwa słuszności podjętej decyzji.
- Niepewne – gdzie nie da się przewidzieć słuszności danego wyboru. Nie można obliczyć z jakichś powodów możliwych konsekwencji np. gdy posiadany jest niepełny zbiór danych potrzebnych do oceny danego wariantu.

2.2. Teoria decyzji w informatyce i grach komputerowych

W informatyce teoria decyzji znajduje zastosowanie w projektowaniu komputerowych systemów wspomagania decyzji. W momencie gdy ilość danych jest bardzo duża i kłopotliwe staje się wskazanie optymalnego rozwiązania, wykorzystuje się w zależności od typu danych i celu który ma zostać osiągnięty różne rozwiązania. Jeśli problem decyzyjny dotyczy dużego przedsięwzięcia i konieczne jest podejście analityczne do problemu, to można wykorzystać do tego celu np. hurtownie danych, algorytmy związane ze sztuczną inteligencją i modele statystyczne. Do analizy danych wykorzystywana jest wiedza z zakresu eksploracji

danych (ang. *Data mining*), dzięki której możliwe jest znajdowanie zależności pomiędzy danymi znajdującymi się w bazie danych.

W przypadku gier komputerowych problem podejmowania decyzji znajduje duże zastosowanie [3]. Twórcy mogą mieć chęć stworzenia sztucznej inteligencji, która będzie reagowała na poczynania gracza lub będzie analizowała różne parametry pobierane w trakcie rozgrywki, by nauczyć przeciwnika sterowanego przez SI taktyki gracza.

W grach RPG pozwala się graczowi wybrać jedną z kilku dostępnych opcji dialogowych, które mają wpływ w mniejszym lub większym stopniu na zachowanie się postaci kierowanych przez sztuczną inteligencję lub mają wpływ na to jaki świat uda się ukształtować graczowi poprzez jego wybory. Dzięki temu można stworzyć zależność np. jaka frakcja będzie przyjazna graczowi, a jaka wroga i jakie korzyści i utrudnienia przy okazji go spotkają. Możliwe jest też nadanie cech ludzkich postaci, z którymi gracz wchodzi w interakcję: gdy oszuka się ją będzie w stosunku do gracza nieufna, gdy wykona się dla niej zadanie może być skłonna do pomocy.

Teoria decyzji znajduje również zapotrzebowanie w innych gatunkach gier. W grach wyścigowych, sztuczna inteligencja może uczyć się od gracza na podstawie różnych parametrów stylu jazdy gracza. Określa jego stopień agresji podczas wyścigu, dokładność wchodzenia w zakręty, prędkość itd. Na podstawie tych czynników w późniejszym etapie jest w stanie sklasyfikować profil gracza i nauczyć w oparciu o tą wiedzę jednostki kontrolowane przez sztuczną inteligencję stylu jazdy gracza, dostosować jej poziom trudności w zależności od umiejętności.

W grach strategicznych można znaleźć bardzo dużo sytuacji, gdy konieczne jest podjęcie jakiejś decyzji. Sztuczna inteligencja w tym przypadku powinna decydować np. o taktyce obronnej w przypadku zaatakowania przez wroga jednostki bazy lub planować atak na bazę gracza. Problemem decyzyjnym może być też taktyka postępowania w zależności od warunków zastanych w grze np. może oszacować opłacalność wydobywania surowców, sens rozbudowy bazy, systemów obronny lub kupno konkretnych jednostek, analizować możliwe warianty ataku, znajdować słabe punkty, w które można uderzyć by osłabić przeciwnika. W

dużej mierze wykorzystanie problemu decyzyjnego w grze zależy od osoby projektującej, to od niej zależy jakie dane będą poddawane analizie i przed jakimi decyzjami postawi się sztuczną inteligencję. Nie można jednak zapomnieć o zachowaniu zdrowego rozsądku i należy pamiętać, że czas podjęcia decyzji powinien być wystarczająco krótki, żeby nie dopuścić do długiej analizy i nie wpłynąć negatywnie na inne elementy gry, jak np. płynność rozgrywki, długie czasy ładowania itp.

3. Przegląd metod i struktur wspomagania decyzji

W przypadku gier komputerowych problem decyzyjny rozwiązuje się wykorzystując różne metody i struktury wspomagania decyzji. Często korzystają one z wiedzy powiązanej z różnymi dziedzinami nauki, by móc zaprojektować taki system decyzyjny, który będzie wskazywał spośród kilku dostępnych opcji tą, która jest najlepsza. Żeby móc otrzymać taki wynik należy sformułować odpowiednie reguły, wskazać odpowiednie kryteria, według których będzie podejmowana decyzja lub wybrać odpowiedni zestaw wzorów. Każdą z metod przedstawionych w tym rozdziale można wykorzystać dla odpowiedniego zestawu danych, niektóre można ze sobą łączyć w celu zwiększenia dokładności klasyfikacji poszczególnych wariantów decyzyjnych. Do tego celu można wykorzystać między innymi drzewa decyzyjne, elementy logiki rozmytej, sztuczne sieci neuronowe i sieci bayesowskie.

3.1. Drzewa decyzyjne

Drzewa decyzyjne są jedną z najbardziej popularnych struktur stosowanych w przedstawieniu problemu decyzyjnego. Przedstawiane w postaci grafu, którego węzły są poszczególnymi opcjami do wyboru i ich możliwymi konsekwencjami. Węzły decyzji i konsekwencji występują naprzemiennie i są analizowane do momentu, kiedy cel zostanie osiągnięty lub do chwili, gdy nie będzie już możliwa analiza. Każda z wybranych ścieżek kończy się węzłem końcowym [5].

Często wykorzystuje się tę metodę do zaprojektowania sztucznej inteligencji w grach planszowych typu szachy lub warcaby. W przypadku tego typów gier drzewo decyzyjne może posłużyć do analizy możliwych ruchów w zależności od aktualnej sytuacji na planszy. Można jednak znaleźć dla tej metody zastosowanie w bardziej skomplikowanych grach komputerowych [3].

Metoda ta może też być wykorzystywana do eksploracji danych [5], dzięki której możliwe jest znalezienie zależności pomiędzy danymi wejściowymi np. gracz wystawił 50 jednostek czołgów, pokonał wszystkie jednostki piechoty wroga i stracił przy takiej taktyce tylko 10 czołgów.

Zaletą zastosowania tej metody jest czytelny zapis problemu decyzyjnego, pomagają znaleźć odpowiednią zależność między danymi i mogą dobrze pracować dla dużego zestawu danych. Wadami jednak może jednak być fakt, że nie nadają się do rozwiązywania każdego problemu decyzyjnego. W przypadku małego zestawu danych mogą okazać się gorszą metodą klasyfikacji [5].

3.1.1. Budowa drzewa decyzyjnego

Przygotowanie i podział danych

Pierwszym krokiem w celu zbudowania drzewa decyzyjnego jest podział otrzymanych danych wejściowych na atrybuty. Źródłem danych może być np. baza danych lub zbiór plików, w których zapisane są informacje przeznaczone do analizy. Atrybuty mogą być odpowiednimi kategoriami i rodzajami danych. Podział na poszczególne atrybuty może być uzależniony od preferencji człowieka. Ważne jest również stworzenie odpowiednich klas określających węzeł końcowy, dzielące zbiór wartości w poszczególnych atrybutach na mniejsze grupy.

Wyznaczanie parametrów

Po dokonaniu podziału zbioru danych należy wyznaczyć, taki atrybut, który najlepiej podzieli zbiór podczas budowy drzewa. W tym celu konieczne jest wyznaczenie kilku parametrów, które ułatwią wybór odpowiedniego atrybutu i wartości [4].

Pierwszym parametrem jest **entropia**, która oznacza informację przechowywaną przez rozkład prawdopodobieństwa. Aby ją wyznaczyć potrzebne jest wyznaczenie prawdopodobieństwa wystąpienia każdej z możliwych klas decyzyjnych dla całego zbioru wartości. Wyznaczenie entropii zostało przedstawione na wzorze (1), w którym P_i – oznacza prawdopodobieństwo dojścia do danej klasy decyzyjnej.

$$I(P) = \sum_{i=1}^n P_i * \log_2(P_i) \quad (1)$$

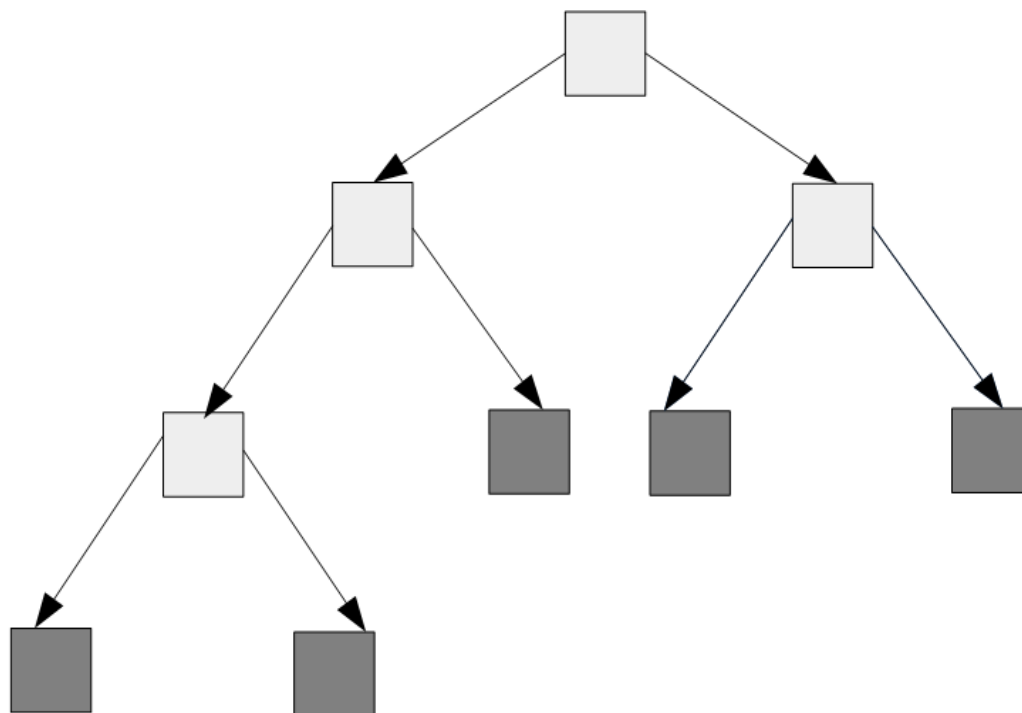
Aby móc wybrać odpowiedni atrybut, który najlepiej podzieli nam zbiór wartości, koniecznej jest wyznaczenie **zysku informacji** dla każdego atrybutu. Wybierany spośród wyznaczonych wartości jest ten atrybut, którego zysk informacji jest największy. Parametr ten jest różnicą przedstawioną na wzorze (2), gdzie $I(P)$ jest entropią dla całego zbioru rozwiązań, a $I(P | a)$ jest entropią dla konkretnego atrybutu.

$$Gain(P|a) = I(P) - I(P|a) \quad (2)$$

W momencie, gdy zbiór wartości w każdym z atrybutów jest bardziej zróżnicowany, może pojawić konieczność wyznaczenia jeszcze jednego parametru – **współczynnika przyrostu informacji**, który został przedstawiony na wzorze (3) i będący ilorazem zysku informacji i podziału informacji dla danego atrybutu.

$$GainRatio(T|a) = \frac{Gain(T|a)}{\sum_{j=1}^m \frac{|T_j|}{|T|} * \log_2 \frac{|T_j|}{|T|}} \quad (3)$$

Tworzenie drzewa trwa do momentu, kiedy podział wartości jest możliwy i wyznaczanie wszystkich parametrów i kolejne podziały są wykonywane rekurencyjnie. Podział kończy się liściem, w którym znajduje się jedna z klas decyzyjnych. Przykładowe drzewo decyzyjne zostało przedstawione na rysunku 1, gdzie węzły odpowiadają zbiorowi danych i dzielą go według wartości atrybutu, każda ścieżka kończy się liśćmi, w których znajdują się klasy decyzyjne przedstawione, jako węzeł o ciemniejszym kolorze.



Rysunek 1. Przykładowe drzewo decyzyjne

Problemy przy budowie drzewa i ich rozwiązywanie

Podczas budowy drzewa istotnym jest przeuczenia się drzewa [6]. Dzieje się tak, gdy zbiór wartości jest bardzo zróżnicowany, występuje bardzo duża ilość klas decyzyjnych lub zbiór danych jest bardzo duży.

