Alkalmazott mesterséges intelligencia (AMI)

http://www.mit.bme.hu/oktatas/targyak/vimibb01

2 ea.- (2023 ősz)

Problémamegoldás kereséssel – a problémára vonatkozó információ felhasználása (heurisztika)

(Dobrowiecki Tadeusz és Hullám Gábor anyagainak felhasználásával)

http://mialmanach.mit.bme.hu/aima/ch04

https://www.esrcheck.com/2023/06/05/artificial-intelligence-ai-experts-sign-statement-on-ai-risk/

Előadó: Pataki Béla

BME I.E. 414, 463-26-79

pataki@mit.bme.hu,

http://www.mit.bme.hu/general/staff/pataki

A neminformált keresési stratégiák – általános (nem problémaspecifikus) módszerek

b - elágazási tényező, *d* - a megoldás mélysége,
m - a keresési fa maximális mélysége, *l* - a mélységkorlát. (*m* és *l* lehet nagyobb d-nél!)

Jellem-ző	SzK	EgyKK	MK	MKK	IMK	KK
Idő-igény	b^d	b^d	b^m	b^l	b^d	$b^{d/2}$
Tár-igény	b^d	b^d	bm	bl	bd	$b^{d/2}$
Opt.?	Igen (ha…)	Igen	Nem	Nem	Igen	Igen
Teljes?	Igen	Igen	Nem	Igen, ha $l \geq d$	Igen	Igen

Az exponenciális tár-, illetve időigény komoly problémát jelent!
Jobb módszerek kellenek!

Heurisztikus, vagy más néven **informált keresési stratégiák**

Irányítsuk a keresést a célállapot irányába.

Ehhez kell valami elképzelés, hogy a cél:

- ***merrefelé***
- ***nagyjából milyen messzire fekszik*.**

Heurisztika, **heurisztikus függvény *h*(*n*)**:

- a probléma minden *n* állapotára ki kell tudnunk számítani
- kifejezi az adott állapotból a célig előrehaladás becsült költségét
- ha pontos, akkor elvben fölöslegessé teszi a keresést

(ha nagyon pontatlan, akkor viszont semmit sem segít)

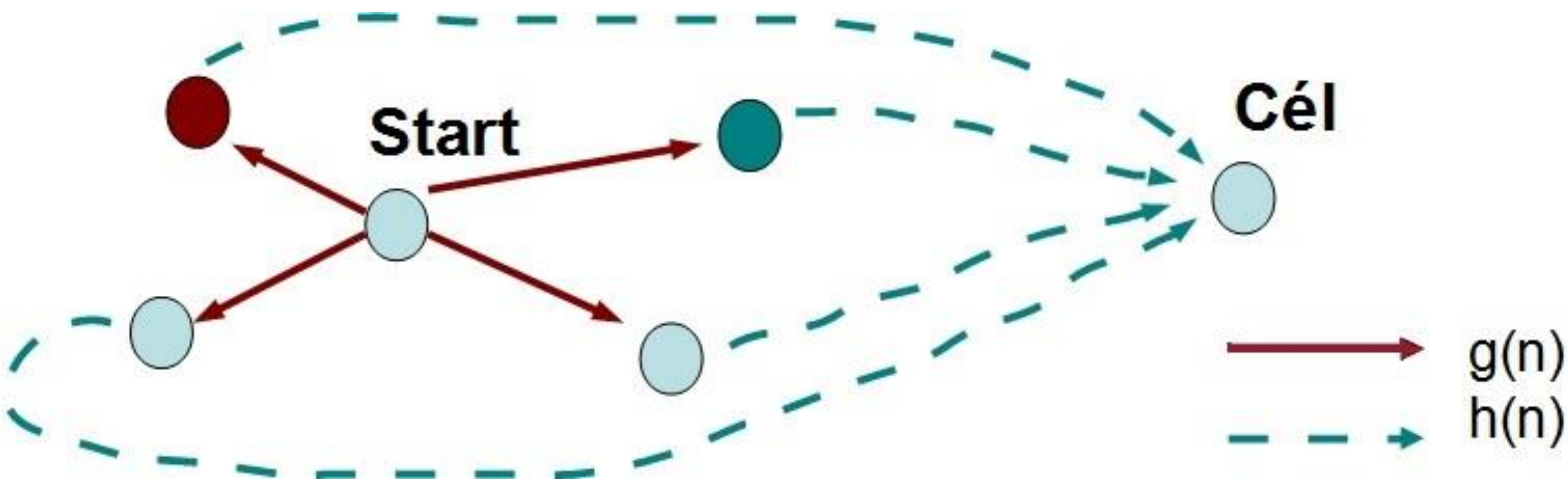
Ez minden problémára más, problémaspecifikus!

Jutalmazzuk a cél irányába történő keresést!

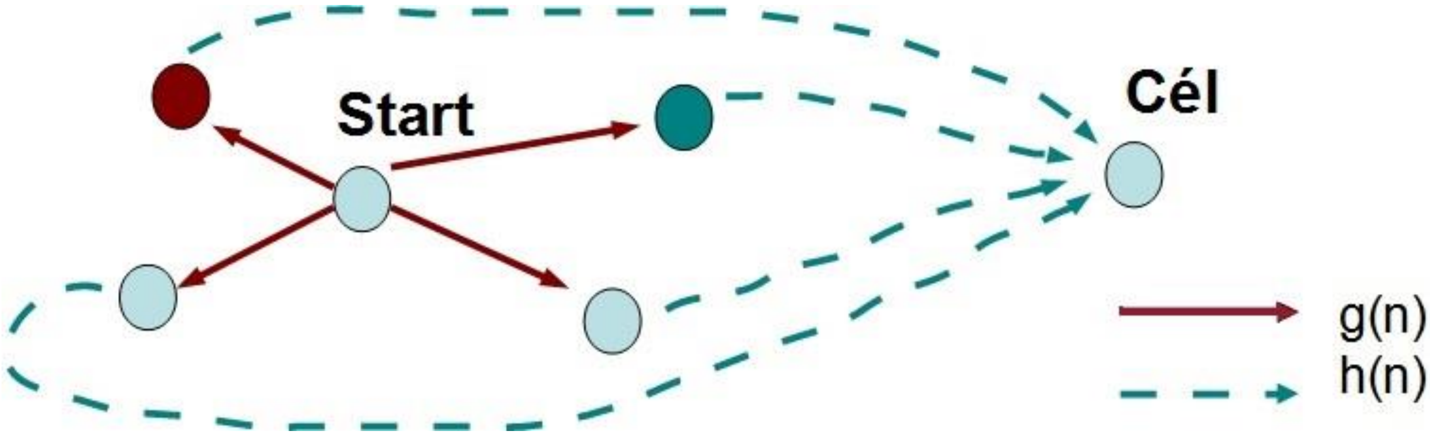
h(*n*) becslés: mekkora a hátralévő út legkisebb költsége.

ez egyben ⇒becslés: a teljes út legkisebb költségére – $f(n)$

$$f(n) = g(n) + h(n)$$



Tökéletes információ (heurisztika) ⇒ előretartás elágazások nélkül (hiszen minden csomópontra pontosan tudjuk, hogy mibe kerül ezen az úton menni) ⇒ lineáris tárigény = lineáris időigény!



$$f(n) = g(n) + h(n)$$

Jó heurisztikus függvény - a tár- és időigény jelentősen csökkenthető (a csökkenés az adott problémától és a *h* függvény minőségétől függ).

