

Mesterséges intelligencia

Gregorics Tibor

people.inf.elte.hu/gt/mi

□ Könyvek

- Fekete István - Gregorics Tibor - Nagy Sára: Bevezetés a mesterséges intelligenciába, LSI Kiadó, Budapest, 1990, 1999. ELTE-Eötvös Kiadó, Budapest, 2006.
- Russel, J. S., Norvig, P.: MI - modern megközelítésben, Panem Kft, 2005.
- Futó Iván (szerk): Mesterséges intelligencia, Aula Kiadó, Budapest, 1999.

□ Internet

- people.inf.elte.hu/gt/m

Bevezetés

1. AZ MI FOGALMA

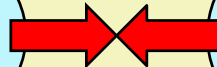
mesterséges intelligencia – MI (artificial intelligence - AI)

Erős MI

Az emberi gondolkodás
reprodukálható
számítógéppel.

MI szkeptikusok

A számítógép soha
nem lesz okosabb az
embernél.



Gyenge MI

Az MI kutatja, fejleszti, rendszerezi azokat az elméleteket és módszereket, amelyek hozzájárulhatnak az intelligens gondolkodás számítógéppel való reprodukálásához.

MI nem egy speciális részterülete az informatikának, hanem egy törekvés, hogy a számítógéppel olyan érdekes és nehéz problémákat oldjunk meg, amelyek megoldásában ma még az ember jobb.

Miről ismerhető fel egy szoftverben az MI?

- Megoldandó feladat: nehéz
 - A feladat **problématere** hatalmas,

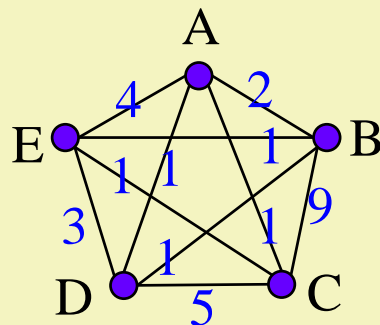
- Szoftver viselkedése

- Felhasznált eszközök



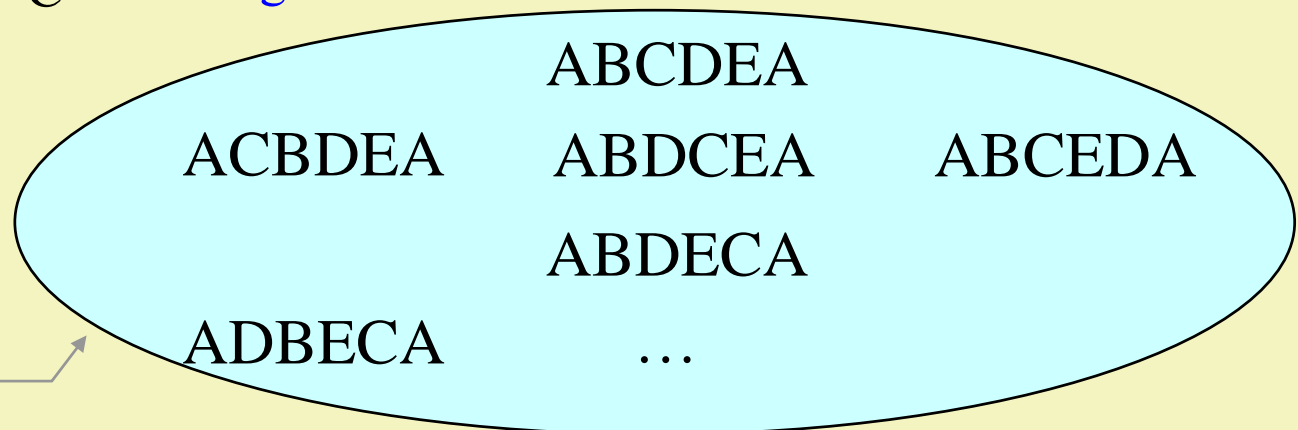
Utazó ügynök problémája

Adott n város a közöttük vezető utak költségeivel. Melyik a legolcsóbb olyan útvonal, amely az A városból indulva mindegyik várost egyszer érintve visszatér az A városba?



lehetséges utak:

n	$(n-1)!$
5	24
50	$6 \cdot 10^{62}$



problématér

Miről ismerhető fel egy szoftverben az MI?