Aby rozwiązać ten problem korzysta się przeważnie z algorytmów przycinania drzewa, które zapobiegają nadmiernego rozrostowi drzewa. Algorytmy te polegają na zastąpieniu pewnych obszarów poddrzewa liśćmi. Jednym ze sposobów jest automatyczne przycinanie podczas rozrostu drzewa, co jednocześnie może spowodować wzrost błędu klasyfikacji, natomiast zwiększy dokładność. W trakcie przycinania najprostszą metodą wyboru klasy decyzyjnej, która ma znaleźć się w liściu zastępującym poddrzewo jest podanie tej klasy, która najczęściej występuje w usuwanej części.

3.1.2. Algorytmy do budowy drzewa

Algorytm ID3

Pierwszym sposobem budowy drzewa decyzyjnego jest algorytm ID3 stworzony przez Ross Quinlan [6]. Polega na podziale danych na atrybuty oraz wyznaczenie dla nich entropii. Przy wyborze atrybutu do podziału jest brany pod uwagę atrybut, dla którego wartość entropii jest najmniejsza. Następnie zbiór danych jest dzielony na zbiory rozłączne. Cała czynność jest podczas działania algorytmu wywoływana rekurencyjnie i podczas każdej kolejnej iteracji usuwany jest jeden atrybut. Kończy swoje działanie w momencie, gdy skończy się liczba atrybutów do podziału lub gdy wszystkie wartości atrybutu decyzyjnego są takie same w danym poddrzewie.

Wadami zastosowania tej metody są m.in. nadmierny rozrost drzewa decyzyjnego i brak metody przeciwdziałającej przeuczeniu się drzewa, nie działa dobrze gdy zbiór danych jest niekompletny, może dawać przy zaszumionych danych błędne wyniki.

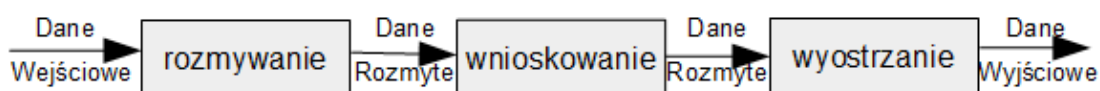
Algorytm C4.5

Modyfikacją algorytmu ID3 jest C4.5, stworzony przez tą samą osobę. Algorytm ten rozwiązuje problemy, na które nie był odporny ID3 [4]. W przypadku tej metody w celu uniknięcia przeuczenia się drzewa skorzystano z możliwości przycinania drzewa. Rozwiązano również problem z nieznaną wartością, gdzie oblicza się prawdopodobieństwo zaistnienia zdarzenia. W przypadku zaistnienia wartości numerycznych zbiór jest dzielony na wartości mniejsze lub równe i większe od wartości dzielącej zbiór. Wyliczane są oprócz wartości entropii atrybutów, wartości zysku informacji i wartość współczynnika zysku informacji i wybierany jest ten atrybut, którego wyznaczona wartość jest największa. Jest to jeden z najpopularniejszych algorytmów budowy drzewa decyzyjnego.

3.2. Logika rozmyta

Do rozwiązywania problemów decyzyjnych można wykorzystać elementy logiki rozmytej [7]. Metoda ta jest powiązana z teorią zbiorów rozmytych i z teorią prawdopodobieństwa. Stanowi ona rozszerzenie klasycznej logiki, w której zamiast określać, czy zdanie jest prawdziwe lub fałszywe, określa się w jakim stopniu wartość lub zdanie należy do konkretnego zbioru rozwiązań. Analiza danych polega na przypisaniu im wartości z zakresu pomiędzy wartościami 0 i 1.

Logika rozmyta w grach komputerowych jest często wykorzystywana w celu nadania postaciom kontrolowanym przez sztuczną inteligencję głębokiej sfery uczuciowej. Na jej podstawie sztuczna inteligencja może określić nastawienie postaci w stosunku do poczynąń gracza w świecie gry i w zależności od tego jak się zachowuje, może taka postać być miła lub zachowywać się w stosunku do niego wrogo. Zachowania postaci w tym przypadku mogą być mniej przewidywalne w porównaniu do zachowania postaci, w przypadku której zastosowano logikę klasyczną. Często wykorzystuje się logikę rozmytą w połączeniu z innymi metodami m.in. sztucznymi sieciami neuronowymi i drzewami decyzyjnymi, co jednoznacznie jest jedną z jej zalet.

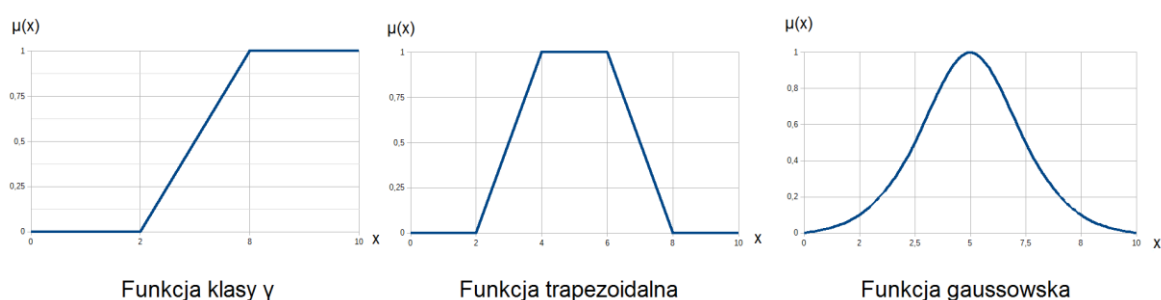


Rysunek 2. Schemat działania regulatora rozmytego

Na rysunku 2. znajduje się schemat działania regulatora rozmytego, który po otrzymaniu danych na wejście rozpoczyna proces analizy danych składającej się z kilku etapów: rozmywania, wnioskowania i wyostozania, by móc ostatecznie przekazać wyniki analizy.

3.2.1. Rozmywanie

Pierwszą czynnością po otrzymaniu danych do analizy jest rozmywanie [7]. Na tym etapie określany jest stopień przynależności poszczególnych danych do określonych zbiorów. Przypisanie konkretnych wartości odbywa się poprzez wykorzystanie funkcji przynależności. Wybór konkretnej funkcji jest zależny od rodzaju problemu oraz rodzaju danych wejściowych. Każda funkcja powinna być opisana konkretnym wzorem.



Rysunek 3. Przykładowe wykresy funkcji przynależności

Wykresy najczęściej wykorzystywanych funkcji przynależności zostały zaprezentowane na Rysunku 3. Należą do nich:

- Funkcja klasy γ , dla której we wzorze (4) wartością parametru a jest dolny zakres przynależności do zbioru rozwiązań, natomiast parametr b górnym zakresem.

$$\mu(x; a, b) = \begin{cases} 0, & \text{dla } x \leq a \\ \frac{x - a}{b - a}, & \text{dla } a < x \leq b \\ 1, & \text{dla } x > b \end{cases} \quad (4)$$

- Funkcja trapezoidalna, która dana jest wzorem (5), gdzie wartości parametru a i d określają granice zbioru, dla których wartość funkcji jest różna od 0,

natomiast c i d są wartościami, dla których wartości znajdujące się w zbiorze pomiędzy nimi mają stopień przynależności do zbioru równy 1.

$$\mu(x; a, b, c, d) = \begin{cases} 0, & \text{dla } x \leq a \text{ lub } x > d \\ \frac{x-a}{b-a}, & \text{dla } a \leq x < b \\ 1, & \text{dla } b \leq x < c \\ \frac{d-x}{d-c}, & \text{dla } c \leq x < d \end{cases} \quad (5)$$

- Funkcja gaussowska (wzór 6.) będąca funkcją eksponencjalną, w której wykładniku wartość parametru a jest środkiem krzywej, a parametr b określa szerokość krzywej.

$$\mu(x; a, b) = \exp\left(-\left(\frac{x-a}{b}\right)^2\right) \quad (6)$$

3.2.2. Wnioskowanie

Kolejnym etapem po rozmyciu danych jest wnioskowanie [7]. W oparciu o zbiór reguł przypisuje się ostateczną wartość stopnia przynależności zestawu danych do tego zbioru. Reguły wnioskowania składają się z zestawu funkcji warunkowych, które określają jakie wartości są pożądane, a jakie nie.

W przypadku sumy logicznej zbiorów rozmytych wykorzystuje się operatory s– normy np.:

- maksimum: $\mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x))$
- suma algebraiczna: $\mu_{A \cup B}(x) = \mu_A(x) + \mu_B(x) - \mu_A(x) * \mu_B(x)$
- suma Hamachera: $\mu_{A \cup B}(x) = \frac{\mu_A(x) + \mu_B(x) - 2 * \mu_A(x) * \mu_B(x)}{1 - \mu_A(x) * \mu_B(x)}$

Natomiast dla iloczynu logicznego zbiorów rozmytych korzysta się z operatorów t-normy np.:

- minimum: $\mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x))$

- iloczyn: $\mu_{A \cap B}(x) = \mu_A(x) * \mu_B(x)$
- iloczyn Hamachera: $\mu_{A \cap B}(x) = \frac{\mu_A(x) * \mu_B(x)}{\mu_A(x) + \mu_B(x) - \mu_A(x) * \mu_B(x)}$

3.2.3. Wyostrzanie

Po przypisaniu wartości stopnia przynależności do zbioru reguł, dane rozmyte trafiają do ostatniego etapu, gdzie następuje ich wyostrzanie [7]. Polega ono na przekształceniu zbioru rozmytego na konkretną wartość wynikową, która ma zostać skierowana na wyjście. W celu jej otrzymania należy skorzystać z metody defuzyfikacji. Zaliczamy do nich: metodę pierwszego maksimum, metodą środka maksimum, metodę ostatniego maksimum i metodą środka ciężkości.

W przypadku skorzystania z metody środka ciężkości, należy skorzystać ze wzoru (7), w którym y jest wartością wyostrzoną, natomiast $\mu_y(x)$ to funkcja przynależności do zbioru.

$$y = \frac{\int x * \mu_y(x) dx}{\int \mu_y(x) dx} \quad (7)$$

3.3. Sztuczne sieci neuronowe

Dość popularną metodą tworzenia sztucznej inteligencji są sztuczne sieci neuronowe. Zostały zaprojektowane w oparciu o działanie układu nerwowego w ludzkim mózgu i polegają na pseudorównoległym przetwarzaniu danych [9]. Jest to metoda, w której wykazuje się tendencje do uogólniania wiedzy w przypadku otrzymania nowego zestawu danych. Sztuczna inteligencja w przypadku tej metody polega na uczeniu się w oparciu o zestaw treningowych zachowań.

Metoda ta wykorzystywana jest m.in. w systemach rozpoznawania mowy ludzkiej, pisma, rozróżniania kolorów, kształtów i gestów. Może mieć zastosowanie w medycynie, gdzie będzie pomocna w diagnostyce chorób w oparciu o zestaw próbek pobranych od pacjenta. W kryminalistyce, gdzie rozpoznawane będą linie papilarne przestępców. Ponieważ jest to jedna z najpopularniejszych metod

tworzenia sztucznej inteligencji znajduje swoje zastosowanie jeszcze w wielu innych dziedzinach nauki np. w fizyce, ekonomii, zarządzaniu.

Sztuczne sieci neuronowe znajdują również zastosowanie przy tworzeniu sztucznej inteligencji w grach komputerowych [3], w których sztuczna inteligencja w oparciu o poczynania gracza w grze uczy się jego taktyki i odpowiada poprzez naukę za dostosowanie poziomu trudności gry do jego umiejętności. Metoda ta może zostać wykorzystywana w grach wyścigowych, w których sztuczna inteligencja uczy się stylu jazdy gracza i w zależności od jego poczynania i umiejętności będzie uczyła się jego zachowania na torze.