Elfogadható heurisztika = ***h(n)*** függvény **soha nem becsüli felül** a cél eléréséhez szükséges **költséget** (**optimista**: a cél közelebbinek, legalábbis nem távolabbinak tűnik, mint amilyen)

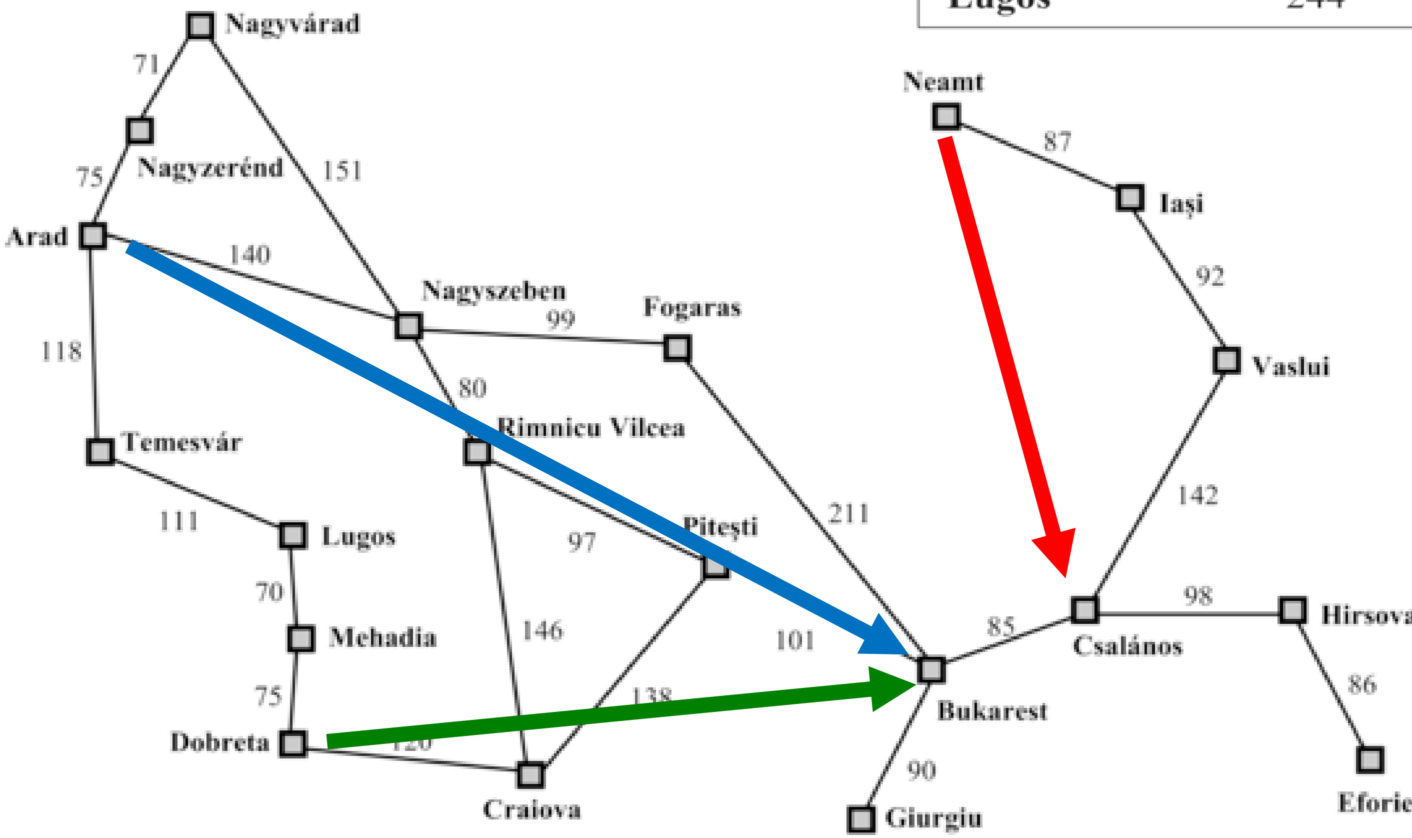
(Optimista heurisztika haszna: lásd Lázár Ervin A hazudós egér c. meséje – pl. a Farkasnak adott heurisztika)

Ha ***h(n)*** elfogadható, akkor ***f(n)*** sohasem becsüli túl az ***n***-dik csomóponton át vezető legjobb megoldás valódi költségét

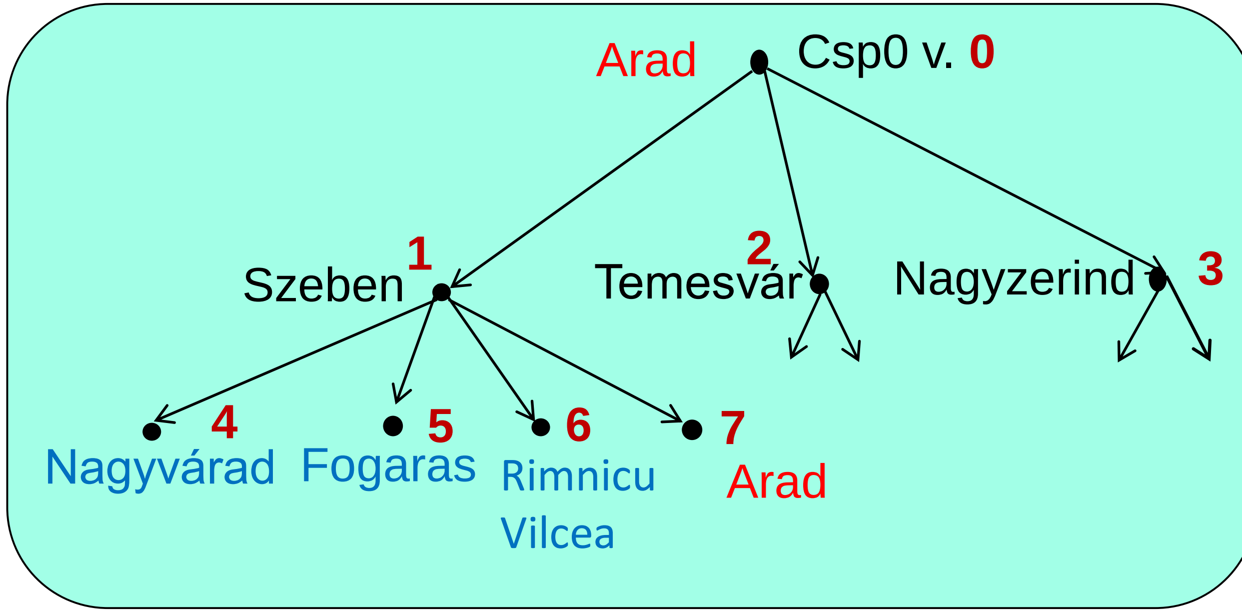
Legyen az útkeresési feladatunkban a heurisztikus függvény a légvonalban mért távolság ($\boldsymbol{h}_{LMT}$)

A feltételeket teljesíti?
Mi mondható a hibájáról?