- ❑ Megoldandó feladat: nehéz
 - A feladat **problématere** hatalmas,
 - szisztematikus keresés helyett intuícióra, kreativitásra (azaz **heurisztikára**) van szükségünk ahhoz, hogy elkerüljük a **kombinatorikus robbanást**.
- ❑ Szoftver viselkedése
- ❑ Felhasznált eszközök



7-11

$x=?, y=?, z=?, t=?$

$$\begin{cases} x + y + z + t = 7.11 \\ x \cdot y \cdot z \cdot t = 7.11 \end{cases}$$



$x, y, z, t \in \{1, \dots, 708\}$

$$\begin{cases} x + y + z + t = 711 \\ x \cdot y \cdot z \cdot t = 711 \cdot 10^6 = 2^6 \cdot 3^2 \cdot 5^6 \cdot 79 \end{cases}$$



$x = 79$

$x = 2 \cdot 79$

...

~~$x = 7 \cdot 79$~~

$x = 8 \cdot 79$

~~$x = 9 \cdot 79 = 711$~~

$$\begin{cases} y + z + t = 632 \\ y \cdot z \cdot t = 2^6 \cdot 3^2 \cdot 5^6 \end{cases}$$

$$\begin{cases} y + z + t = 553 \\ y \cdot z \cdot t = 2^5 \cdot 3^2 \cdot 5^6 \end{cases}$$

...

$$\begin{cases} y + z + t = 79 \\ y \cdot z \cdot t = 2^3 \cdot 3^2 \cdot 5^6 \end{cases}$$

Miről ismerhető fel egy szoftverben az MI?

- ❑ **Megoldandó feladat: nehéz**
 - A feladat **problématere** hatalmas,
 - szisztematikus keresés helyett intuícióra, kreativitásra (azaz **heurisztikára**) van szükségünk ahhoz, hogy elkerüljük a **kombinatorikus robbanást**.
- ❑ **Szoftver viselkedése: intelligens**
 - Turing teszt
- ❑ **Felhasznált eszközök**

ELIZA

↖ I feel you are bored with me lately.

Pattern: *I <a> you me <c>.*

1. Why do you think that you <a> I you <c>?
2. Let us suppose that I you <c>. Would that make a difference?

Recall:

„I am getting tired of replying the same sentence over and over.”

Carry on:

What else do you want to talk about?

I see. Please continue. This is very interesting.

Miről ismerhető fel egy szoftverben az MI?

Intelligens szoftver jellemzői

- megszerzett ismeret tárolása
- automatikus következtetés
- tanulás
- term. nyelvű kommunikáció
+ gépi látás, gépi cselekvés

❑ Megoldandó feladat: nehéz

- A feladat **problématere** hatalmas,
- szisztematikus keresés helyett intuícióra, kreativitásra (azaz **heurisztikára**) van szükségünk ahhoz, hogy elkerüljük a **kombinatorikus robbanást**.