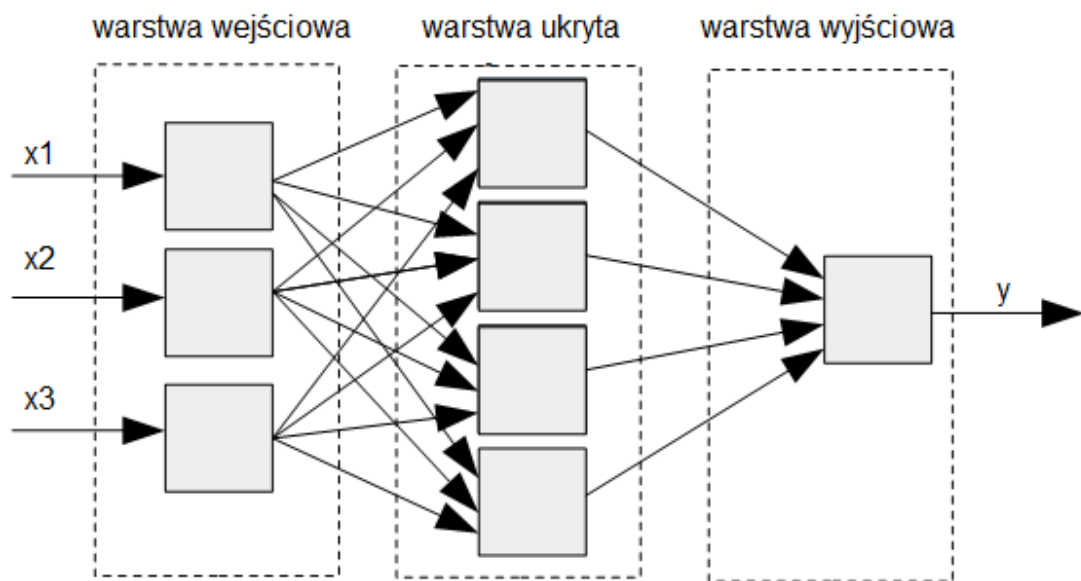
Zaletami zastosowania tej metody jest łatwość uczenia się w przypadku zastosowania algorytmu wstecznej propagacji błędów. Można zaimplementować system przewidujący zachowanie się gracza. Możliwość połączenia tej metody z metodą np. logiki rozmytej. Wadami może być w bardziej rozbudowanych przypadkach potrzeba dużej mocy obliczeniowej. Mogą też być mało skuteczne dla niektórych przypadków.

3.3.1. Budowa sztucznej sieci neuronowej

W oparciu o naturalne sieci neuronowe, opracowano matematyczny opis ich działania. Skupiono się głównie na sposobie przekazywania danych i ich przetwarzania. Przykładowa sieć neuronowa ukazana na rysunku 4. składa się z trzech warstw: wejściowej, ukrytej i wyjściowej. Warstwa wejściowa jest odpowiedzialna za przyjęcie danych. Warstwa ukryta ma za zadanie przetwarzać informacje. Wyjściowa służy do przekazania wyniku działania sieci neuronowej na wyjście. Każdy z węzłów w tej sieci jest odrębnym neuronem, natomiast strzałki są oznaczeniem połączenia między neuronami znajdującymi się w poszczególnych warstwach. Wejściom każdej komórki można przypisać odpowiednie wagi, które w zależności od tego, czy są wartościami dodatnimi czy ujemnymi, mogą działać pobudzająco lub hamująco na neurony. Iloczyn wag i sygnału dostarczanego na wejście jest nazywany funkcją aktywacji i jest wyrażony wzorem (8), gdzie w_i jest wektorem wag neuronu, a x_i jest wektorem sygnału wejściowego.

$$f(u(x)) = f\left(\sum_{i=1}^n w_i x_i\right) = f(w^T x) \quad (8)$$

W zależności od wartości funkcji aktywacji neuronu i jeśli przekracza ona wartość aktywacji jest przekazywana na wyjściu do kolejnego neuronu.



Rysunek 4. Schemat sztucznej sieci neuronowej

3.3.2. Metody uczenia sztucznej sieci neuronowej

Uczenie sieci neuronowej polega na dostarczeniu jej danych treningowych, które wskażą jak reagować na nowe przypadki po zakończeniu tego procesu. W danych przeznaczonych do uczenia sieci, zamieszczone są przykładowe dane, które mogą wystąpić na wejściu i dane wyjściowe, czyli rezultat jaki powinno się osiągnąć dla podanego zestawu. Podczas budowy uczenia tworzy się wagi na wejściu każdego neuronu i określa się błąd uczenia i tolerancję, czyli wartość stopu, poniżej której uznaje się, że sieć została nauczona.

Metoda wstecznej propagacji błędów

Jednym z popularnych algorytmów uczenia sztucznej sieci neuronowej jest metoda wstecznej propagacji błędów [8].

Inicjalizacja uczenia się sieci neuronowej w tym algorytmie rozpoczyna się od wprowadzenia na wejście wektora danych służących do nauki. Taki wektor składa się z wektora wejściowego, który zawiera oczekiwane dane na wejście i wektor wyjściowy przechowujący rezultat, jaki ma zostać osiągnięty na wyjściu po przeprowadzeniu wnioskowania. Następnie przetwarzane są dane z wektora wejściowego przez kolejne warstwy, gdzie obliczane są wartości aktywacji, aż do warstwy wyjściowej. Wyniki otrzymane na wyjściu porównuje się z wartościami oczekiwanymi podanymi w wektorze wyjściowym i oblicza się wartość błędów. W oparciu o wartość błędu następuje korekcja wag w poprzednich warstwach. Proces poprawy wag jest wykonywany w pętli do momentu, gdy wartość błędu spadnie poniżej wartości stwierdzającej, że sieć została nauczona.

3.4. Sieci bayesowskie

Metoda wykorzystująca sieci bayesowskie do budowy systemu wspomagania decyzji może wydawać się podobna do metody związanej z wykorzystaniem logiki rozmytej, tylko w przeciwieństwie do niej nie sprawdza stopnia przynależności do zbioru, lecz prawdopodobieństwo zajścia danego zdarzenia. Bazuje na wiedzy z zakresu teorii decyzji i teorii prawdopodobieństwa. Sieć bayesowska jest grafem skierowanym nie posiadającym cykli, w których w węzłach znajdują się zmienne losowe, a krawędzie łączące poszczególne węzły przedstawiają ciąg przyczynowo-skutkowy. Sprawdza się dobrze w przypadku systemu, który musi podjąć decyzję w warunkach niepewności. Metoda ta znajduje zastosowanie w medycynie, statystyce, ekonomii i genetyce. Zaletami tej metody są m.in. czytelna reprezentacja danych o charakterze przyczynowo-skutkowym i efektywny algorytm wnioskowania. Wadami może być konieczność znania rozkładu prawdopodobieństw wystąpienia zdarzenia w analizowanym problemie.

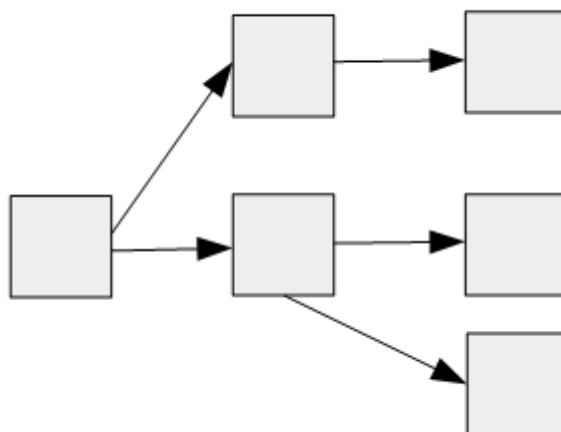
Konstruowanie sieci bayesowskiej

Konstrukcja sieci bayesowskiej [11] opiera się na budowie grafu, w którym zdarzenia są opisane jako ciąg przyczynowo-skutkowy. Przez taką konstrukcję powstaje zależność pomiędzy poszczególnymi węzłami, więc wystąpienie jednego zdarzenia ma wpływ na wystąpienie innego zdarzenia potomnego w sieci. Każdy węzeł tego grafu zawiera nazwę atrybutu i tablicę zależności warunkowych pomiędzy nim a jego rodzicami [10].

Zależność warunkowa jest dana wzorem (9), gdzie $P(A|B)$ jest prawdopodobieństwem zajścia zdarzenia A, jeśli zaszło zdarzenie B.

$$P(A|B) = \frac{P(A \cap B)}{P(B)} = \frac{P(A)P(B|A)}{P(B)} \quad (9)$$

Zanim jednak rozpocznie się budowę takiej sieci, istotne jest odpowiednie zdefiniowanie i uporządkowanie wartości oraz określenie prawdopodobieństwa warunkowego [11] i wstawienie ich do sieci. Wstawianie kolejnych węzłów do sieci odbywa się do momentu, gdy na liście znajdują się jeszcze niewstawione wartości. Przykładowa sieć bayesowska została przedstawiona na rysunku 5.



Rysunek 5. Przykładowa budowa sieci bayesowskiej

Klasyfikowanie nowego przypadku w sieci

W momencie gdy istnieje potrzeba sklasyfikowania konkretnego przypadku w sieci bayesowskiej konieczne jest wyliczenie iloczynu bezwarunkowego [12] prawdopodobieństwa przynależności przypadku do klasy z prawdopodobieństwem warunkowym. Wartość jest sprawdzana dla każdej klasy i każdego atrybutu dla wartości analizowanego przypadku. Klasą do której należy analizowany przypadek jest ta której iloczyn jest największy.

4. Środowisko Badawcze: gra strategiczna

4.1. Przygotowanie środowiska badawczego

W ramach środowiska badawczego związanego z realizacją tematu pracy dyplomowej została stworzona aplikacja, w której zaimplementowano algorytmy odpowiedzialne za rozwiązywanie problemu decyzyjnego.

Aplikacja jest strategiczną grą komputerową, w której celem jest wystawienie takich jednostek militarnych, które pokonają wojska wystawione przez sztuczną inteligencję. Podczas trwania rozgrywki gracz musi rozbudowywać swoje wojska, czyli kupować jednostki za dostępne w grze pieniądze. Po upływie czasu powinien zdecydować jakie z zakupionych jednostek wystawi, a następnie po zatwierdzeniu wyboru, zobaczyć rezultat bitwy.

Oprócz funkcji odpowiedzialnych za rozgrywkę, dostępny jest również podgląd z działania systemu wspomagania decyzji, który w zależności od wybranego algorytmu wyświetla odpowiednią strukturę danych.

4.1.1. Sformułowanie problemu decyzyjnego

W trakcie tworzenia środowiska badawczego, należało sformułować problem decyzyjny, którego analizą miał zająć się system wspomagania. W związku z tym w grze strategicznej sztuczna inteligencja ma za zadanie wybrać odpowiednie rodzaje i ilość jednostek, aby pokonać wojska gracza. Analiza danych historycznych jest oparta na danych zapisanych w katalogach w zależności od wybranego czasu gry. Poddawane analizie są rodzaje wojsk wystawionych w poprzednich rozgrywkach, ilość jednostek wystawionych na pole bitwy, ilość jednostek ocalałych siła wojsk oraz rezultat bitwy. Podczas ładowania danych do analizy jest brany pod uwagę rezultat, ponieważ brana jest pod uwagę tylko strategia wygrywająca, więc sztuczna inteligencja uczy się strategii wygrywania zarówno w oparciu o własne sukcesy, jak i sukcesy gracza. Wczytane dane są następnie przekazywane w zależności od wyboru gracza odpowiednim algorytmom, które dokonują klasyfikacji każdej

Denis Wychowalek - Podejmowanie decyzji w grach komputerowych w oparciu o analizę danych historycznych

strategii wygrywającej i wybiera tą, która jest najlepsza dla danych podanych na wejście przez gracza.

4.1.2. Wymagania funkcjonalne i нефunkcjonalne Środowiska Badawczego

Wymagania funkcjonalne

- Gracz powinien mieć możliwość wyboru algorytmu wspomaganie decyzji i długości trwania rozgrywki.
- Aplikacja umożliwia zakup wybranego rodzaju i ilości jednostek ze sklepu przez określony czas.
- Możliwy jest wybór przez grającego jednostek, które zostaną wysłane do walki ze sztuczną inteligencją.
- Przygotowanie możliwości odczytania historycznych danych z plików.
- Implementacja wybranych algorytmów odpowiedzialnych za działanie systemu wspomaganie decyzji.
- Dostępny jest system walki, w którym po wybraniu przez strony konfliktu jednostek do bitwy nawzajem są eliminowane.
- Po zakończeniu bitwy wyświetlane są wyniki bitwy oraz rezultat działania systemu. Wyświetlenie jego rodzaju, struktury oraz czasu wykonania.
- W przypadku wybrania drzewa decyzyjnego wyświetlona jest struktura drzewa wraz z nazwami i wartościami atrybutów, które dzielą zbiór rozwiązań.
- Dla logiki rozmytej wyświetlana jest tabela z podzielonymi danymi na atrybuty wraz z wyliczonymi wartościami związanymi z przynależnością do zbioru rozwiązań.
- Wynik bitwy zapisywany jest do odpowiedniego katalogu w zależności od okresu czasu wybranego przez gracza.