Arad	366	Mehadia	241
Bukarest	0	Neamt	234
Craiova	160	Nagyvárad	380
Dobreta	242	Piteşti	100
Eforie	161	Rimniu Vilcea	193
Fogaras	176	Nagyszeben	253
Giurgiu	77	Temesvár	329
Hirsova	151	Csalános	80
Iasi	226	Vaslui	199
Lugos	244	Nagyzerénd	374



Keresési fa



<u>A gráf csomópontjainak adatszerkezete (pl.)</u>:

1. **szülő csomópont**
2. **az állapottérnek a csomóponthoz tartozó állapota**
3. **a csomópontig tartó útköltség *g*(*n*)**
4. **a hátralévő legrövidebb út optimista becslése *h*(*n*)**

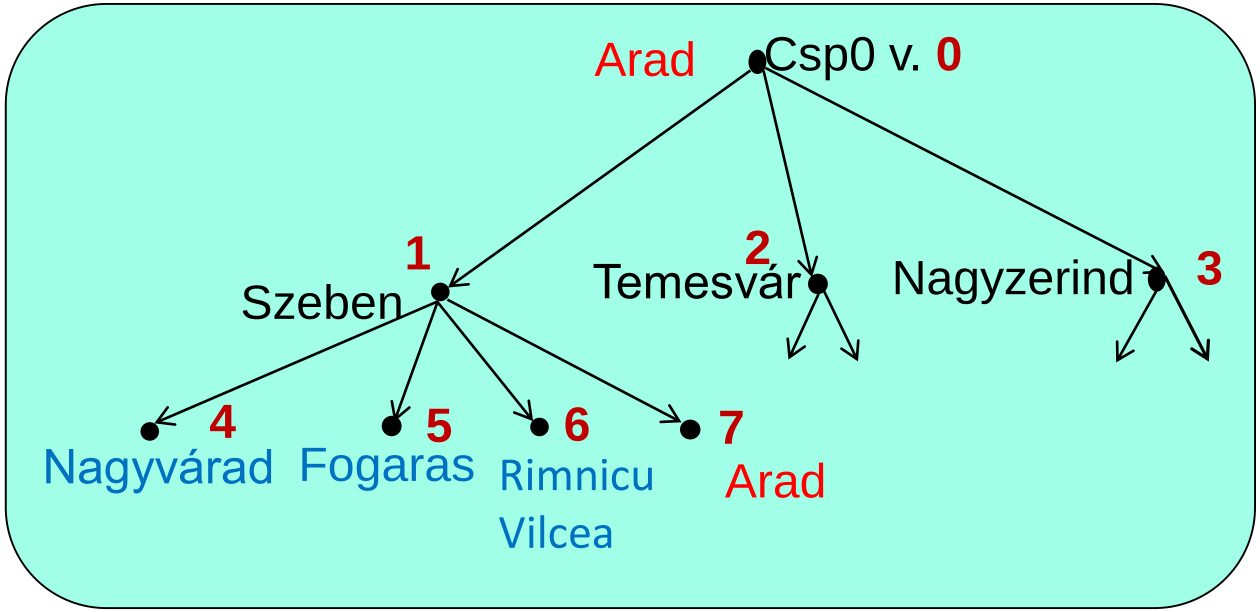
(-,csp0,Arad,0,366)
(csp0,csp1,Szeben,140,253)
(csp0,csp2,Temesvár,118,329)
(csp0,csp3,Nagyzerind,75,374)
(csp1,csp4,Nagyvárad,146,380)
(csp1,csp5,Fogaras,239,176)
(csp1,csp6,Rimnicu Vilcea,220,193)
(csp1,csp7,Arad,280,366)
....

Mohó keresés

oL rendezése itt: a ***h(n)* heurisztika *növekvő* sorrendjében**.

0. Iniciálás:

- cL={} üres
- oL={Csp0}



1. Vesszük az oL első elemét, célvizsgálat. Ha cél, készen vagyunk, ha nem cél, kifejtjük, majd a keletkező gyermek csomópontokat felvesszük oL-re, a ***hátralévő úthossz h(n) becslésének növekvő sorrendjében***. A vizsgált első elem cL-be.

oL={Csp1, Csp2, Csp3}, cL={Csp0}

h(n) 253 329 374

Arad	366	Mehadia	241
Bukarest	0	Neamt	234
Craiova	160	Nagyvárad	380
Dobreta	242	Pitești	100
Eforie	161	Rimniu Vilcea	193
Fogaras	176	Nagyszeben	253
Giurgiu	77	Temesvár	329
Hirsova	151	Csalános	80
Iasi	226	Vaslui	199
Lugos	244	Nagyzerénd	374

2. oL első eleme – cél?. Ha nem, kifejtjük - gyermek csomópontok ***h(n) becslésének növekvő sorrendjében*** oL-be. A vizsgált első elem oL→ cL.

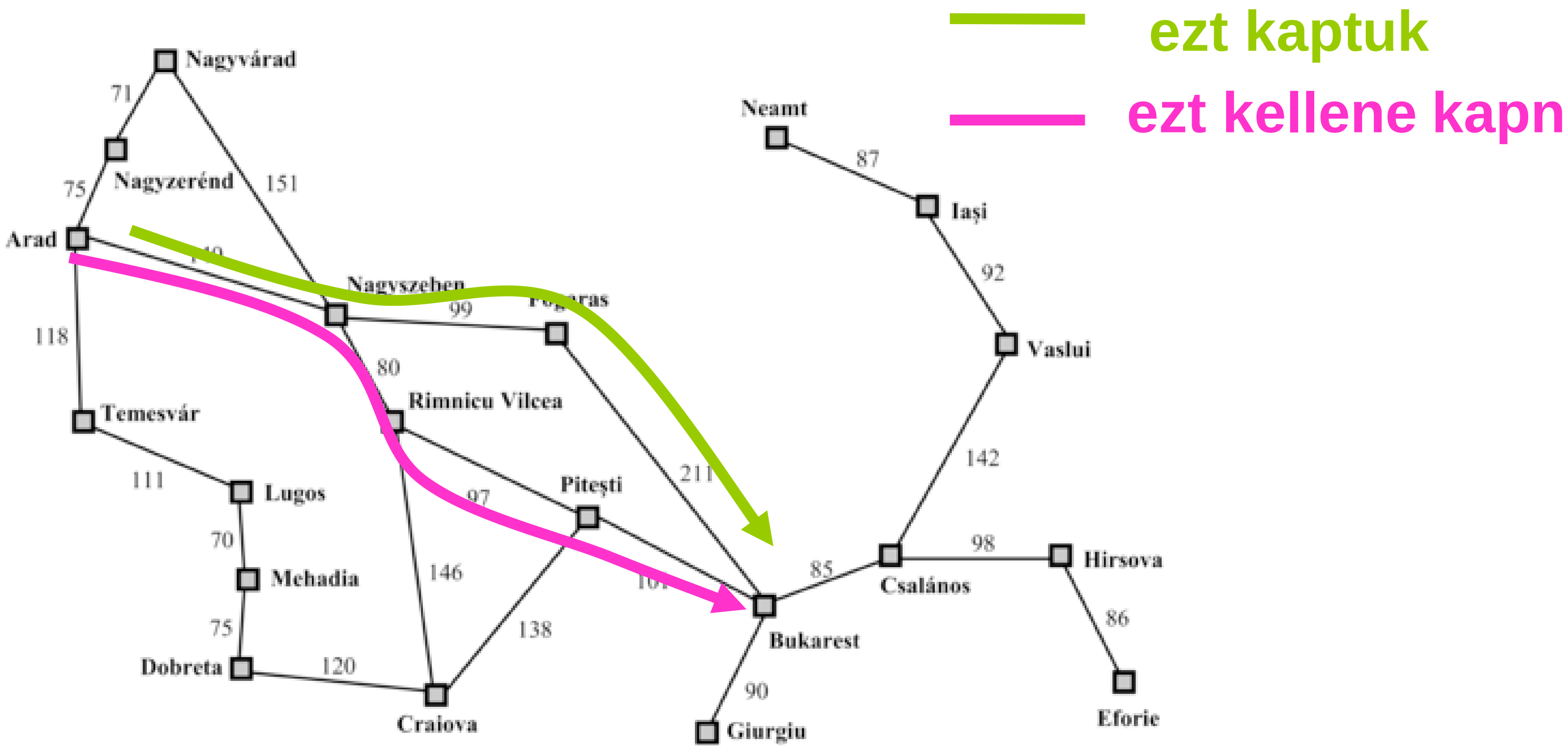
oL={Csp5, Csp6, Csp2, Csp3, Csp7, Csp4}, cL={Csp0, Csp1}

