

# Spis treści

<b>1</b>	<b>Wspomagane wyszukiwanie drużyn - wstęp teoretyczny</b>	<b>2</b>
1.1	Problem optymalizacji wielokryterialnej . . . . .	2
1.1.1	Tradycyjny sposób rozwiązywania problemu . . . . .	2
1.1.2	Normalizacja - metody liniowe . . . . .	2
1.1.3	Agregacja - Metoda ważonych kryteriów . . . . .	3
1.2	Lokalizacja problemu w projektowanym systemie . . . . .	3
1.2.1	Wymagania czasowe dla algorytmu . . . . .	3
1.2.2	Propozycja algorytmu - ogólny schemat . . . . .	3

# 1 Wspomagane wyszukiwanie drużyn - wstęp teoretyczny

## 1.1 Problem optymalizacji wielokryterialnej

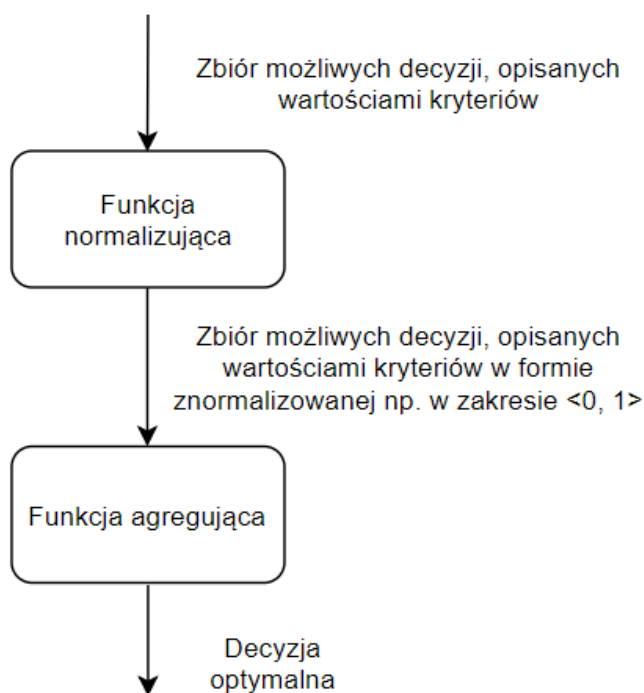
Problem optymalizacji wielokryterialnej jest rozszerzeniem problemu optymalizacji jednokryterialnej, gdzie poszukiwana jest decyzja optymalna, ze zbioru możliwych decyzji na podstawie jednego kryterium. Problem ten sprowadza się do poszukiwania maksimum (bądź minimum) funkcji oceny danego kryterium. Jeżeli kryterium jest ilościowe, np. maksymalna prędkość samochodu, to podejmując decyzję o zakupie jedynie ze względu na to kryterium naturalnie wybierzemy samochód, który osiąga największą prędkość.

Często jednak podjęcie decyzji może być uwarunkowane większą ilością czynników, na przykład, w przypadku samochodu może to być jego cena, koszt utrzymania, czy czynniki nie ilościowe takie jak jego kolor (w zależności od preferencji kupca). Podejmując decyzję należy rozpatrzyć wiele kryteriów oraz relacje między nimi. Decyzja, która jest optymalna ze względu na jedno z kryteriów, nie musi być optymalną ze względu na pozostałe - z reguły tak nie jest.

W literaturze można znaleźć wiele sposobów na rozwiązywanie tego typu problemu. W dalszej części pracy zostanie wybrany sposób, który najbardziej pasuje do domeny problemu.

### 1.1.1 Tradycyjny sposób rozwiązywania problemu

Tradycyjne algorytmy składają się z dwóch etapów. W pierwszej kolejności następuje normalizacja wartości kryteriów, a następnie agregacja, której wynikiem jest decyzja optymalna.



Rysunek 1: Ogólny schemat metody rozwiązywania problemu optymalizacji wielokryterialnej

### 1.1.2 Normalizacja - metody liniowe

Tradycyjnymi metodami normalizacji kryteriów są metody normalizacji liniowej. W wyniku przekształceń wartości kryteriów zostają znormalizowane do przedziału wartości  $<0, 1>$ .

Tabela 1: Liniowe metody normalizacji kryteriów [1]

	Procedure 1	Procedure 2	Procedure 3
Definition	$v_i = \frac{x_i}{\max x_i}$	$v_i = \frac{x_i - \min x_i}{\max x_i - \min x_i}$	$v_i = \frac{x_i}{\sum x_i}$
Normalized vector*	$0 \leq v_i \leq 1$	$0 \leq v_i \leq 1$	$0 \leq v_i \leq 1$
Constraints	$\max v_i = 1$	$\min v_i = 0, \max v_i = 1$	$\sum v_i = 1$
Proportionality conserved	Yes	No	Yes
Interpretation	% of maximum $x_i$	% of range ( $\max x_i - \min x_i$ )	% of total $\sum x_i$

### 1.1.3 Agregacja - Metoda ważonych kryteriów

Metoda ważonych kryteriów polega na opisanu funkcji oceny decyzji jako sumy ważonej ocen poszczególnych kryteriów. Konieczne jest przyporządkowanie wagi dla każdego z kryteriów. Ocena poszczególnych decyzji obliczana jest według wzoru:

« Wstawić wzór »

## 1.2 Lokalizacja problemu w projektowanym systemie

Jednym z podstawowych zadań systemu jest wspomaganie wyboru przeciwników. System powinien zaproponować kapitanowi drużyny listę sugerowanych rywali będącą jak najbardziej dopasowaną do jego preferencji. Można zdefiniować wiele kryteriów, które wpływają na jakość dopasowania. Mogą to być m.in średni wiek zawodników, łączny staż gry, odległość punktów macierzystych drużyn, pokrycie preferowanych godzin gry, poziom fair play drużyny przeciwnej, ocena poprzednich rozgrywek z daną drużyną, poziom rankingowy w systemie.