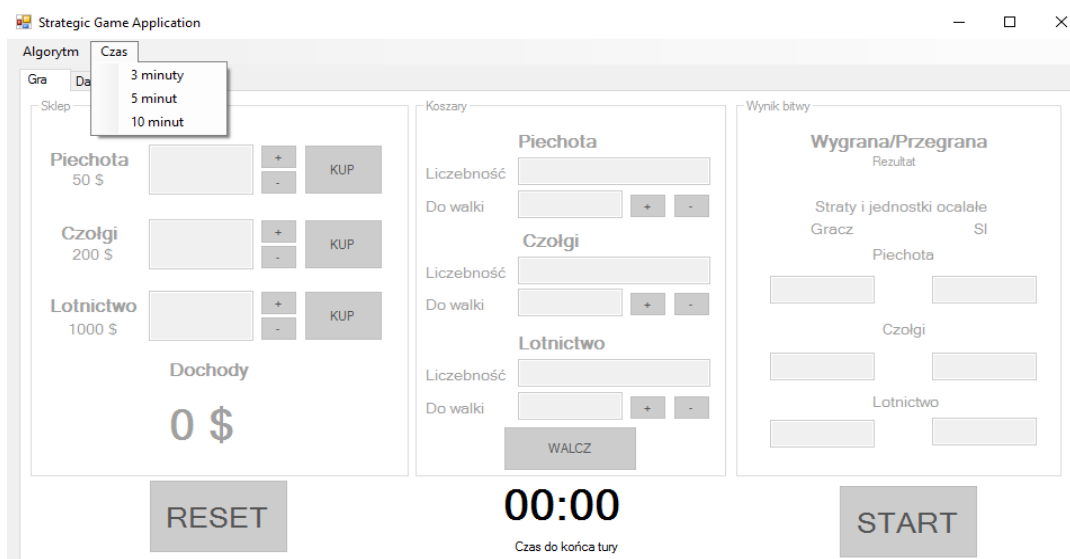
Wymagania niefunkcjonalne

- Grę można uruchomić na wszystkich wersjach systemu Windows.
- Aplikację można uruchomić na systemach 32 i 64 bitowych.

4.2. Funkcje dostępne w aplikacji

W aplikacji zostało oddanych do użytku kilka typów funkcjonalności, które są odpowiedzialne za poszczególne elementy aplikacji takie jak: ustawienia czasu przez jaki możliwe jest kupowanie jednostek w sklepie i wybór używanego algorytmu, funkcje odpowiedzialne za możliwość kupowania jednostek w sklepie, wybór i akceptację wystawionych jednostek przez gracza do bitwy, wyświetlanie rezultatu walki i danych z przeprowadzonego procesu decyzyjnego.

Funkcjonalność związana z ustawieniami

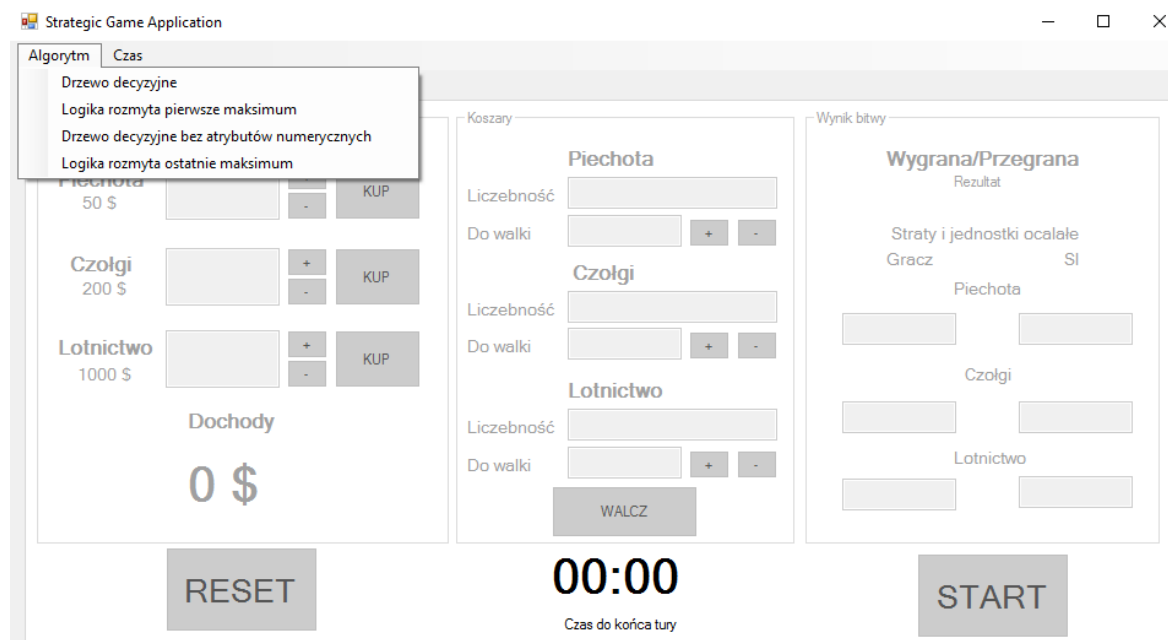


Rysunek 6. Ustawienie czasu trwania rozgrywki

W grze do dyspozycji gracza zostały oddane funkcje, w których może ustawić parametry gry. Pierwszym parametrem przedstawionym na rysunku 6. jest ustawienie czasu. Wybranie tego parametru jest istotne dla określenia długości

Denis Wychowalek - Podejmowanie decyzji w grach komputerowych w oparciu o analizę danych historycznych

trwania rozgrywki, jak i do określenia katalogu, w którym znajdują się dane potrzebne do analizy.



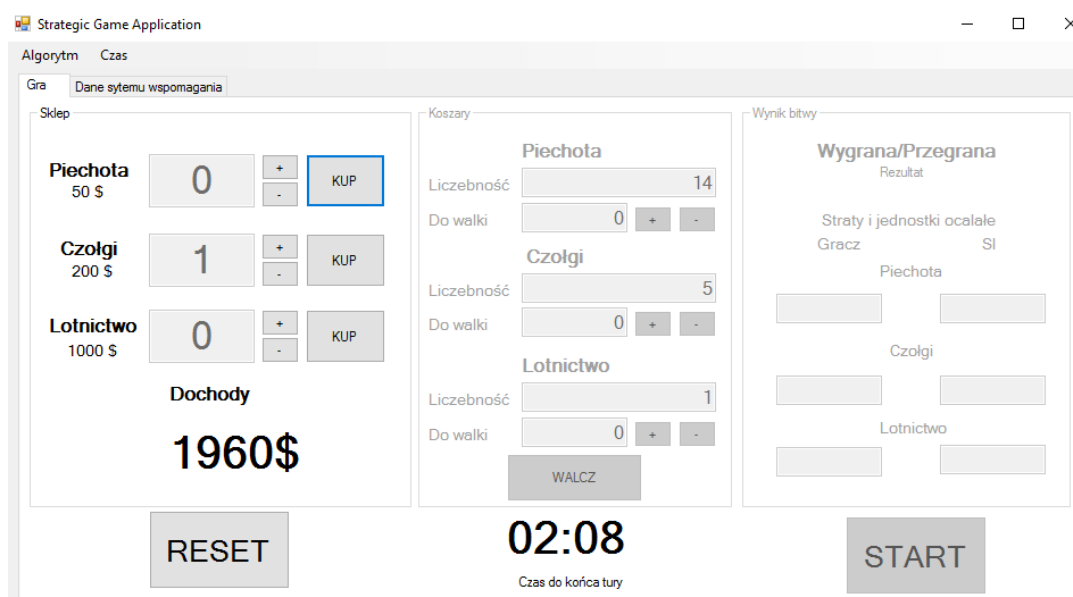
Rysunek 7. Ustawienie algorytmu podejmowania decyzji

Drugim parametrem na rysunku 7. do ustawienia jest metoda wspomaganie decyzji. Poprzez wybór jednej z czterech dostępnej opcji, po rozpoczęciu walki wybierany jest jeden z dwóch algorytmów, które dokonają analizy danych i podejmą decyzję. Wybór decyduje też o rodzaju danych wyświetlanych w karcie Dane systemu wspomaganie po zakończeniu rozgrywki.

Funkcjonalność związana z rozgrywką

Po rozpoczęciu rozgrywki przez wcześniej wybrany czas jest dostępny sklep, w którym gracz może wybierać liczbę i rodzaj jednostek do zakupu (rysunek 8). W przypadku nie wystarczającej kwoty posiadanych pieniędzy, nie powiększa się ich zasób w Koszarach. Po upływie czasu dostęp do Sklepu jest blokowany i w udostępnionym do edycji dziale Koszary można wybrać jednostki, które mają być wysłane na pole walki. W dziale wyniki bitwy jest wyświetlany rezultat walki po wysłaniu jednostek.

Denis Wychowalek - Podejmowanie decyzji w grach komputerowych w oparciu o analizę danych historycznych



Rysunek 8. Wyświetlane informacje w trakcie rozbudowy wojska

W zakładce Dane systemu wspomagania zaprezentowanym na rysunku 9. są wyświetlane informacje związane z podejmowaniem decyzji, gdzie wyświetlana jest rodzaj wybranej metody, struktura danych i czas wykonania dla danej metody.

Strategic Game Application

AlgotrytmCzas

GraDane systemu wspomagania

Uzyty algorytm: Logika rozmyta

| | SoldierCount | TankCount | AircraftCount | FireSummary | SoldierSurvive | TankSurvive | AircraftSurvive | DecisionClass | | | | | | | |
|----|--------------|-----------|---------------|-------------|----------------|-------------|-----------------|---------------|-------|----|-------|---|-------|---------------------|-------|
| 18 | 1,000 | 0 | 0,000 | 0 | 0,000 | 36 | 0,655 | 4 | 1,000 | 0 | 0,000 | 0 | 0,000 | Soldier | 0,655 |
| 5 | 1,000 | 4 | 0,800 | 0 | 0,000 | 66 | 1,000 | 0 | 1,000 | 2 | 0,167 | 1 | 0,250 | SoldierTankAircraft | 0,250 |
| 5 | 1,000 | 0 | 0,000 | 0 | 0,000 | 46 | 0,836 | 3 | 0,750 | 1 | 0,083 | 1 | 0,250 | SoldierTankAircraft | 0,750 |
| 5 | 1,000 | 0 | 0,000 | 4 | 1,000 | 126 | 1,000 | 0 | 0,000 | 0 | 0,000 | 3 | 0,750 | SoldierTankAircraft | 0,750 |
| 5 | 1,000 | 0 | 0,000 | 5 | 1,000 | 146 | 1,000 | 0 | 0,000 | 0 | 0,000 | 3 | 0,750 | SoldierTankAircraft | 0,750 |
| 5 | 1,000 | 7 | 1,000 | 5 | 1,000 | 135 | 1,000 | 0 | 0,000 | 0 | 0,000 | 2 | 0,500 | SoldierTankAircraft | 0,500 |
| 5 | 1,000 | 7 | 1,000 | 0 | 0,000 | 35 | 0,636 | 0 | 0,000 | 3 | 0,250 | 0 | 0,000 | SoldierTank | 0,250 |
| 5 | 1,000 | 9 | 1,000 | 0 | 0,000 | 45 | 0,818 | 0 | 0,000 | 2 | 0,167 | 0 | 0,000 | SoldierTank | 0,167 |
| 5 | 1,000 | 11 | 1,000 | 0 | 0,000 | 55 | 1,000 | 0 | 0,000 | 2 | 0,167 | 0 | 0,000 | SoldierTank | 0,167 |
| 1 | 0,000 | 1 | 0,200 | 0 | 0,000 | 7 | 0,127 | 0 | 0,000 | 0 | 0,000 | 2 | 0,500 | Aircraft | 0,127 |
| 5 | 1,000 | 0 | 0,000 | 2 | 1,000 | 40 | 0,727 | 0 | 0,000 | 0 | 0,000 | 2 | 0,500 | Aircraft | 0,500 |
| 23 | 1,000 | 0 | 0,000 | 0 | 0,000 | 46 | 0,836 | 1 | 0,250 | 2 | 0,167 | 0 | 0,000 | SoldierTank | 0,250 |
| 5 | 1,000 | 0 | 0,000 | 3 | 1,000 | 60 | 1,000 | 0 | 0,000 | 0 | 0,000 | 2 | 0,500 | SoldierTankAircraft | 0,500 |
| 5 | 1,000 | 0 | 0,000 | 1 | 1,000 | 20 | 0,364 | 0 | 0,000 | 0 | 0,000 | 2 | 0,500 | SoldierTankAircraft | 0,364 |
| 5 | 1,000 | 0 | 0,000 | 5 | 1,000 | 100 | 1,000 | 0 | 0,000 | 0 | 0,000 | 2 | 0,500 | SoldierTankAircraft | 0,500 |
| 5 | 1,000 | 5 | 1,000 | 3 | 1,000 | 85 | 1,000 | 0 | 0,000 | 0 | 0,000 | 2 | 0,500 | SoldierTankAircraft | 0,500 |
| 5 | 1,000 | 10 | 1,000 | 3 | 1,000 | 110 | 1,000 | 0 | 0,000 | 0 | 0,000 | 2 | 0,500 | SoldierTankAircraft | 0,500 |
| 5 | 1,000 | 10 | 1,000 | 5 | 1,000 | 150 | 1,000 | 0 | 0,000 | 0 | 0,000 | 2 | 0,500 | SoldierTankAircraft | 0,500 |
| 5 | 1,000 | 25 | 1,000 | 5 | 1,000 | 225 | 1,000 | 0 | 0,000 | 0 | 0,000 | 3 | 0,750 | SoldierTankAircraft | 0,750 |
| 5 | 1,000 | 25 | 1,000 | 5 | 1,000 | 259 | 1,000 | 0 | 0,000 | 0 | 0,000 | 2 | 0,500 | SoldierTankAircraft | 0,500 |
| 5 | 1,000 | 25 | 1,000 | 5 | 1,000 | 281 | 1,000 | 0 | 0,000 | 0 | 0,000 | 2 | 0,500 | SoldierTankAircraft | 0,500 |
| 5 | 1,000 | 25 | 1,000 | 5 | 1,000 | 311 | 1,000 | 0 | 0,000 | 0 | 0,000 | 3 | 0,750 | SoldierTankAircraft | 0,750 |
| 5 | 1,000 | 2 | 0,400 | 0 | 0,000 | 56 | 1,000 | 1 | 0,250 | 3 | 0,250 | 0 | 0,000 | SoldierTank | 0,250 |
| 5 | 1,000 | 25 | 1,000 | 5 | 1,000 | 327 | 1,000 | 0 | 0,000 | 0 | 0,000 | 2 | 0,500 | SoldierTankAircraft | 0,500 |
| 5 | 1,000 | 30 | 1,000 | 5 | 1,000 | 352 | 1,000 | 0 | 0,000 | 0 | 0,000 | 4 | 1,000 | SoldierTankAircraft | 1,000 |
| 5 | 1,000 | 41 | 1,000 | 5 | 1,000 | 407 | 1,000 | 0 | 0,000 | 0 | 0,000 | 4 | 1,000 | SoldierTankAircraft | 1,000 |
| 5 | 1,000 | 41 | 1,000 | 0 | 0,000 | 307 | 1,000 | 0 | 0,000 | 12 | 1,000 | 1 | 0,250 | SoldierTankAircraft | 1,000 |
| 5 | 1,000 | 24 | 1,000 | 0 | 0,000 | 120 | 1,000 | 0 | 0,000 | 3 | 0,250 | 1 | 0,250 | SoldierTankAircraft | 0,250 |
| 5 | 1,000 | 10 | 1,000 | 0 | 0,000 | 50 | 0,909 | 0 | 0,000 | 2 | 0,167 | 0 | 0,000 | SoldierTank | 0,167 |
| 5 | 1,000 | 6 | 1,000 | 0 | 0,000 | 30 | 0,545 | 0 | 0,000 | 3 | 0,250 | 1 | 0,250 | SoldierTankAircraft | 0,250 |
| 5 | 1,000 | 12 | 1,000 | 0 | 0,000 | 60 | 1,000 | 0 | 0,000 | 2 | 0,167 | 1 | 0,250 | SoldierTankAircraft | 0,250 |