h(n) 176 193 329 374 366 380

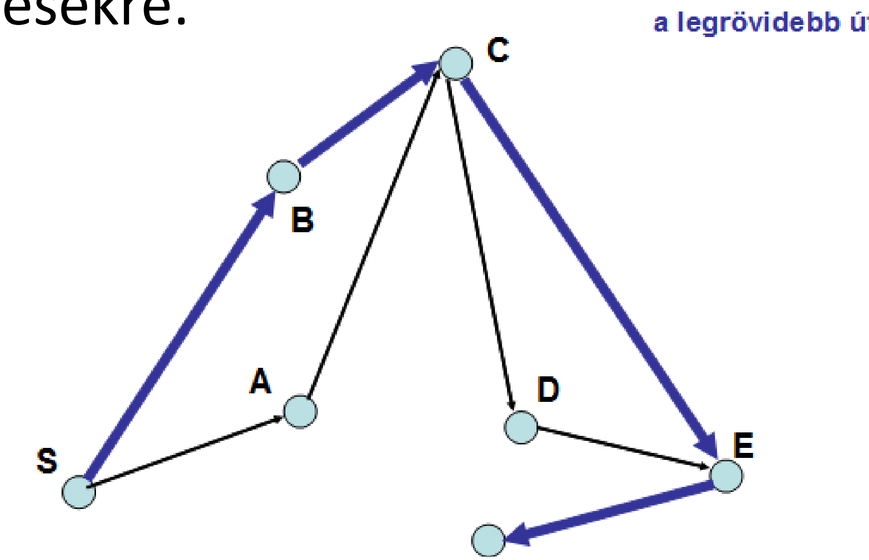
Mohó keresés:

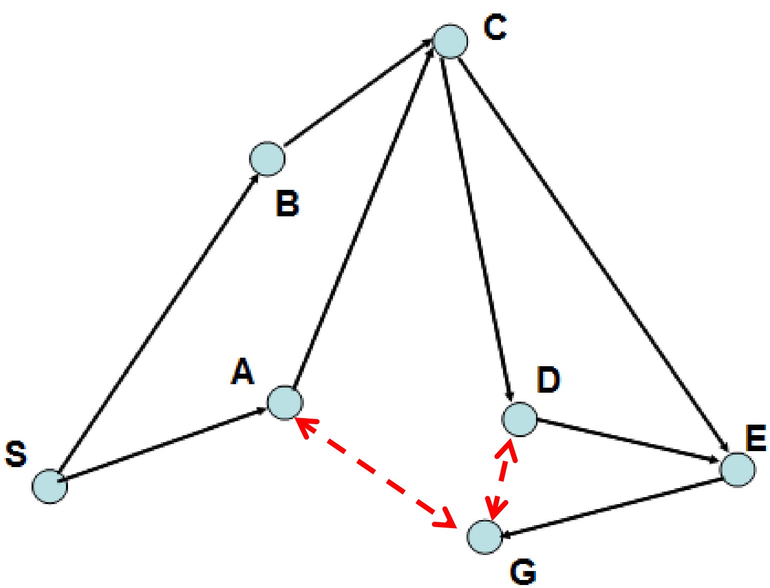
a célhoz *legközelebbinek tűnő* csomópont először

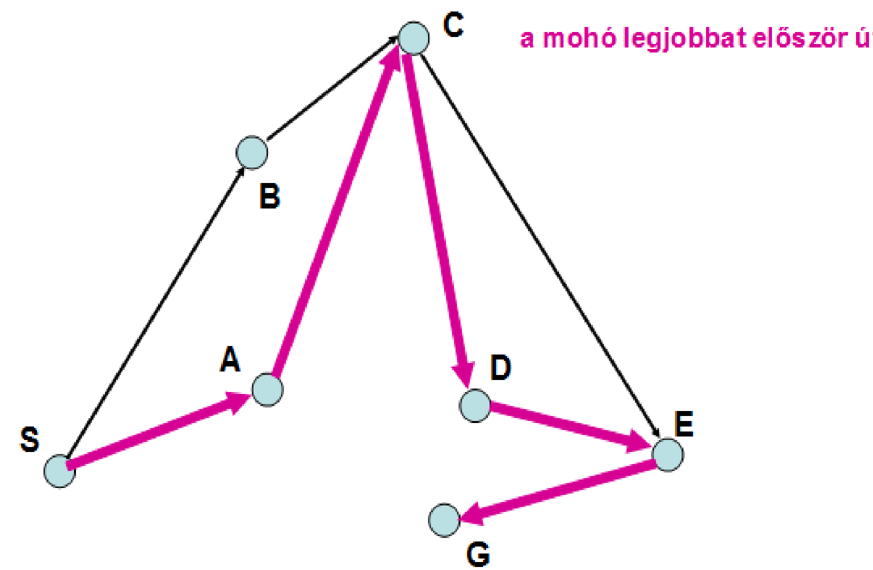
<u>Stratégia</u>: a következő lépésben azt a csp-t fejti ki, amelyhez rendelt problémaállapotot **a legközelebbinek ítéli a célállapothoz** (legkisebb az *n*-edik csp-hoz rendelt *h*(*n*) heurisztikus érték).



Mohó algoritmus általában **gyorsan** megtalálja a megoldást, de **nem mindig az optimális** megoldást találja meg. A mohó keresés érzékeny a hibás kezdőlépésekre.