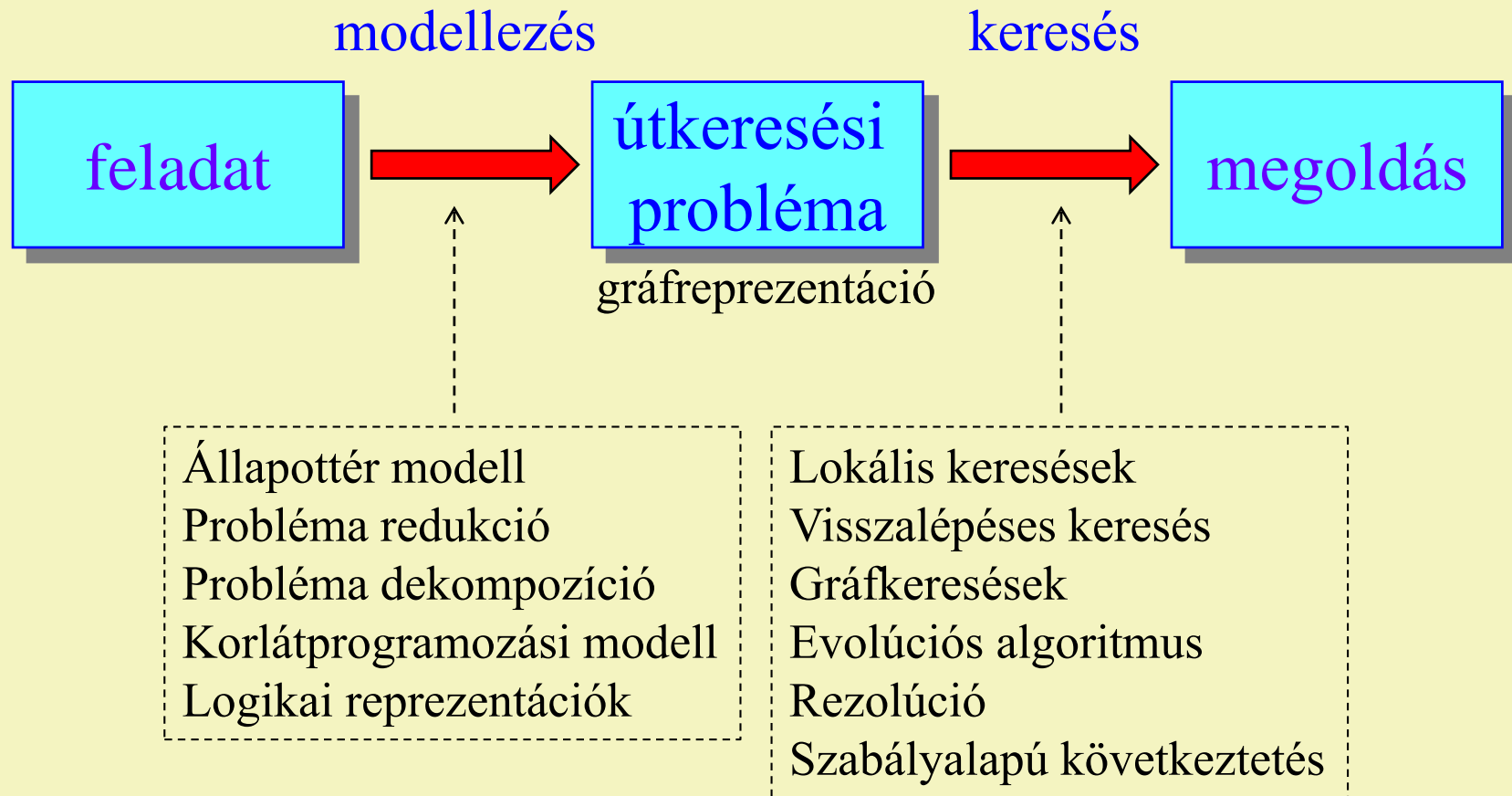
❑ Szoftver viselkedése: intelligens

- Turing teszt vs. kínai szoba elmélet
- „mesterjelölt szintű” és „egy problémára fókuszál”

❑ Felhasznált eszközök: sajátosak

- átgondolt reprezentáció a feladat **modellezéséhez**
- heurisztikával megerősített hatékony **algoritmusok**
- **gépi tanulás** módszerei

2. MODELLEZÉS & KERESÉS



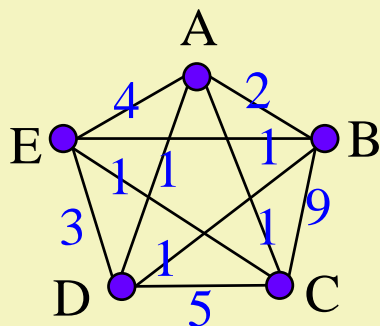
Mire kell a modellezésnek fókuszálni

- ❑ *Problématér elemei*: probléma lehetséges válaszai.
- ❑ *Cél*: egy helyes válasz (megoldás) megtalálása
- ❑ *Keresést segítő ötletek* (heurisztikák):
 - Problematér hasznos elemeinek elválasztása a haszontalanoktól.
 - Kiinduló elem kijelölése.
 - Az elemek szomszédsági kapcsolatainak kijelölése, hogy a probléma tér elemeinek szisztematikus bejárását segítsük.
 - Adott pillanatban elérhető elemek rangsorolása.



Utazó ügynök problémája

Adott n város a közöttük vezető utak költségeivel. Melyik a legolcsóbb olyan útvonal, amely az A városból indulva mindegyik várost egyszer érintve visszatér az A városba?