Rysunek 9. Zakładka danych systemu wspomagania decyzji

4.3. Obsługa

Po uruchomieniu aplikacji użytkownik przenoszony jest do okna głównego, które zostało zaprezentowane na rysunku 8. W aplikacji znajduje się menu odpowiedzialne za ustawienie opcji oraz dwie zakładki. W zakładce Gra znajdują się trzy działy odpowiedzialne za rozgrywkę, a w zakładce Dane systemu wspomagania są wyświetlane informacje o wynikach systemu podejmowania decyzji.

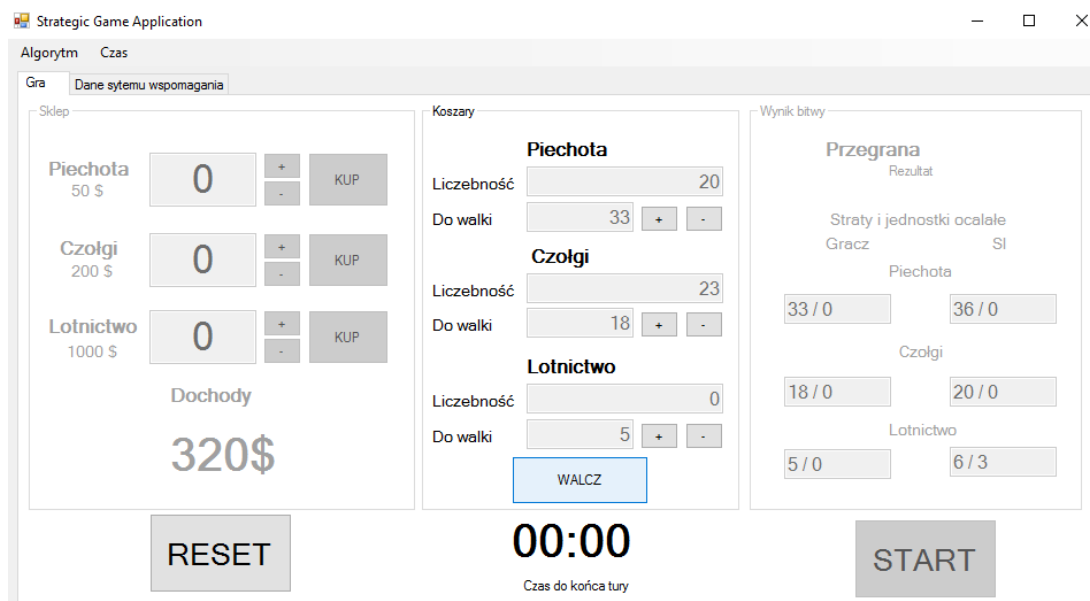
Pierwszym krokiem, przed rozpoczęciem rozgrywki jest wybranie jednego z trzech dostępnych czasów rozgrywki i wybranie jednego z dwóch dostępnych algorytmów, by aktywował się przycisk START. Przycisk ten jest odpowiedzialny za rozpoczęcie odliczania czasu i aktywowanie działu odpowiedzialnego za Sklep.

Po rozpoczęciu rozgrywki użytkownik wybiera poprzez przyciski + i – liczbę jednostek, które chce kupić. Zakup jest dokonywany poprzez naciśnięcie przycisku KUP i w momencie gdy posiada odpowiednią ilość środków finansowych, których stan jest wyświetlony poniżej napisu Dochody. W grze dostępne są trzy możliwe jednostki: piechota, czołgi i samoloty.

Po upływie czasu następuje blokada działu odpowiedzialnego za sklep oraz odblokowuje się dział o nazwie Koszary. W tym dziale gracz wybiera poprzez naciśnięcie na przyciski + i – ilość jednostek do walki. Wybór jednostek jest możliwy do momentu naciśnięcia przycisku WALCZ, który akceptuje wojska do walki i inicjuje system podejmowania decyzji oraz rozpoczyna walkę między jednostkami gracza a jednostkami sztucznej inteligencji.

W momencie zakończenia walki wyświetla się rezultat gry w dziale Wynik bitwy, gdzie wyświetlana jest informacja kto wygrał pojedynek i wyświetlany jest bilans jednostek wystawionych i straconych (rysunek 10). Wynik gry jest zapisywany do pliku w trakcie wyświetlenia informacji. Po zakończeniu bitwy wyświetlane są również wyniki analizy systemu wspomagania decyzji (rysunek 9.) w zakładce Dane systemu wspomagania.

Denis Wychowalek - Podejmowanie decyzji w grach komputerowych w oparciu o analizę danych historycznych



Rysunek 10. Wyświetlenie wyniku bitwy

W trakcie działania aplikacji udostępniony jest również przycisk RESET, który przerywa działanie gry i usuwa wszystkie ustawione dane w aplikacji.

4.4. Specyfikacja wewnętrzna

4.4.1. Organizacja danych

W środowisku badawczym istnieje konieczność wczytania dużej liczby danych. Wszystkie dane historyczne są zapisane w plikach tekstowych, które znajdują się w katalogach o nazwach odpowiednich ustawień czasowych. Każdy plik zawiera dane, które są zapisywane po zakończeniu rozgrywki. Nazwa pliku zawiera nazwę oraz numer rozgrywki.

W pierwszej kolejności zapisywana jest nazwa strony, która wygrała starcie a następnie zapisywane są dane o stanie wojsk gracza, gdzie podane są: liczba wystawionych żołnierzy, czołgów i samolotów, liczba ocalałych jednostek z podziałem na ich rodzaj oraz sumaryczna siła ognia wojska. W pliku zapisywane są też te same dane dla rezultatu osiągniętego przez sztuczną inteligencję.

Wszystkie dane przy odczycie z pliku są odpowiednio przygotowywane do analizy. W zależności od strony, która wygrała są pobierane w pierwszej kolejności dane przegranego, które przypisywane są do tymczasowej listy wartości atrybutów, a dane wygranego związane z wystawionymi jednostkami są zamieniane na klasę decyzyjną. Na etapie analizy danych wszystkie dane są przydzielane z tymczasowych list do atrybutów, które zawierają nazwę atrybutu, listę wszystkich wartości atrybutu oraz w zależności od tego czy wybrano drzewa decyzyjne, czy logikę rozmytą – wartości specyficzne dla danej metody.

4.4.2. Klasy

Wszystkie napisane klasy związane ze środowiskiem badawczym zostały podzielone na cztery odrębne projekty, które są odpowiedzialne za poszczególne elementy aplikacji czyli: za obsługę graficznego interfejsu użytkownika, elementów gry strategicznej, drzewa decyzyjnego i logiki rozmytej.

Klasa związana z obsługą graficznego interfejsu użytkownika

W projekcie StrategicGame znajduje się klasa, w której znajdują się funkcje mające za zadanie zapewnić obsługę wszystkich udostępnionych funkcjonalności. Jest odpowiedzialna za wyświetlanie wszystkich potrzebnych informacji na ekran związanych z elementami rozgrywki, wynikiem gry oraz wynikiem działania systemu decyzyjnego. W projekcie tym wywoływane są również metody odpowiedzialne za podejmowanie decyzji oraz za dobór jednostek przed rozpoczęciem walki. Jest tutaj także inicjowana walka.

Klasy związane z grą strategiczną

W projekcie GameLogic znajdują się klasy odpowiedzialne za funkcjonalność związaną z grą strategiczną. Zaimplementowane zostały klasy odpowiedzialne za funkcjonowanie sklepu i koszar. Znajduje się także klasa związana z generowaniem środków finansowych do zakupu jednostek oraz system walki inicjowany po zgłoszeniu jednostek do bitwy przez gracza i sztuczną inteligencję.

Klasy związane z drzewem decyzyjnymi

Projekt DecisionTree zawiera klasy, które zawierają metody odpowiedzialne za wczytywanie danych z katalogu, funkcje tworzące atrybuty, klasy decyzyjne oraz przydzielające poszczególne wartości do odpowiedniej struktury. Znajdują się funkcje obliczające potrzebne do działania algorytmu wartości. Zaimplementowana jest budowa struktury drzewa decyzyjnego oraz funkcja odpowiedzialna za podejmowanie decyzji.

Klasy związane z logiką rozmytą

W projekcie FuzzyLogic przygotowane zostały dwie klasy, w których zaimplementowano funkcje odpowiedzialne za elementy regulatora rozmytego, metody przygotowujące dane do analizy oraz obliczające przynależność do poszczególnych zbiorów. Znajduje się również system wnioskujący, który wybiera spośród zbioru wszystkich rozwiązań, te które uzna za optymalne.

4.5. Implementacja wybranych metod

4.5.1. Drzewa decyzyjne

Do rozwiązania problemu decyzyjnego zostały wykorzystane drzewa decyzyjne, których główną zaletą jest najbardziej czytelne i zrozumiałe dla każdego człowieka przedstawienie możliwych wariantów rozwiązania problemu. Kluczowym elementem poprawnego działania systemu wspomagania decyzji był odpowiedni dobór atrybutów, które najlepiej opisywałyby czynniki mające wpływ na wybór jednej z kilku dostępnych opcji.

W przypadku stworzonej aplikacji po przeprowadzeniu bitwy generowane są dane informujące o rezultacie bitwy, czyli o tym która strona wygrała bitwę. Kolejnymi informacjami jest stan wojsk przed i po rozegraniu bitwy dla każdej ze stron. Wszystkie wartości zapisywane do pliku są rodzaju numerycznego, co w przypadku doboru danych i poddawaniu ich analizie miało duży wpływ na rozmiar generowanego drzewa decyzyjnego.