Probléma: A és D légvonalban nagyon közel van a G célhoz, de az úthálózaton nem

Mohó keresés

mélységi keresésre hasonlít, egyetlen út végigkövetését preferálja a célig, zsákutcából visszalép.

Ua. a problémák, mint a mélységi keresésnél:

- nem optimális,
- nem teljes

Az összes csomópontot a memóriában tartja: ezért a legrosszabb (worst-case) idő- és tárigény: $O(b^m)$ – ha m a problématér mélysége

Jó heurisztikus függvény:

- a tár- és időigény jelentősen csökkenthető,
- a csökkenés az adott problémától és a h minőségétől függ.

A mohó keresés ígéretes, egyszerű, sokszor akár használják is, de mégsem igazán jó.

A* keresés: a teljes útköltség minimalizálása

(de facto standard az útkeresésben)

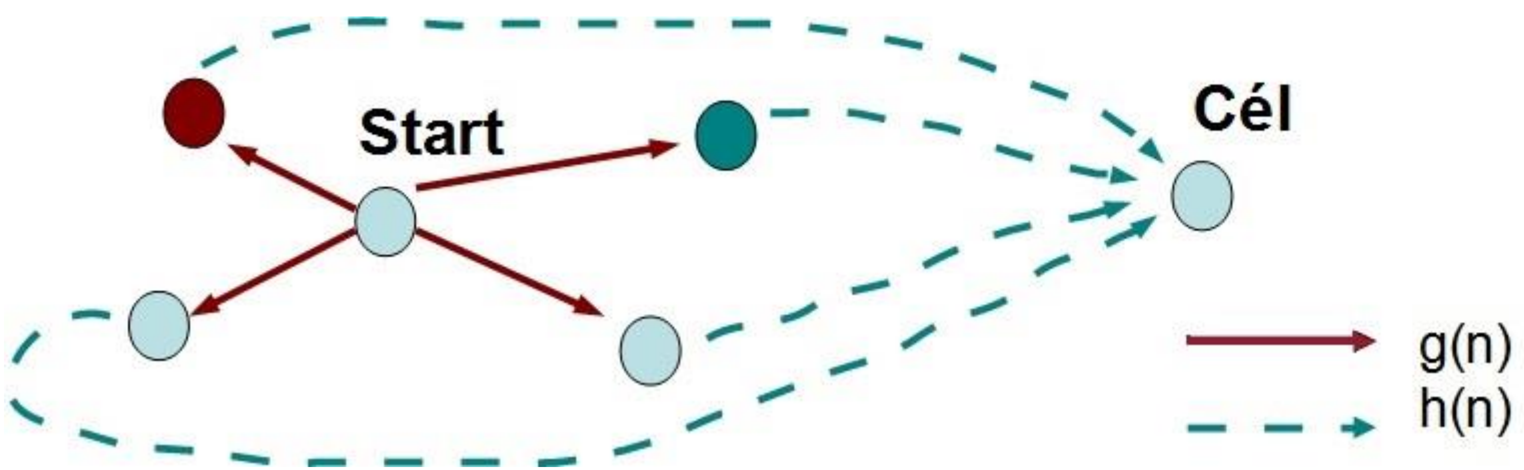
mohó: **min {*h*(*n*)}** a célhoz vezető útköltség becslőjét minimalizálja
nem teljes (nem optimális)

egyenletes költségű: **min {*g*(*n*)}** a megtett út költségét minimalizálja
optimális (egyben **teljes** is), de nagyon **rossz hatékonyságú**

a két stratégia ötvözése: **min {*f*(*n*)} = min {*h*(*n*) + *g*(*n*)}**

***g*(*n*)**: a kiinduló cs-ponttól az *n* cs-pontig számított út **tényleges** költsége
***h*(*n*)**: az *n* cs-ponttól a célba vezető legolcsóbb költségű út **becsült** értéke

***f*(*n*)=g(n)+h(n)**: a kiinduló csp-tól az adott *n*-edik csp-on át a célba vezető legolcsóbb költségű út **becsült** értéke



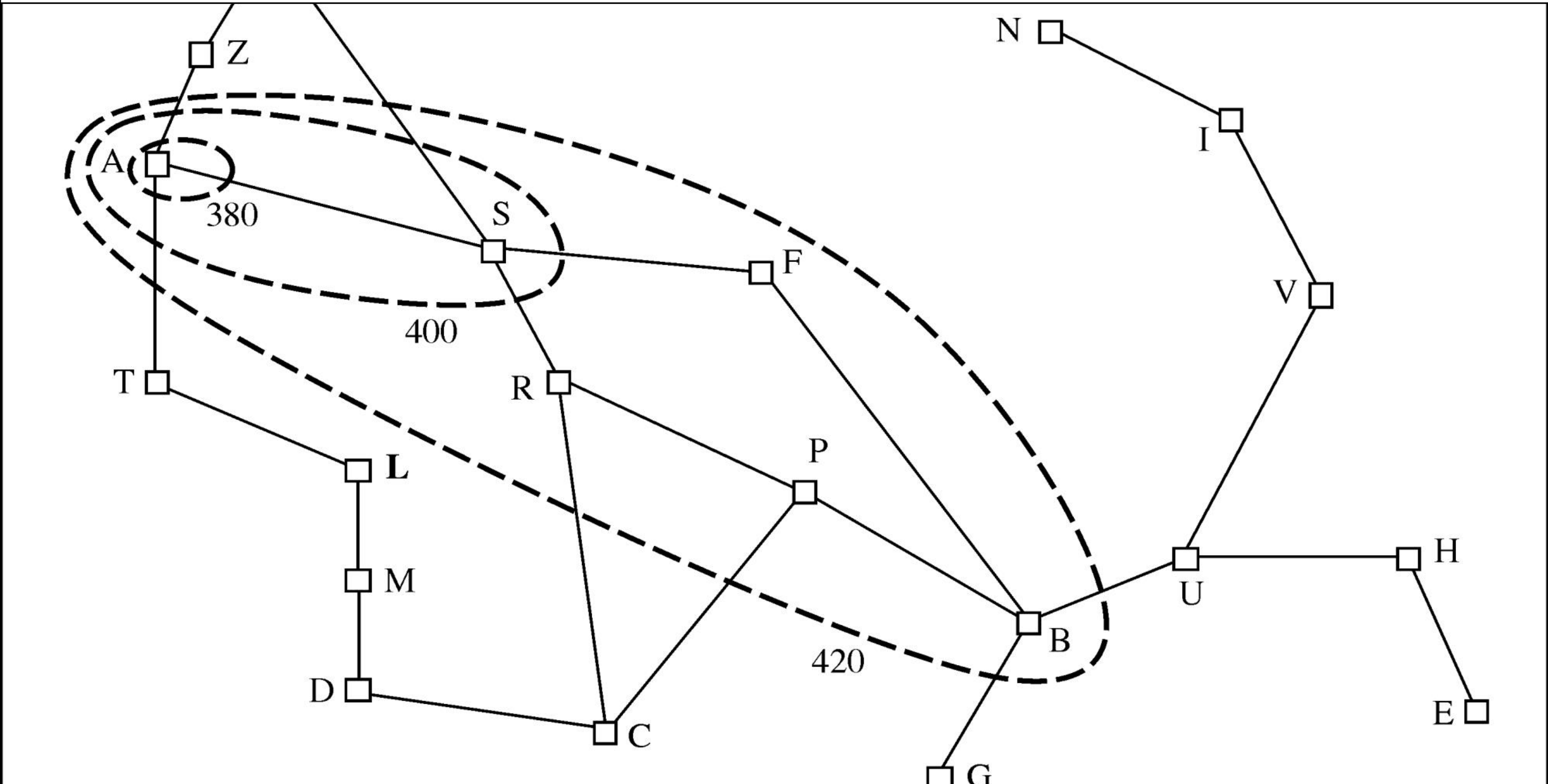
Ha ***h(n)*** elfogadható, akkor ***f****(n)* sohasem becsüli túl az ***n***-edik csomóponton át vezető legjobb megoldás valódi költségét.