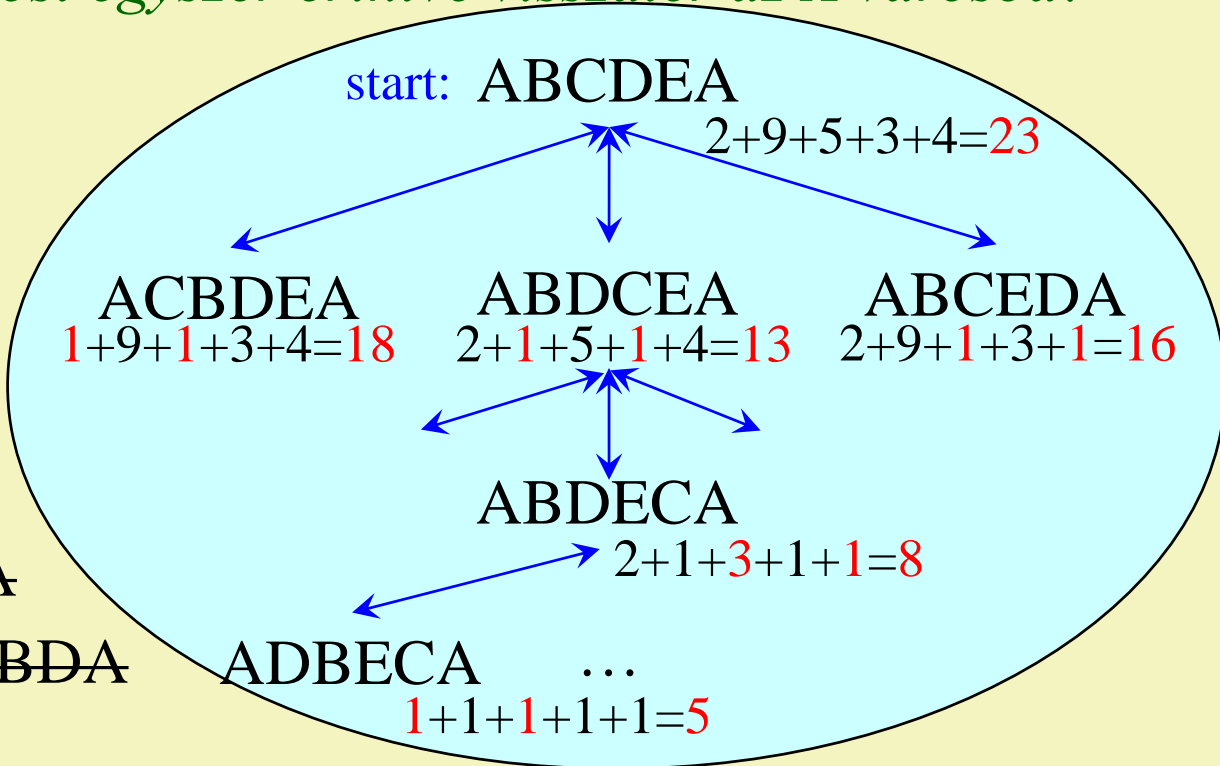
~~AEDCBA~~

~~AEDBCA~~

~~ACEDBA~~

felesleges

~~ACEBDA~~



Útkeresési probléma

- ❑ Számos olyan modellező módszert ismerünk, amely a kitűzött feladatot útkeresési problémává fogalmazza át.
- ❑ Az útkeresési probléma megoldását egy alkalmas élsúlyozott irányított gráfnak vagy egy adott (cél-) csúcsa, vagy egy adott (startcsúcsból célcsúcsba vezető) útja (esetleg a legolcsóbb ilyen út) szimbolizálja.
- ❑ Ez a gráf lehet végtelen nagy, de csúcsainak kifoka véges, és van egy konstans globális pozitív alsó korlátja (δ) az éleinek súlyának (költségének) (δ -gráf).

Gráf fogalmak 1.

- csúcsok, irányított élek
- él n -ből m -be
- n utódai
- n szülei
- irányított gráf
- véges sok kivezető él
- élköltség
- δ -tulajdonság ($\delta \in \mathbb{R}^+$)
- δ -gráf

$N, A \subseteq N \times N$ (végtelen számosság)

$(n, m) \in A \quad (n, m \in N)$

$\Gamma(n) = \{m \in N \mid (n, m) \in A\}$

$\pi(n) \in \Pi(n) = \{m \in N \mid (m, n) \in A\}$

$R = (N, A)$

$|\Gamma(n)| < \infty \quad (\forall n \in N)$

$c: A \rightarrow \mathbb{R}$

$c(n, m) \geq \delta > 0 \quad (\forall (n, m) \in A)$

δ -tulajdonságú, véges sok kivezető
élű, élsúlyozott irányított gráf

Gráf fogalmak 2.

- irányított út

δ -gráfokban ez végtelen sok út esetén is értelmes.

Értéke ∞ , ha nincs egy út se.