Struktura decyzyjna w przypadku zastosowania tej metody jest drzewem binarnym. Do budowy drzewa decyzyjnego skorzystano z algorytmu C 4.5, który w oparciu o wartości atrybutu ustala reguły decyzyjne. Tworzy się je w oparciu o znajdowanie zależności pomiędzy wartościami poszczególnych atrybutów, dzięki czemu nie było konieczne wyznaczanie sztywnych reguł i można było ten proces zautomatyzować. W celu wspomagania procesu decyzyjnego zostały zaimplementowane funkcje obliczające entropię zbioru, zysk i współczynnik zysku informacji, które dzieliły zbiór rozwiązań do momentu, gdy wszystkie rozwiązania nie zostały uwzględnione lub w momencie gdy nie dało się już ich bardziej rozdzielić.

Problemem podczas implementacji okazał się dobór właściwego typu reprezentacji poszczególnych wartości. W przypadku użycia wyłącznie wartości numerycznych, prowadziło to do klasyfikowania jednostkowych wariantów decyzji i były one uzależnione w dużej mierze od danych podawanych na wejście i w łatwy sposób można było spowodować złe wnioskowanie systemu decyzyjnego.

Rozwiązaniem okazało się większe uogólnienie części z dostępnych danych oraz zawężenie liczby klas do sześciu możliwych wariantów, które mogą wystąpić podczas gry, czyli wyboru konkretnego rodzaju wojsk, a nie jak dotychczas odpowiedniej liczby jednostek. Zamianie typu danych z numerycznego na wartości logiczne, uległy dane odpowiedzialne za wystawione jednostki, natomiast pogrupowaniu wartości uległy dane związane z sumaryczną siłą ognia. Dzięki tej zamianie zamiast przeprowadzania testów nierównościowych, są przeprowadzane testy równościowe. Aby przedstawić różnice pomiędzy użyciem wartości numerycznych a ich grupowaniem przygotowano dwie różne funkcje uwzględniające ten element podczas budowy drzewa.

Ponieważ na wejście nie są otrzymywane informacje na temat ocalałych wojsk, gdyż podejmowana decyzja ma być podjęta przed rozpoczęciem rozgrywki, to należało ustalić kryterium dokonywania wyboru w przypadku, gdy podczas przechodzenia przez drzewo napotkany zostałby taki atrybut. W tym przypadku wzięta została pod uwagę wartość maksymalna z dostępnego zbioru wartości dla danego atrybutu.

Mimo znacznego ograniczenia nadmiernego rozrostu drzewa należało również stworzyć funkcję, która ma eliminować nieoptymalne rozwiązania, czyli zaimplementować metodę podcinania drzewa. W przypadku stworzonej struktury zastosowano wybór spośród możliwych klas decyzyjnych tej klasy, która posiada najwięcej spośród wszystkich dostępnych w zbiorze rodzajów jednostek.

4.5.2. Logika rozmyta

W przypadku logiki rozmytej czynnikiem decydującym o wyborze tej metody była prosta implementacja oraz duża dokładność wyników analizy poszczególnych wartości historycznych wczytywanych z pliku. Systemem decyzyjnym dla tej metody został regulator rozmyty, który składa się z trzech etapów: rozmywania, wnioskowania i wyostrzania.

Na początku ważny był odpowiedni podział wczytywanych wartości. Tak samo jak w przypadku metody drzewa decyzyjnego, zostały na początku wczytane wszystkie zapisane wartości numeryczne, lecz w przeciwieństwie do tej metody, nie zaistniała konieczność ich zamiany na wartości logiczne. Uogólnianie nie było tutaj konieczne, gdyż zadaniem metody logiki rozmytej jest klasyfikacja każdego przypadku w zależności od stopnia jego przynależności do zbioru dopuszczalnych wartości.

Pierwszym etapem, który należało zaimplementować była funkcja odpowiedzialna za rozmywanie wszystkich wartości. Dla podzielonych atrybutów należało wybrać odpowiednie zakresy zbiorów dopuszczalnych wartości, do których przypisywany byłby stopień przynależności. Dane otrzymywane na wejście były górną granicą zbioru rozwiązań, natomiast dolną granicą była najmniejsza wartość w dostępnym zbiorze. Dla danych nieotrzymywanych na wejście, czyli liczby ocalałych jednostek górną granicą stawała się wartość maksymalna zbioru, a dolną minimalna. Dzięki temu uzyskano wartości zmiennoprzecinkowe, które określały przynależność do zbioru, w przypadku klasyfikacji wszystkich wartości skorzystano z własności funkcji klasy γ , tzn. poniżej dolnej granicy zbioru wszystkie wartości przyjmowały wartości 0, natomiast powyżej górnej - przypisywano wartość 1.

Po uzyskaniu zbioru wartości rozmytych przekazano je do kolejnego etapu związanego z wnioskowaniem. W tej części istotne było określenie zależności pomiędzy poszczególnymi zbiorami rozmytymi atrybutów. Dla zbiorów związanych z wartościami jednostek wystawionych do bitwy zastosowano sumę zbiorów rozmytych, dla których ich wartość została określona poprzez zastosowanie z operatorów s-norm wartości maksymalnej. Tego samego operatora użyto dla określenia wartości wspólnej sumy zbiorów rozmytych jednostek ocalałych. Po określeniu tych wartości dla każdego przypadku wyznaczano iloczyn tych zbiorów razem ze zbiorem rozmytym wartości sumarycznej siły ognia, dla których określenia zastosowano z operatorów t-norm wybór wartości minimalnej.

Po dokonaniu wnioskowania wyliczone wartości zostały przekazane do ostatniego etapu związanego z wyostrzaniem danych, gdzie spośród wszystkich wartości musi zostać wybrana jedna. Dla danego zbioru danych została wybrana metoda pierwszego maksimum, które spośród wszystkich wartości wybiera tą o największej wartości. Na podstawie wybranego danego przypadku, wybierana jest konkretna wartość klasy decyzyjnej.

Podczas implementacji elementów logiki rozmytej podstawowym problemem było określenie relacji pomiędzy poszczególnymi zbiorami rozmytymi, gdyż w przypadku wybrania zamiast sumy zbiorów, iloczynu mogłoby doprowadzić do eliminacji części rozwiązań tylko ze względu na fakt, że nie występowały one dla danych podanych na wejściu. Zastosowane rozwiązanie doprowadziło do większego zróżnicowania wszystkich przypadków.

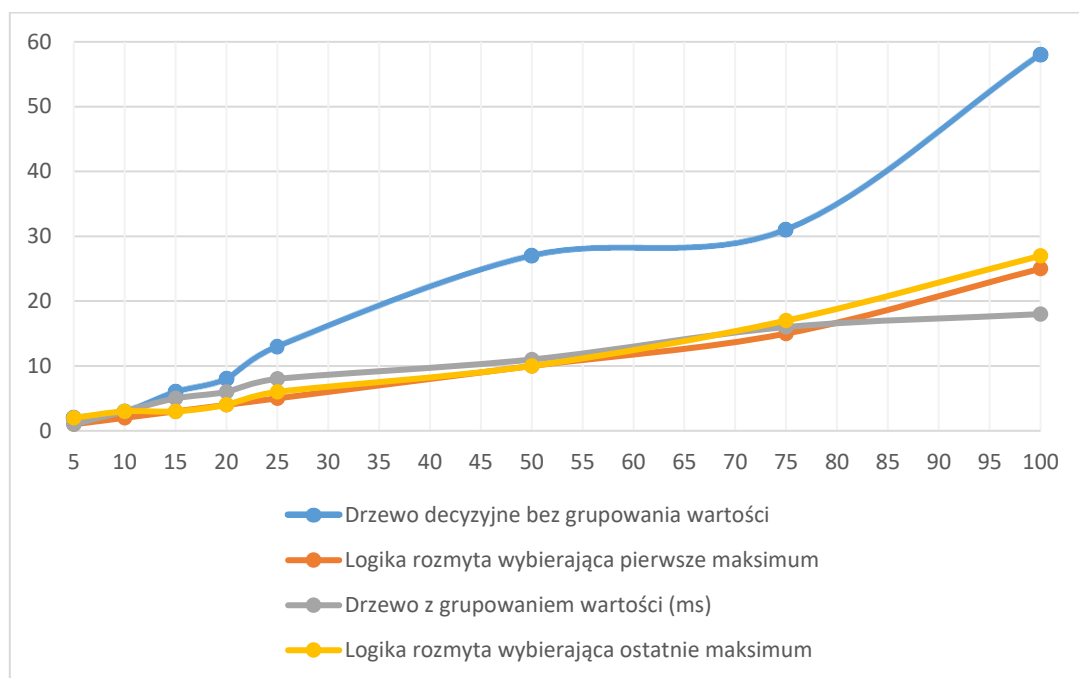
5. Testowanie i analiza wyników

5.1. Pomiar czasu wykonania algorytmów

Po zaimplementowaniu algorytmów związanych z systemem podejmowania decyzji, przystąpiono do przeprowadzenia testów, które polegały na przeprowadzeniu kilkudziesięciu rozgrywek. Dla takiego samego zestawu danych zapisanych i dla takich samych danych wejściowych przeprowadzono testy związane z czasem wykonania drzew decyzyjnych i różnych sposobów wnioskowania w logice rozmytej. Zastosowane rozwiązanie doprowadziło do większego zróżnicowania wszystkich przypadków. Wyniki tych testów zostały zestawione w poniższej tabeli i na wykresie przedstawionym na rysunku 11.

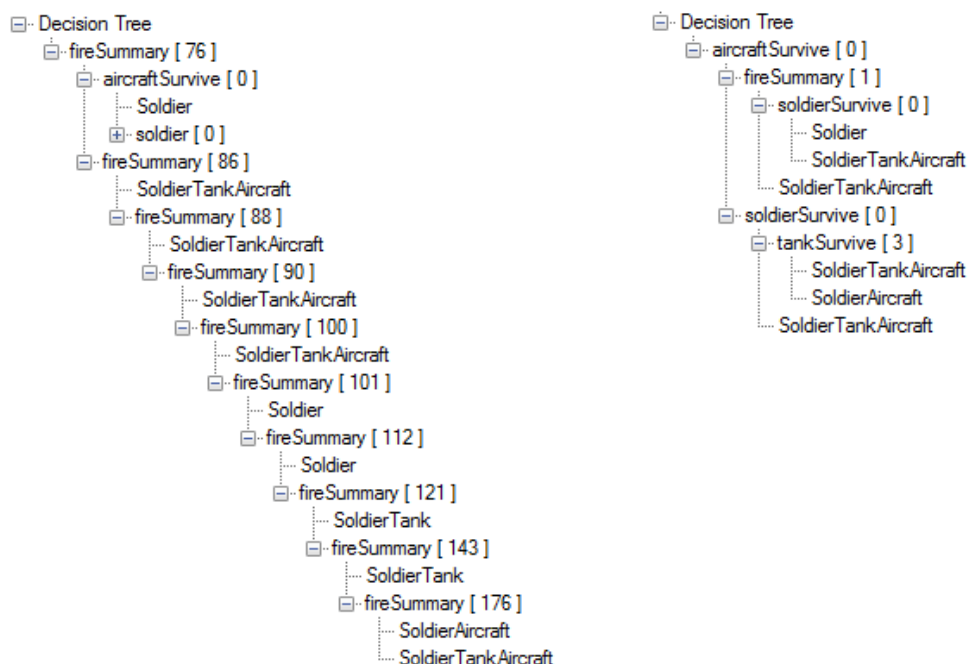
Tabela 1. Zestawienie wartości czasu wykonania algorytmów

| Ilość danych przekazanych do analizy | Drzewo decyzyjne bez grupowania wartości (ms) | Logika rozmyta Wybierająca pierwsze maksimum (ms) | Drzewo z grupowaniem wartości (ms) | Logika rozmyta wybierająca ostatnie maksimum (ms) |
|--------------------------------------|---|---|------------------------------------|---|
| 5 | 2 | 1 | 1 | 2 |
| 10 | 3 | 2 | 3 | 3 |
| 15 | 6 | 3 | 5 | 3 |
| 20 | 8 | 4 | 6 | 4 |
| 25 | 13 | 5 | 8 | 6 |
| 50 | 27 | 10 | 11 | 10 |
| 75 | 31 | 15 | 16 | 17 |
| 100 | 58 | 25 | 18 | 27 |



Rysunek 11. Wykres przedstawiający czas wykonania zaimplementowanych algorytmów w zależności od liczby wczytanych przykładów

Jak można zauważyć algorytm związany z drzewem decyzyjnym bez możliwości grupowania wartości związanych z sumaryczną siłą ognia osiągał najdłuższe czasy wykonania. O wiele lepsze okazało się zastosowanie elementów logiki rozmytej i wraz ze wzrostem ilości danych przekazywanych do analizy metody osiągały o ponad połowę mniejsze czasy niż pierwszy algorytm. Najlepiej podczas testu wypadł algorytm drzew decyzyjnych z możliwością grupowania, który do ilości 75 danych spisywał się nieco gorzej od algorytmu logiki rozmytej, lecz przy większej ilości danych osiągnął o wiele lepszy czas wykonania. Pomiedzy dwoma metodami wnioskowania w logice rozmytej uzyskano zbliżone do siebie czasy wykonania ze wskazaniem na korzyść dla metody wnioskowania - pierwsze maksimum.