A* keresés:

az ***f(n)=g(n)+h(n)*** függvényt alkalmazó legjobbat-először keresés, ahol ***h*** elfogadható heurisztika

Az A* **teljes** és **optimális!**

Egyenletes költségű keresés: olyan A* keresés, amelynél ***h(n)*** = 0 minden csomópontra (ezért *g*(*n*) szerint alakítjuk a sorrendet). Így a vizsgálandó csp-sávok a kiinduló csp. köré húzott többé-kevésbé koncentrikus „körök”. Jó heurisztikus függvény esetén: az A* keresésnél a sávok a **cél-állapot felé nyúlnak, fókuszálódnak az optimális út körül.**



http://mialmanach.mit.bme.hu/demonstraciok/a_kereses_usa_terkepen_konturkovetessel

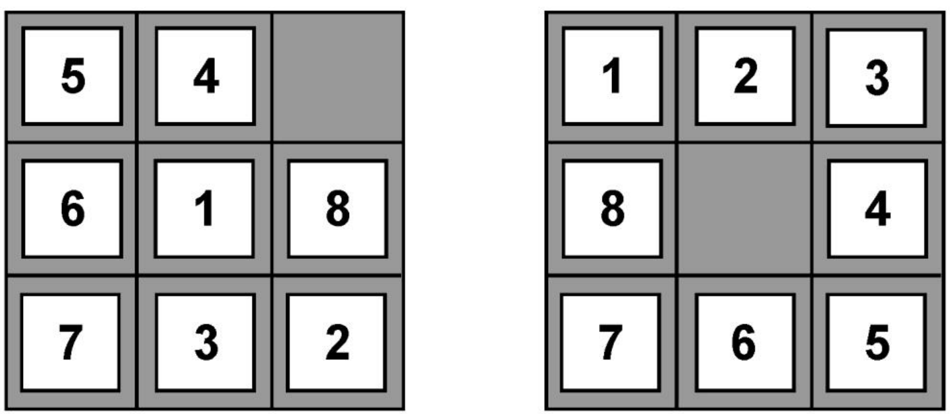
Heurisztikus függvények létrehozása

Jól bemutatható egy demóproblémán, a 8-as tili-toli logikai játékon. A logikai játék szabályai:

- Csak a szomszédos mezőre lehet lépni egy lapkával,
- Csak akkor lehet a szomszédos mezőre lépni, ha az üres
- A célállapotot a lenti jobboldali ábra mutatja

Elfogadható heurisztikus függvény kell, mert:

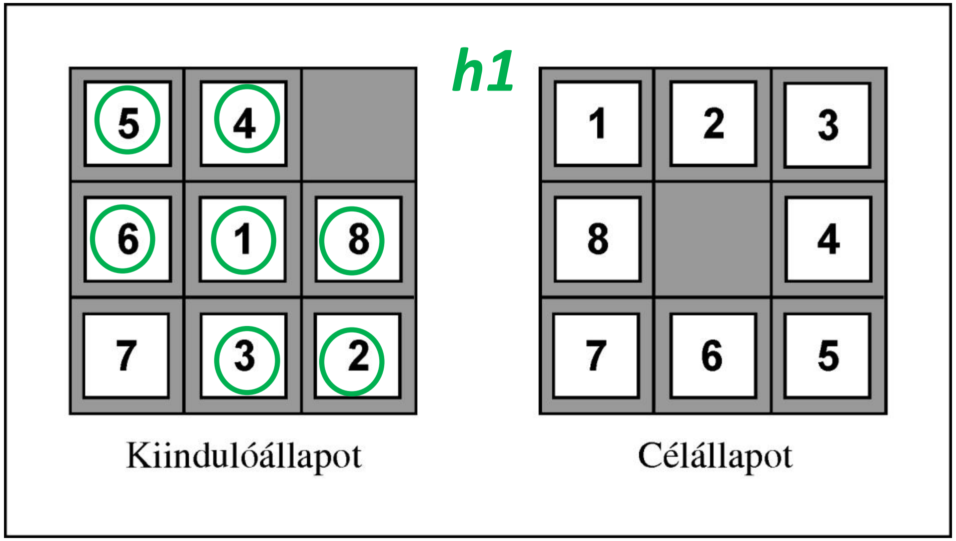
- kb. 20 lépésben lehet átlagosan megoldani,
- $b \sim 3$,
- kimerítő keresés $\Rightarrow$ kb. $3^{20} \approx 3{,}5 \cdot 10^9$ állapot

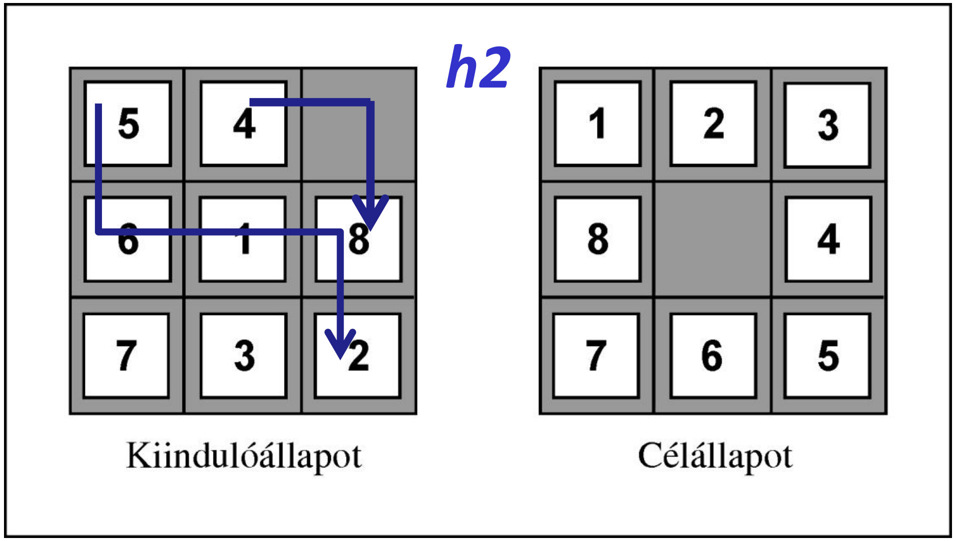


Kiindulóállapot

Célállapot

Heurisztikus függvények létrehozása





h*1* = **a rossz helyen lévő lapkák száma.**

elfogadható: minden rossz helyen lévő lapkát legalább egyszer mozgatni kell (az ábrán látható állapotra értéke 7)

h*2* =**a lapkák céltól mért** (vízsz. és függ.) **távolságainak összege**

háztömb- vagy Manhattan-távolság

(az ábrán látható állapotra értéke 4+2+2+2+2+0+3+3=18)

Heurisztikus függvények létrehozása

Gyakran egy ún**. relaxált probléma** pontos megoldása adja a heurisztikát. Az eredeti probléma nem minden kötöttségét tartjuk meg.