- út hossza

- út költsége

- opt. költség

- opt. költségű út

$$\alpha = (n, n_1), (n_1, n_2), \dots, (n_{k-1}, m)$$

$$= \langle n, n_1, n_2, \dots, n_{k-1}, m \rangle$$

$$n \rightarrow^\alpha m, n \rightarrow m, n \rightarrow M \quad (M \subseteq N)$$

$$\{n \rightarrow m\}, \{n \rightarrow M\} \quad (M \subseteq N)$$

az út éleinek száma: $|\alpha|$

$$c(\alpha) = c^\alpha(n, m) := \sum_{i=1..k} c(n_{i-1}, n_i)$$

$$\text{ha } \alpha = \langle n = n_0, n_1, n_2, \dots, n_{k-1}, m = n_k \rangle$$

$$c^*(n, m) := \min_{\alpha \in \{n \rightarrow m\}} c^\alpha(n, m)$$

$$c^*(n, M) := \min_{\alpha \in \{n \rightarrow M\}} c^\alpha(n, m)$$

$$n \rightarrow^* m := \min_c \{ \alpha \mid \alpha \in \{n \rightarrow m\} \}$$

$$n \rightarrow^* M := \min_c \{ \alpha \mid \alpha \in \{n \rightarrow M\} \}$$

Gráfrepresentáció fogalma

- Minden útkeresési probléma rendelkezik egy (a probléma modellezéséből származó) gráfrepresentációval, ami egy (R, s, T) hármas, amelyben
 - $R = (N, A, c)$ δ -gráf az ún. **representációs gráf**,
 - az $s \in N$ **startcsúcs**,
 - a $T \subseteq N$ halmazbeli **célcsúcsok**.
- és a probléma megoldása:
 - egy $t \in T$ cél megtalálása, vagy
 - egy $s \rightarrow T$, esetleg $s \rightarrow^* T$ optimális út megtalálása

s -ből T egyik csúcsába vezető irányított út

s -ből T egyik csúcsába vezető legolcsóbb irányított út

Keresés

- Az útkeresési problémák megoldásához azok reprezentációs gráfjainak nagy mérete miatt speciális (nem determinisztikus, heurisztikus) útkereső algoritmusokra van szükség, amelyek
 - a startcsúcsból **indulnak** (kezdeti aktuális csúcs);
 - minden lépésben **nem-determinisztikus** módon új aktuális csúcs(ka)t **választanak** a korábbi aktuális csúcs(ok) alapján (gyakran azok gyerekei közül);
 - **tárolják** a már feltárt reprezentációs gráf egy részét;
 - **megállnak**, ha célcsúcsot találnak vagy nyilvánvalóvá válik, hogy erre semmi esélyük.

Kereső rendszer (KR)

Procedure KR

1. **ADAT** \leftarrow kezdeti érték
 2. **while** \neg terminálási feltétel(**ADAT**) **loop**
 3. **SELECT SZ FROM** alkalmazható szabályok
 4. **ADAT** $:=$ **SZ**(**ADAT**)
 5. **endloop**
- end**

globális munkaterület

tárolja a keresés során megszerzett és megőrzött ismeretet (egy részgráfot)
(kezdeti érték ~ start csúcs,
terminálási feltétel ~ célcsúcs)

keresési szabályok

megváltoztatják a globális munkaterület tartalmát
(előfeltétel, hatás)

vezérlési stratégia

alkalmazható szabályok közül kiválaszt egy „megfelelőt”
(általános elv + heurisztika)

Kereső rendszerek vizsgálata

- ❑ helyes-e (azaz korrekt választ ad-e)
- ❑ teljes-e (minden esetben választ ad-e)
- ❑ optimális-e (optimális megoldást ad-e)
- ❑ idő bonyolultság
- ❑ tár bonyolultság

3. GÉPI TANULÁS

- ❑ Egy algoritmus tanul, ha egy feladat megoldása során olyan változások következnek be a működésében, hogy később ugyanazt a feladatot vagy hasonló feladatokat jobban (eredmény, hatékonyság) képes megoldani, mint korábban.
- ❑ Gépi tanulással a feladat modelljét (reprezentációját és/vagy heurisztikáit), illetve a megoldó algoritmust (többnyire annak bizonyos paramétereit) lehet automatikusan előállítani.
- ❑ A tanuláshoz a megoldandó probléma néhány konkrét esetére, tanító példákra van szükség. A tanulás attól függően lesz **felügyelt**, **felügyelet nélküli**, vagy **megerősítéses**, hogy a tanító példák input-output párok, csak inputok, vagy input-hasznosság párok.