### 1.2.1 Wymagania czasowe dla algorytmu

System musi być w stanie dokonać oceny przeciwników w rozsądnym czasie. Rozsądnym limitem czasowym jaki kapitan może oczekiwać na wynik działania systemu jest 5 sekund. Wyszukiwanie przeciwników będzie odbywać się dla konkretnej dyscypliny sportu oraz regionu. Biorąc pod uwagę skalę systemu, można założyć, że w danym regionie nie powinno być więcej niż 5000 drużyn. W przypadku większych regionów powinny one zostać podzielone na mniejsze podregiony. Z tych ustaleń wynika, że zaproponowany oraz zaimplementowany algorytm powinien być w stanie przetworzyć 5000 drużyn w czasie nie większym niż 5 sekund. Zaimplementowany algorytm zostanie przetestowany pod kątem tych ograniczeń.

### 1.2.2 Propozycja algorytmu - ogólny schemat

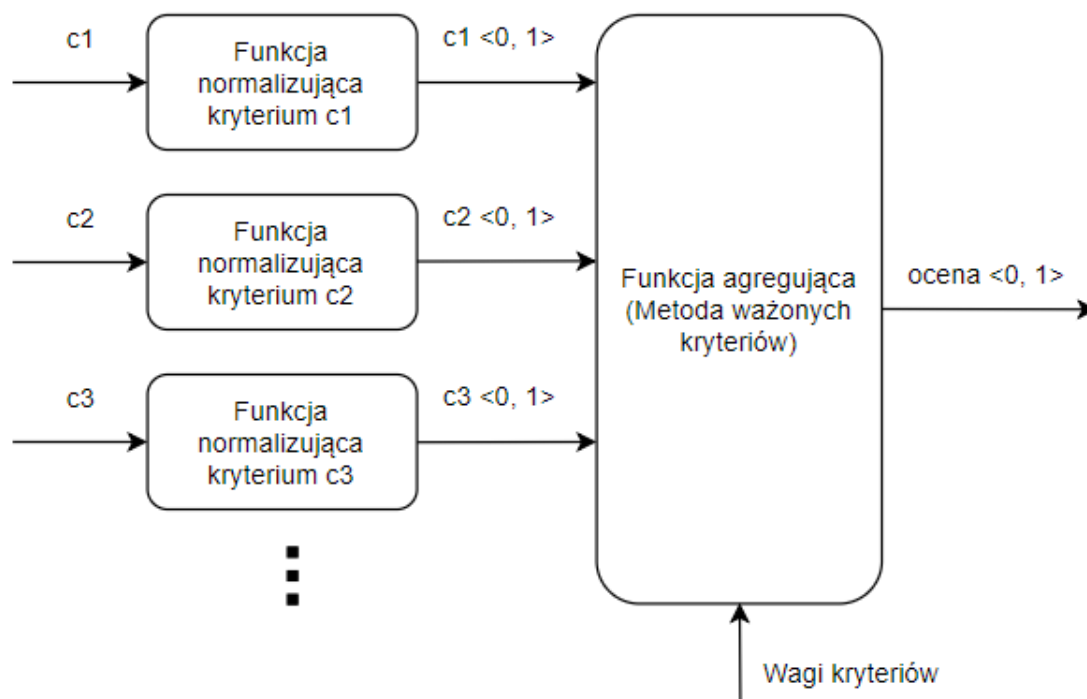
Proponując algorytm należy mieć na uwadze domenę problemu oraz charakter danych, na jakich będzie on operował. W związku z tym do tradycyjnego schematu oceny rozwiązań zostanie wprowadzona modyfikacja.

Normalizacja poszczególnych kryteriów będzie przebiegać przy użyciu różnych funkcji, dopasowanych do charakteru danych. Przykładem uzasadniającym konieczność zastosowania takiej modyfikacji może być porównanie dwóch kryteriów: odległości drużyn oraz średniego wieku zawodników. O ile kryterium odległości może być normalizowane w pełni liniowo, o tyle dla różnicy wieku taka metoda normalizacji jest błędna. Różnica wieku między dwoma zawodnikami, którzy mają 15 oraz 20 lat jest znacznie bardziej istotna niż różnica między zawodnikami w wieku 30 oraz 35 lat - nie można tutaj zastosować operatora liniowego.

Dodatkowo w domenie problemu można wyróżnić kryteria nieliczbowe – cechy drużyny, które mogą mieć duży wpływ na jakość dopasowania, np. zadeklarowana chęć ponownej gry z daną drużyną.

Jako funkcja agregująca wybrana została tradycyjna metoda ważonych kryteriów ze względu na możliwość dynamicznego doboru wag poszczególnych kryteriów. Nietóre z tych wag będą dobierane przez kapitana zgodnie z preferencjami jego drużyny.

Na schemacie (Rysunek 2.) został przedstawiony ogólny schemat algorytmu. Symbole  $c_1$ ,  $c_2$ ,  $c_3$  oznaczają przykładowe kryteria.



Rysunek 2: Uogólniony schemat algorytmu oceny decyzji

## Literatura

- [1] dr hab. Mieczysław Połoński prof. SGGW: *Analiza wielokryterialna – wstęp do zagadnienia*
- [2] Lucia Vavříková, Department of Mathematics, Faculty of Civil Engineering, Slovak University of Technology: *Multicriteria Decision Making and Rankings Based on Aggregation Operators*