Rysunek 12. Porównanie rozmiaru drzew decyzyjnych bez grupowania i z grupowaniem wartości sumarycznej siły ognia

W przypadku algorytmu drzew decyzyjnych kluczowe okazało się pogrupowanie wartości związanych z sumaryczną siłą ognia. Przy dużej liczbie danych dla algorytmu bez utworzenia grup, algorytm zaczął uwzględniać przypadki pojedyncze ze względu na wartości numeryczne tej jednej grupy, co przedstawia rysunek 12., na którym struktura bez pogrupowania wartości znajduje się po lewej stronie. Z możliwością grupowania udało się w znacznym stopniu uniknąć nadmiernego rozrostu drzewa, co też wykazano podczas testów czasu wykonania, gdzie różnica między tymi dwoma strukturami wraz ze wzrostem liczby danych do testów m.in. w czasie wykonania, jak i rozmiarze struktury. Na rysunku 12. przedstawiono strukturę w postaci umieszczenia w węzłach nazw atrybutów, które dzielą zbiór rozwiązań oraz wartości, które dzielą ten zbiór.

Denis Wychowalek - Podejmowanie decyzji w grach komputerowych w oparciu o analizę danych historycznych

| Użyty algorytm: Logika rozmyta | | | | | | | | | | | | | | | |
|--------------------------------|-----------|---------------|-------------|----------------|-------------|------------------|---------------|---|-------|---|-------|---|-------|---------------------|-------|
| SoldierCount | TankCount | AircraftCount | FireSummary | SoldierSurvive | TankSurvive | Aircraft Survive | DecisionClass | | | | | | | | |
| 38 | 0,740 | 0 | 0,000 | 0 | 0,000 | 76 | 0,185 | 1 | 0,167 | 0 | 0,000 | 0 | 0,000 | Soldier | 0,167 |
| 1 | 0,000 | 0 | 0,000 | 0 | 0,000 | 2 | 0,000 | 0 | 0,000 | 0 | 0,000 | 3 | 0,600 | Aircraft | 0,000 |
| 51 | 1,000 | 0 | 0,000 | 5 | 1,000 | 100 | 0,245 | 0 | 0,000 | 0 | 0,000 | 2 | 0,400 | SoldierTankAircraft | 0,245 |
| 9 | 0,160 | 25 | 0,568 | 0 | 0,000 | 143 | 0,353 | 0 | 0,000 | 0 | 0,000 | 5 | 1,000 | SoldierTankAircraft | 0,353 |
| 51 | 1,000 | 24 | 0,545 | 5 | 1,000 | 266 | 0,660 | 0 | 0,000 | 0 | 0,000 | 2 | 0,400 | SoldierTankAircraft | 0,400 |
| 51 | 1,000 | 26 | 0,591 | 0 | 0,000 | 176 | 0,435 | 0 | 0,000 | 3 | 0,429 | 1 | 0,200 | SoldierTankAircraft | 0,429 |
| 51 | 1,000 | 26 | 0,591 | 0 | 0,000 | 180 | 0,445 | 0 | 0,000 | 7 | 1,000 | 1 | 0,200 | SoldierTankAircraft | 0,445 |
| 51 | 1,000 | 2 | 0,045 | 0 | 0,000 | 88 | 0,215 | 3 | 0,500 | 0 | 0,000 | 0 | 0,000 | Soldier | 0,215 |
| 43 | 0,840 | 0 | 0,000 | 0 | 0,000 | 86 | 0,210 | 1 | 0,167 | 0 | 0,000 | 0 | 0,000 | Soldier | 0,167 |
| 38 | 0,740 | 0 | 0,000 | 0 | 0,000 | 76 | 0,185 | 6 | 1,000 | 5 | 0,714 | 0 | 0,000 | SoldierTank | 0,185 |
| 51 | 1,000 | 7 | 0,159 | 0 | 0,000 | 121 | 0,298 | 0 | 0,000 | 3 | 0,429 | 0 | 0,000 | SoldierTank | 0,298 |
| 48 | 0,940 | 1 | 0,023 | 0 | 0,000 | 101 | 0,248 | 0 | 0,000 | 0 | 0,000 | 1 | 0,200 | SoldierAircraft | 0,200 |
| 51 | 1,000 | 11 | 0,250 | 5 | 1,000 | 185 | 0,458 | 0 | 0,000 | 0 | 0,000 | 2 | 0,400 | SoldierTankAircraft | 0,400 |
| 51 | 1,000 | 16 | 0,364 | 0 | 0,000 | 112 | 0,275 | 0 | 0,000 | 2 | 0,286 | 1 | 0,200 | SoldierTankAircraft | 0,275 |
| 51 | 1,000 | 18 | 0,409 | 0 | 0,000 | 90 | 0,220 | 0 | 0,000 | 2 | 0,286 | 1 | 0,200 | SoldierTankAircraft | 0,220 |

Czas wykonania: 7 ms

Rysunek 13. Struktura wygenerowanych wyników działania logiki rozmytej

Dla algorytmu logiki rozmytej strukturą przedstawiającą zbiór rozwiązań algorytmu jest tabela zawierająca listy wartości podzielonych na poszczególne atrybuty oraz listę wartości oznaczających przynależność do danego zbioru rozwiązań. Struktura została przedstawiona na rysunku 13. Powodem wolniejszego wzrostu czasu wykonania dla rosnącej liczby danych przekazywanych do analizy jest stały przyrost liczby danych przekazanych do analizy. W przypadku pojawiania się nowego rozwiązania, zwiększa się liczba wykonywanych obliczeń. W przeciwieństwie do metody drzew decyzyjnych, zbiór rozwiązań po podziale na poszczególne listy atrybutów nie jest dzielony na mniejsze podzbiory i nie istnieje potrzeba przeprowadzania dodatkowych obliczeń po przydzieleniu każdemu przypadkowi stopnia przynależności do zbioru. Lepszy czas wykonania algorytmu drzew decyzyjnych z zastosowaniem grupowania wartości osiągnął lepszy wynik od metody logiki rozmytej, ponieważ rozmiar struktury był mniejszy, co łączy się też z mniejszą liczbą operacji potrzebnych do przeprowadzenia.

5.2. Test jakości otrzymanych wyników

Dla każdego z zaimplementowanych algorytmów sprawdzono jakość wyników otrzymywanych z systemu wspomagania decyzji. Dla każdej z metod przeprowadzono dziesięć rozgrywek, podczas których na wejście przekazano takie

same dane, aby sprawdzić jakie wyniki zostaną przez nie wypisane na wyjście i jaki będzie rezultat bitwy.

Drzewo decyzyjne

Tabela 2. Zestawienie wyników gier dla przykładowych danych wejściowych dla drzewa decyzyjnego bez zaimplementowanej metody grupowania

| Numer testu | Wystawione jednostki do walki przez gracza (żołnierze-czołgi-samoloty) | Stracone jednostki gracza podczas walki (żołnierze-czołgi-samoloty) | Wystawione jednostki do walki przez sztuczną inteligencję (żołnierze-czołgi-samoloty) | Stracone jednostki sztucznej inteligencji podczas walki (żołnierze-czołgi-samoloty) | Wynik bitwy (Wygrana gracza/SI) |
|-------------|--|---|---|---|---------------------------------|
| 1 | 15-0-0 | 3-0-0 | 13-0-0 | /0-0-0 | Gracz |
| 2 | 12-6-0 | 1-6-0 | 12-0-0 | 0-0-0 | Gracz |
| 3 | 12-2-0 | 0-0-0 | 15-3-0 | 0-3-0 | SI |
| 4 | 12-2-3 | 0-0-3 | 14-3-0 | 0-0-0 | Gracz |
| 5 | 16-0-2 | 0-0-0 | 17-1-3 | 0-0-2 | SI |
| 6 | 0-5-0 | 0-0-0 | 1-6-1 | 0-2-1 | SI |
| 7 | 0-2-0 | 0-2-0 | 1-0-0 | 0-0-0 | Gracz |
| 8 | 0-0-1 | 0-0-1 | 1-1-0 | 0-0-0 | Gracz |
| 9 | 0-15-1 | 0-10-1 | 0-0-2 | 0-0-0 | Gracz |
| 10 | 8-0-1 | 0-0-0 | 0-1-2 | 0-0-1 | SI |

W tabeli 2. zostało przedstawione zestawienie wyników gier dla drzewa decyzyjnego bez zastosowania metody grupowania wartości sumarycznej siły ognia. Wyniki testów pokazują, że spośród dziesięciu rozegranych bitew, tylko cztery z nich zakończyły się sukcesem dla sztucznej inteligencji. Taki wynik jest spowodowany nadmiernym rozrostem drzewa i podziałem wartości ze względu na wartość liczbową, co powoduje powstawanie nowych węzłów specyficznych dla konkretnej wartości siły ognia.

W przypadku zastosowania funkcji grupującej wartości siły ognia, wyniki są o wiele lepsze, co można zaobserwować w tabeli 3. Dla tej metody siedem spośród dziesięciu decyzji zakończyło się sukcesem podczas bitwy. Przyczyną uzyskania takiego wyniku jest wolniejsze rozrastanie się drzewa oraz większe uogólnienie reguł decyzyjnych powstających podczas tworzenia drzewa.

Porównując obydwa warianty klasyfikacji wartości dla drzewa decyzyjnego pod względem jakości wyników i czasów wykonania, można jednoznacznie stwierdzić, że unikając nadmiernego używania wartości liczbowych, czas budowy i przeszukiwania drzewa będzie krótszy oraz jakość wyników systemu wspomagania procesu decyzyjnego będzie większa.

Tabela 3. Zestawienie wyników gier dla przykładowych danych wejściowych dla drzewa decyzyjnego z zaimplementowaną metodą grupowania

| Numer testu | Wystawione jednostki do walki przez gracza (żołnierze-czołgi-samoloty) | Stracone jednostki gracza podczas walki (żołnierze-czołgi-samoloty) | Wystawione jednostki do walki przez sztuczną inteligencję (żołnierze-czołgi-samoloty) | Stracone jednostki sztucznej inteligencji podczas walki (żołnierze-czołgi-samoloty) | Wynik bitwy (Wygrana gracza/SI) |
|-------------|--|---|---|---|---------------------------------|
| 1 | 15-0-0 | 3-0-0 | 13-0-0 | 0-0-0 | Gracz |
| 2 | 12-6-0 | 1-6-0 | 12-0-0 | 0-0-0 | Gracz |
| 3 | 12-2-0 | 0-0-0 | 15-3-0 | 0-3-0 | SI |
| 4 | 12-2-3 | 0-0-3 | 14-3-0 | 0-0-0 | Gracz |
| 5 | 16-0-2 | 0-0-0 | 20-1-3 | 0-0-3 | SI |
| 6 | 0-5-0 | 0-0-0 | 1-6-1 | 1-2-1 | SI |
| 7 | 0-2-0 | 0-0-0 | 1-3-1 | 0-2-1 | SI |
| 8 | 0-0-1 | 0-0-0 | 1-1-2 | 0-0-2 | SI |
| 9 | 0-15-1 | 0-0-0 | 1-17-2 | 0-0-0 | SI |
| 10 | 8-0-1 | 0-0-0 | 10-1-2 | 0-0-2 | SI |

Logika rozmyta

Tabela 4. Zestawienie wyników gier dla przykładowych danych wejściowych dla logiki rozmytej z zastosowaniem metody pierwszego maksimum na etapie wyostrzania

| Numer testu | Wystawione jednostki do walki przez gracza (żołnierze-czołgi-samoloty) | Stracone jednostki gracza podczas walki (żołnierze-czołgi-samoloty) | Wystawione jednostki do walki przez sztuczną inteligencję (żołnierze-czołgi-samoloty) | Stracone jednostki sztucznej inteligencji podczas walki (żołnierze-czołgi-samoloty) | Wynik bitwy (Wygrana gracza/SI) |
|-------------|--|---|---|---|---------------------------------|
| 1 | 15-0-0 | 0-0-0 | 17-0-0 | 3-0-0 | SI |
| 2 | 12-6-0 | 0-6-0 | 15-0-0 | 0-0-0 | Gracz |
| 3 | 12-2-0 | 2-2-0 | 11-0-0 | 0-0-0 | Gracz |
| 4 | 12-2-3 | 2-2-3 | 11-0-0 | 0-0-0 | Gracz |
| 5 | 16-0-2 | 1-0-2 | 16-0-0 | 0-0-0 | Gracz |
| 6 | 0-5-0 | 0-5-0 | 1-0-0 | 0-0-0 | Gracz |
| 7 | 0-2-0 | 0-2-0 | 1-0-0 | 0-0-0 | Gracz |
| 8 | 0-0-1 | 0-0-1 | 1-0-0 | 0-0-0 | Gracz |
| 9 | 0-15-1 | 0-9-1 | 1-0-2 | 0-0-0 | Gracz |
| 10 | 8-0-1 | 0-0-0 | 0-1-2 | 0-0-1 | SI |

Dla logiki rozmytej, w której zastosowano metodę pierwszego maksimum na etapie wyostrzania, wyniki przeprowadzonych testów zostały zestawione w tabeli 4. Na podstawie rezultatów zakończonych bitew można zaobserwować problem związany z pomijaniem rozwiązań znajdujących się za pierwszym maksimum pomimo wystąpienia kilku wartości o najwyższej wartości przynależności do zbioru rozwiązań. Dla przykładowego zestawu testowego okazało się to sporym problemem, gdyż tylko dwie decyzje przyniosły zwycięstwo w rozegranych bitwach z graczem.

