

# Mesterséges Intelligencia Szorgalmi Feladat

Téma: Kör illesztése adott ponthalmazra 2.

**Készítette:** Mile Kolos

**Neptun-kód:** OXEZ80

**E-mail:** kolosk5@gmail.com

**Konzulens:** Dr. Póka György

Budapesti Műszaki és Gazdaságtudományi Egyetem  
Gépészmérnöki Kar  
Gyártástudomány és Technológia Tanszék

Budapest, 2025. november 18.

# Contents

<b>1 A feladat leírása</b>	<b>2</b>
1.1 A probléma bemutatása . . . . .	2
1.2 Célkitűzések . . . . .	2
<b>2 A megoldás elve, módszere</b>	<b>3</b>
2.1 Technológiai háttér . . . . .	3
2.2 Ponthalmaz generálása hibákkal . . . . .	3
<b>3 Az implementáció bemutatása</b>	<b>4</b>
3.1 A projekt struktúrája . . . . .	4

# 1 A feladat leírása

## 1.1 A probléma bemutatása

A feladat egy klasszikus geometriai-optimalizálási probléma modern, mesterséges intelligencia alapú megközelítése. A cél egy adott 2D ponthalmazra a **legkisebb befoglaló kör** (minimum enclosing circle) meghatározása. A kihívást az jelenti, hogy a ponthalmaz nem ideális, hanem valós mérési vagy digitalizálási folyamatokból származó hibákat szimulál, úgymint:

- **Alakhiba:** A pontok nem egy tökéletes köríven helyezkednek el.
- **Véletlen zaj:** minden pont pozíciója egy kis mértékű, véletlenszerű eltolással terhelt.
- **Kiugró pontok (outlierek):** A ponthalmaz tartalmaz néhány, a fő csoporttól távol eső, hibás mérési eredményt szimuláló pontot.

Az algoritmusnak robusztusnak kell lennie, hogy ezekkel a hibákkal megbirkózzon, és a definíció szerint megtalálja azt a legkisebb sugarú kört, amely a ponthalmaz összes elemét tartalmazza.

## 1.2 Célkitűzések

A projekt során a következő fő célokat kellett elérni:

1. **Ponthalmaz generálása:** Egy olyan programmodul létrehozása, amely képes paraméterező módon, a fent említett hibákkal terhelt ponthalmazokat generálni.
2. **Algoritmus fejlesztése:** Egy mesterséges intelligencia alapú algoritmus (esetünkben genetikus algoritmus) implementálása, amely a generált ponthalmazra illeszti a legkisebb befoglaló kört.
3. **Kiértékelés:** A kifejlesztett módszer teljesítményének objektív elemzése. Vizsgálni kell a futási időt a bemeneti adatok méretének függvényében, elemezni kell az algoritmus paramétereinek (pl. mutációs ráta) hatását az eredményre, és vizsgálni kell a megoldás konvergenciáját.
4. **Dokumentálás:** A teljes folyamat, a módszer és az elért eredmények részletes dokumentálása a követelményeknek megfelelően.

## 2 A megoldás elve, módszere

A probléma megoldására egy Python-alapú szoftveres megoldás készült, amely egy genetikus algoritmust alkalmaz a legkisebb befoglaló kör megkeresésére.

### 2.1 Technológiai háttér

A projekt az alábbi technológiákra épül:

- **Nyelv:** Python 3
- **Könyvtárak:**
  - **NumPy:** A numerikus számítások (távolságmérés, koordináta-manipuláció) hatékony elvégzéséért felel. Nélkülözhetetlen a nagy mennyiségű pontadat gyors feldolgozásához.
  - **Matplotlib:** Az adatok és eredmények vizualizációjáért felel. Segítségével ábrázoljuk a generált ponthalmazokat, az illesztett köröket és a kiértékelés során kapott grafikonokat.

### 2.2 Ponthalmaz generálása hibákkal

A kiindulási adathalmazt egy dedikált modul (`point_generator.py`) hozza létre, amely egy ideális körből indul ki, és szisztematikusan hibákat ad hozzá:

1. **Alap kör definiálása:** Egy  $(x, y)$  középpontú,  $r$  sugarú körön egyenletesen elhelyezünk  $N$  számú pontot.
2. **Alakhiba hozzáadása:** A pontok koordinátáit enyhén torzítjuk, ami egy ellipszis-szerű alakot eredményez.
3. **Véletlen zaj hozzáadása:** minden pont  $x$  és  $y$  koordinátájához egy normális eloszlású véletlen értéket adunk, ami a mérési pontatlanságot szimulálja.
4. **Kiugró pontok generálása:** A fő ponthalmazon kívül, nagyobb távolságra elhelyezünk néhány pontot, amelyek a durva mérési hibákat reprezentálják.

### 3 Az implementáció bemutatása

A szoftver Python 3 nyelven készült, moduláris felépítéssel, hogy a különböző funkciók (pontgenerálás, algoritmus, kiértékelés) logikailag elkülönüljenek.

#### 3.1 A projekt struktúrája

A projekt fő mappái és fájljai a következők:

```
.  
|-- docs/                      # Dokumentacio es feladatkiiras  
|-- src/                        # A Python forraskodok  
|   |-- point_generator.py  
|   |-- genetic_algorithm.py  
|   |-- evaluation.py  
|   |-- main.py  
|-- venv/                       # Virtualis környezet  
|-- .gitignore  
|-- requirements.txt            # Projekt függősegek  
|-- README.md
```