Pl.

1. Az autóval eljussunk A-ból B-be (Arad→Bukarest) kötöttsége, hogy csak autóúton mehetünk. A légvonalbeli távolság heurisztikát úgy kapjuk, hogy ezt a kötöttséget figyelmen kívül hagyjuk, és mehetünk árkon-bokron át egyenesen (vagy az autónk tud repülni).
2. Az előző dián látható h1 heurisztika: mintha nem csak a szomszédos mezőre léphetnénk, és léphetnénk olyan mezőre is, ahol már van egy másik lapka.

   A h2 heurisztika kevesebb szabályt töröl el: itt már csak szomszédos mezőre léphetünk, de itt is megengedjük, hogy olyan mezőkön is átgázoljunk, amelyeken van lapka. (Nem csak üres mezőre léphetünk.)

   Mivel a problémán könnyítettünk (relaxáltuk), ezért a heurisztika optimista lesz, nem lesz nagyobb a költség az eredetinél!

Heurisztikus függvény: pontosság és hatékonyság

a ***h1, h2*** heurisztikus függvények tesztelése:

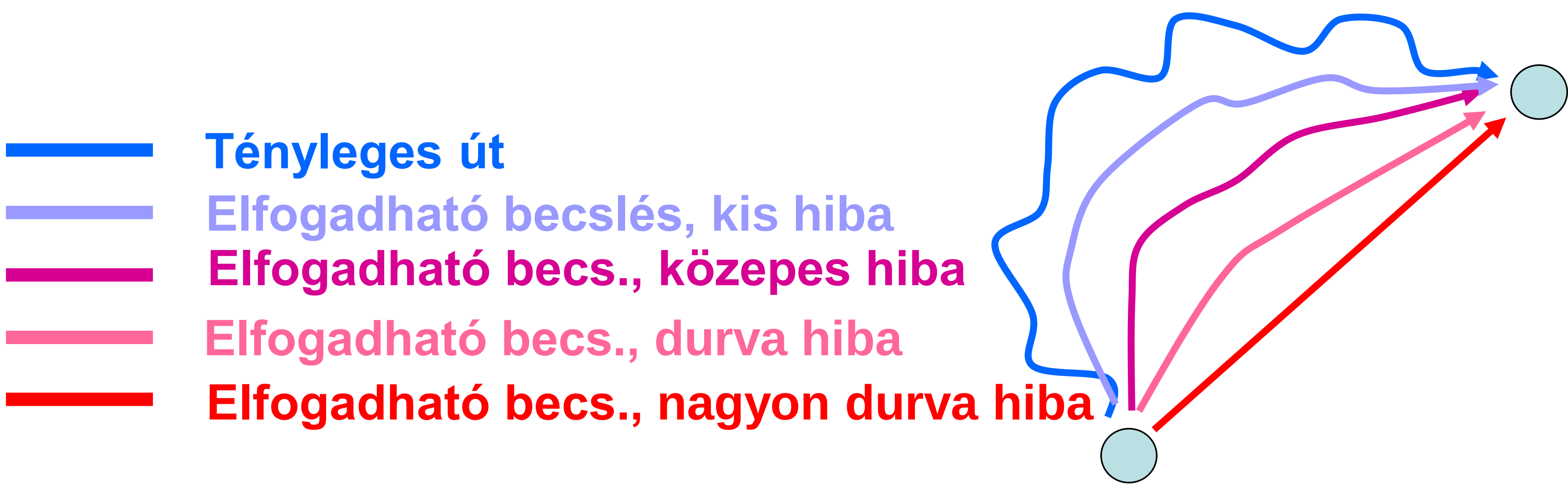
vajon a ***h2*** mindig jobb-e, mint a ***h1***? **Igen** (miért?).

minden *n* csp-ra ***h2* (*n*) ≥*h1* (*n*)**:

h2 **dominálja** ***h1***–et, **dominancia ⇒ hatékonyság**

a ***h2***-t használó A* *kevesebb csp-t fog kifejteni, mint a* ***h1***–et használó!

> **Mindig jobb, ha nagyobb értékeket adó heurisztikus függvényeket alkalmazunk, amíg nem becsüljük túl a valódi költséget**. (Persze, ha a költség nem negatív.)



Heurisztikus függvény: pontosság és hatékonysága

A heurisztikus függvények minősítése (általában nem könnyű feladat a minősítés, a megoldások összehasonlítása – skalár mérőszám kell!):

*b** **effektív elágazási tényező**

Ha az A* által kifejtett összes csomópont száma *N*, a megoldás mélysége *d*, akkor *b** annak a *d* mélységű kiegyensúlyozott fának az elágazási tényezője, amely *N* csomópontot tartalmazna:

$$N = 1 + b^* + (b^*)^2 + \ldots + (b^*)^d$$

(pl. 5 mélységben fekvő megoldás **52** cs-pont kifejtésével:

az effektív elágazási tényező 1.91

$1 + 1^1 + 1^2 + 1^3 + 1^4 + 1^5 = 6 < \mathbf{52} < 63 = 1 + 2 + 2^2 + 2^3 + 2^4 + 2^5$)