Tabela 5. Zestawienie wyników gier dla przykładowych danych wejściowych dla logiki rozmytej z zastosowaniem metody ostatniego maksimum na etapie wyostrzania

| Numer testu | Wystawione jednostki do walki przez gracza (żołnierze-czołgi-samoloty) | Stracone jednostki gracza podczas walki (żołnierze-czołgi-samoloty) | Wystawione jednostki do walki przez sztuczną inteligencję (żołnierze-czołgi-samoloty) | Stracone jednostki sztucznej inteligencji podczas walki (żołnierze-czołgi-samoloty) | Wynik bitwy (Wygrana gracza/SI) |
|-------------|--|---|---|---|---------------------------------|
| 1 | 15-0-0 | 0-0-0 | 18-0-0 | 4-0-0 | SI |
| 2 | 12-6-0 | 0-6-0 | 15-0-0 | 0-0-0 | Gracz |
| 3 | 12-2-0 | 0-0-0 | 13-3-0 | 0-2-0 | SI |
| 4 | 12-2-3 | 0-0-3 | 14-3-0 | 0-0-0 | Gracz |
| 5 | 16-0-2 | 0-0-0 | 18-1-3 | 0-0-2 | SI |
| 6 | 0-5-0 | 0-0-0 | 1-6-1 | 0-2-1 | SI |
| 7 | 0-2-0 | 0-0-0 | 1-3-1 | 0-2-1 | SI |
| 8 | 0-0-1 | 0-0-0 | 1-1-2 | 0-0-2 | SI |
| 9 | 0-15-1 | 0-0-0 | 1-17-2 | 0-0-2 | SI |
| 10 | 8-0-1 | 0-0-0 | 9-1-2 | 0-0-2 | SI |

O wiele lepsze wyniki uzyskano stosując metodę ostatniego maksimum. Na podstawie tabeli 5., w przypadku tego kryterium wyboru najlepszego rozwiązania, można zauważyć dużą skuteczność podejmowanych decyzji, ponieważ osiem z dziesięciu decyzji zakończyło się dla sztucznej inteligencji zwycięstwem.

Na podstawie otrzymanych wyników można stwierdzić, że lepszą jakość wyników, pomimo nieznacznie dłuższego czasu wykonania, otrzymano wykorzystując metodę ostatniego maksimum podczas etapu wyostrzania.

6. Podsumowanie

Problem decyzyjny dotyczy wielu dziedzin nauki, które miały wpływ na opracowanie metod jego rozwiązywania. W przypadku dużej liczby danych potrzebnych do przeprowadzenia oceny każdego wariantu decyzyjnego, istnieje konieczność stworzenia systemu wspomagania decyzji. Taki system tworzony jest często poprzez stworzenie sztucznej inteligencji, która nie tylko pomaga ocenić każdy przypadek, lecz potrafi znaleźć między nimi zależność, dokonuje odpowiedniej klasyfikacji i w oparciu o nią dokonuje decyzji.

Wśród algorytmów możliwych do zaimplementowania można znaleźć algorytmy tworzące drzewa decyzyjne, wykorzystujące elementy logiki rozmytej, sztuczne sieci neuronowe, sieci bayesowskie i wiele innych metod, które mogą być połączeniem wymienionych metod w celu np. zmniejszenia czasu ich wykonania lub zwiększeniu dokładności analizy.

W przypadku środowiska badawczego istotne było sformułowanie odpowiedniego problemu decyzyjnego oraz odpowiednie dobranie algorytmów, które mogłyby sobie z nim poradzić. Zostały wykorzystane metody tworzące drzewa decyzyjne i regulator rozmyty. Do zaimplementowania potrzebnych metod stworzono projekt w środowisku .NET, w którym poprzez wykorzystanie biblioteki Windows Forms było możliwe przygotowanie aplikacji z graficznym interfejsem użytkownika. Aby przedstawić problem decyzyjny występujący w grach komputerowych ważne było przygotowanie systemu umożliwiającego działanie elementów gry strategicznej. Dla tego typu gry postanowiono wybrać problem decyzyjny związany z doбором rodzaju jednostek, które w oparciu o wybór gracza sztuczna inteligencja miała wybrać takie, aby pokonać jego jednostki w trakcie walki.

Cechami wyróżniającymi drzewa decyzyjne było czytelne przedstawienie dylematu decyzyjnego oraz pokazanie zależności występujących pomiędzy poszczególnymi atrybutami, na które podzielono zbiór rozwiązań. Dobranie odpowiednich atrybutów oraz ich typów również pełniło kluczową rolę, ponieważ ich zły dobór mógł spowodować nieprawidłowe wyniki analizy lub mogły spowodować

w przypadku nieprawidłowego typu danych nadmierny rozrost drzewa, co wydłużałoby czas działania systemu podejmowania decyzji.

Logika rozmyta w przeciwieństwie do drzew decyzyjnych nie potrzebowała specjalnego dostosowania typu danych, gdyż w jej przypadku różnorodność danych pomagała w lepszym sklasyfikowaniu wartości. Zaletą tej metody była klasyfikacja wszystkich przypadków, które zostały podzielone na mniejsze podzbiory, dla których oceniono przynależność poszczególnych wartości do pożądanego zbioru rozwiązań. Poprzez system wnioskowania określono relacje pomiędzy poszczególnymi podzbiorami, co w ostateczności pomagało wybrać spośród wszystkich rozwiązań, te które w największym stopniu przynależały do zbioru rozwiązań.

Bibliografia

- [1] *Sytuacja decyzyjna, fazy procesu decyzyjnego*, 1.1., Waldemar Rebizant, *Metody Podejmowania Decyzji*, Oficyna Wydawnicza Politechniki Wrocławskiej, ISBN 978-83-7493-688-0
- [2] *Ryzyko i jego ocena*, 1.2., Waldemar Rebizant, *Metody Podejmowania Decyzji*, Oficyna Wydawnicza Politechniki Wrocławskiej, ISBN 978-83-7493-688-0
- [3] *Zastosowanie algorytmów sztucznej inteligencji w grach komputerowych*, Beata Kuźmińska-Sołśnia, Tomasz Siwiec, <http://www.bks.pr.radom.pl/publikacje/SI%20w%20grach.pdf>, ostatni dostęp 23.10.2016
- [4] *Taiga: Performance Optimization of the C4.5 Decision Tree Construction Algorithm*, Yi Yang and Wenguang Chen, Tsinghua Science and Technology, ISSN 11007-0214/1106/111pp415–425 Volume 21, Number 4, August 2016
- [5] *Drzewa decyzyjne*, Łukasz Bujak, Uniwersytet Mikołaja Kopernika Wydział Fizyki, Astronomii i Informatyki Stosowanej, Toruń 2008, <http://www.is.umk.pl/~duch/Wyklady/CIS/Prace%20zalicz/08-Bujak.pdf>, ostatni dostęp 23.10.2016
- [6] *Automatyczne pozyskiwanie wiedzy*, Monika Demichowicz, Paweł Mazur, 2003 <https://www.ii.pwr.edu.pl/~kwasnicka/tekstystudenckie/apw/id3.htm>, ostatni dostęp 23.10.2016
- [7] *Regulator rozmyty*, Politechnika Warszawska, Wydział Elektryczny, Instytut Sterowania i Elektroniki Przemysłowej, Zakład Napędu Elektrycznego, <http://www.isep.pw.edu.pl/ZakladNapedu/dyplomy/fuzzy/>, ostatni dostęp 23.10.2016

[8] *Use of neural networks as decision makers in strategic situations*, Benoit Couraud, Peilin Liu, Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai, China , Proceedings of the Eighth International Conference on Machine Learning and Cybernetics, Baoding, 12-15 July 2009

[9] *Sztuczne sieci neuronowe*, Krystyna Ambroch, Gdańsk, <http://www.msn.uph.edu.pl/smp/msn/32/ambroch.pdf> , ostatni dostęp 23.10.2016

[10] *Wnioskowanie na podstawie wiedzy niepewnej i niepełnej*, http://wazniak.mimuw.edu.pl/index.php?title=Sztuczna_inteligencja/SI_Modu%C5%82_4_-_Wnioskowanie_na_podstawie_wiedzy_nieprawnej_i_niepe%C5%82nej, Hasło: Wnioskowanie bayesowskie, ostatni dostęp 23.10.2016

[11] *Sieci bayesowskie jako narzędzie wspomagające proces podejmowania decyzji*, Aleksander Król, Zeszyty Naukowe Politechniki Śląskiej 2014 Seria: Organizacja i Zarządzanie Z. 71 Nr Kol. 1917

[12] *Klasyfikacja metodą Bayesa*, Tadeusz Pankowski, <http://etacar.put.poznan.pl/tadeusz.pankowski/hd-08-klasyf-bayesa.pdf>, ostatni dostęp 23.10.2016

Spis rysunków

| | |
|---|----|
| Rysunek 1. Przykładowe drzewo decyzyjne..... | 13 |
| Rysunek 2. Schemat działania regulatora rozmytego | 15 |
| Rysunek 3. Przykładowe wykresy funkcji przynależności | 16 |
| Rysunek 4. Schemat sztucznej sieci neuronowej..... | 20 |
| Rysunek 5. Przykładowa budowa sieci bayesowskiej | 22 |
| Rysunek 6. Ustawienie czasu trwania rozgrywki | 26 |
| Rysunek 7. Ustawienie algorytmu podejmowania decyzji | 27 |
| Rysunek 8. Wyświetlane informacje w trakcie rozbudowy wojska..... | 28 |
| Rysunek 9. Zakładka danych systemu wspomagania decyzji | 28 |
| Rysunek 10. Wyświetlenie wyniku bitwy | 30 |
| Rysunek 11. Wykres przedstawiający czas wykonania zaimplementowanych algorytmów w zależności od liczby wczytanych przykładów | 37 |
| Rysunek 12. Porównanie rozmiaru drzew decyzyjnych bez grupowania i z grupowaniem wartości sumarycznej siły ognia..... | 38 |
| Rysunek 13. Struktura wygenerowanych wyników działania logiki rozmytej | 39 |

Spis tabel

| | |
|--|----|
| Tabela 1. Zestawienie wartości czasu wykonania algorytmów | 36 |
| Tabela 2. Zestawienie wyników gier dla przykładowych danych wejściowych dla drzewa decyzyjnego bez zaimplementowanej metody grupowania | 40 |
| Tabela 3. Zestawienie wyników gier dla przykładowych danych wejściowych dla drzewa decyzyjnego z zaimplementowaną metodą grupowania | 41 |
| Tabela 4. Zestawienie wyników gier dla przykładowych danych wejściowych dla logiki rozmytej z zastosowaniem metody pierwszego maksimum na etapie wyostrozania | 42 |
| Tabela 5. Zestawienie wyników gier dla przykładowych danych wejściowych dla logiki rozmytej z zastosowaniem metody ostatniego maksimum na etapie wyostrozania | 43 |

Dodatek A.

Załączona płyta DVD zawiera:

- Kody źródłowe,
- Aplikację ze środowiskiem badawczym,
- Pliki związane z obsługą bazy danych historycznych,
- Elektroniczną wersję pracy w formacie pdf.