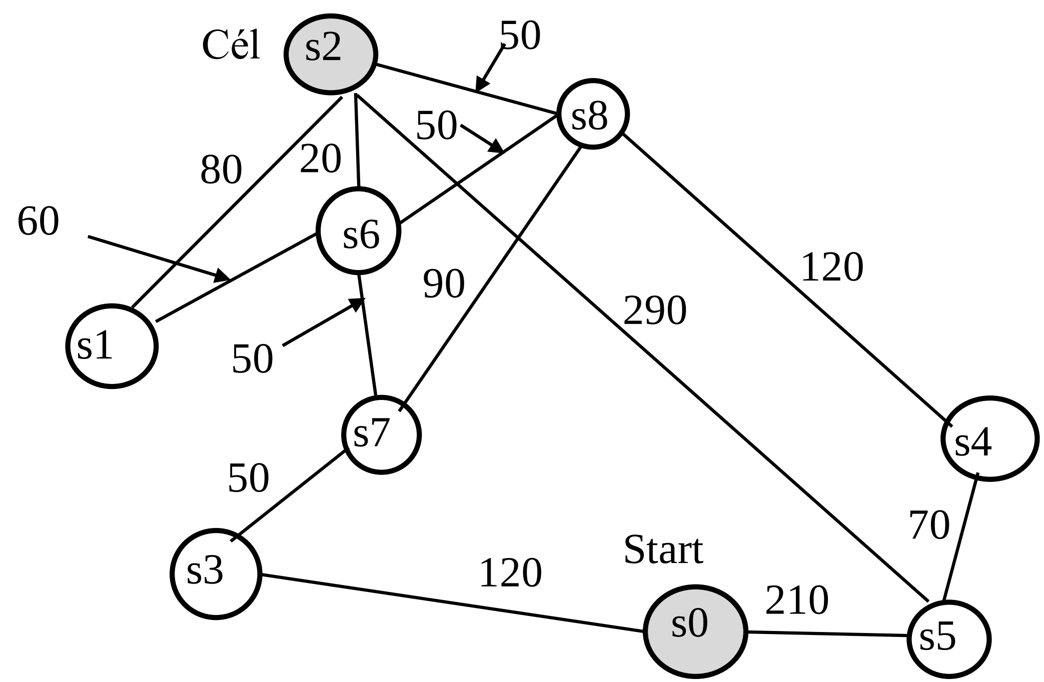
- Egy adott heurisztikus függvény által generált fa effektív elágazási tényezője általában nagyjából állandó egy adott problémaosztály számos egyedére.
- A *b** kis számú problémahalmazon végzett kísérleti mérése rendszerint jó becslés.
- Egy jól megtervezett heurisztikus függvény effektív elágazási tényezője **1 körüli érték.** (Ideális lenne az 1.)

h1 és ***h2*** tesztelése (8-as kirakójáték) : 100-100 random problémapéldány, 2, 4, …, 24 mélységű megoldással: A*, illetve a neminformált iteratívan mélyülő keresés (IMK)

	Keres. ktg. (Kifejtett csp.-ok száma)			b*		
d	IMK	A*(*h1*)	A*(*h2*)	IMK	A*(*h1*)	A*(*h2*)
2	10	6	6	2.45	1.79	1.79
4	112	13	12	2.87	1.48	1.45
6	680	20	18	2.73	1.34	1.30
8	6384	39	25	2.80	1.33	1.24
10	47127	93	39	2.79	1.38	1.22
12	364404	227	73	2.78	1.42	1.24
14	3473941	539	113	2.83	1.44	1.23
16	-	1301	211	-	1.45	1.25
18	-	3056	363	-	1.46	1.26
20	-	7276	676	-	1.47	1.27
22	-	18094	1219	-	1.48	1.28
24	-	39135	1641	-	1.48	1.26

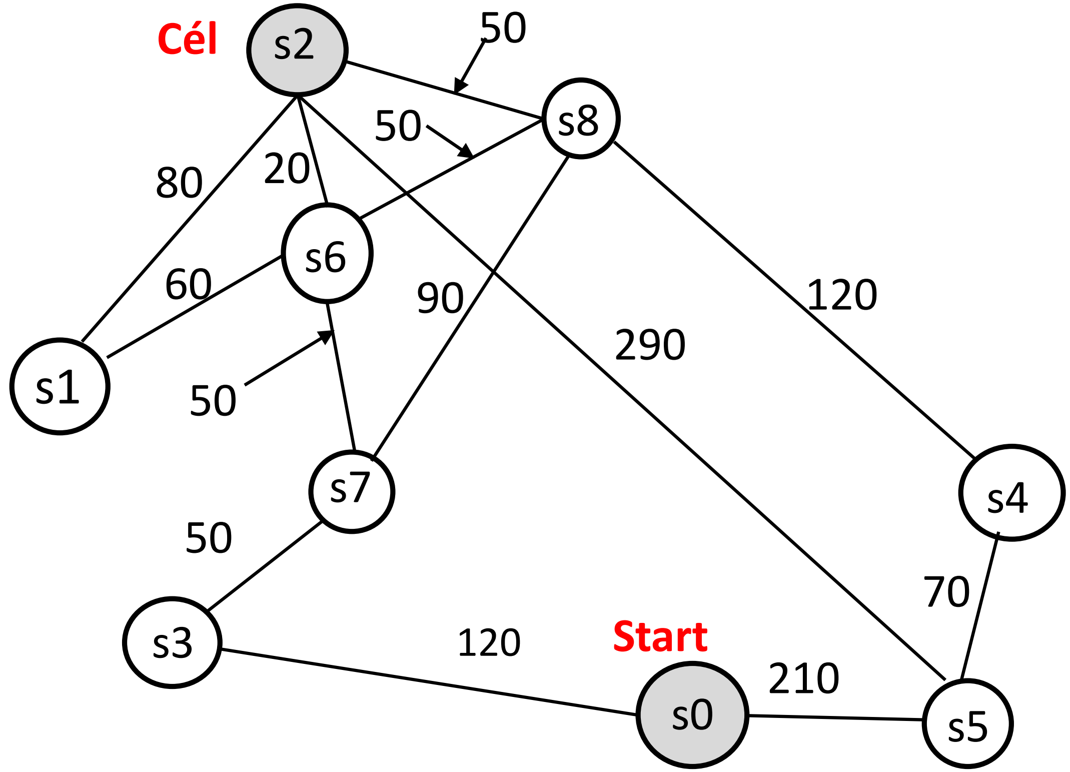
Régebbi zh példa (2 dián van): Az alábbi problémát A* kereséssel oldjuk meg, nem lépünk vissza abba az állapotba, ahonnan érkeztünk. A mellékelt táblázat mutatja a heurisztikánk egyes állapotokhoz tartozó értékeit:

Állapot	h(*sn*)
s0	210
s1	70
s2	0
s3	110
s4	155
s5	285
s6	18
s7	65
s8	50



Az ágens két listát épít, az elsőben azok a csomópontok szerepelnek, amiket már kifejtett, a másodikban azok, amelyekhez már eljutott, de még nem fejtette ki ezeket. Mindegyik listaelem 5 mezőből épül fel:
(<u>szülőcsomópont</u>, <u>aktuális csomópont</u>, <u>állapot</u>, <u>eddig megtett út költsége</u>, <u>az akt. csomópont heurisztika értéke</u>) például a gyökércsomópontra: (-,cs0,s0,0,210).

Állapot	h(sn)
s0	210
s1	70
s2	0
s3	110
s4	155
s5	285
s6	18
s7	65
s8	50

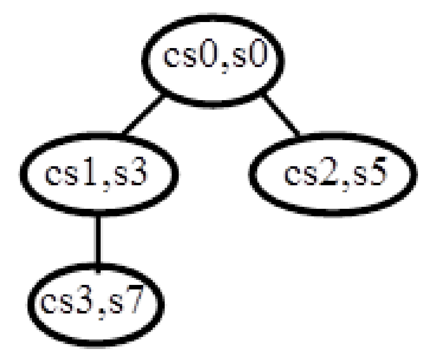


(szülőcsomópont, aktuális csomópont, állapot, eddig megtett út költsége, az akt. csomópont heurisztika értéke

A második lépés után a következők vannak a két listán

cL={(-,cs0,s0,0,210), (cs0,cs1,s3,120,110)}

oL={(cs1,cs3,s7,170,65),(cs0,cs2,s5,210,285)}



Adja meg a következő lépés után kialakuló két listát!