1 AZ MI FOGALMA

1.1 Miről ismerhető fel az MI?

Megoldandó feladat: nehéz
- A feladat problématere hatalmas
-Szisztematikus keresés helyett intuícióra, kreativitásra (azaz heurisztikára) van szükségünk ahhoz, hogy elkerljük a kombinatorikus robbanást.
Szoftver viselkedése: intelligens (tárol ismeretet, automatikusan következtet, tanul, gépi látás, gépi cselekvés)
-Turing teszt vs. kínai szoba elmélet
-"mesterjelölt szintű" mesterséges intelligencia
Felhasznált eszközök: sajátosak
-Átgondolt reprezentáció a feladat modellezéséhez
-Heurisztikával megerősített hatékony algoritmusok
-Gépi tanulás módszerei

2 MODELLEZÉS ÉS KERESÉS

feladat → útkeresési probléma → megoldás
Feladat és útkeresési probléma között helyezkedik el a modellezés. Itt található az állapottér-reprezentáció, a probléma dekompozíció, a korlátprogramozási modell, és a logikai reprezentáció.
Az útkeresési problémát egy gráffal reprezentálhatjuk. Az útkeresési probléma és a megoldás között található a keresés. A keresésbe tartoznak a lokális keresések, a visszalépéses keresések, a gráfkeresések, az evolúciós algoritmus, a rezolúció és a szabályalapú következtetés.

2.1 Mire kell a modellezésnek fókuszálni

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

2.2 Útkeresési probléma

Egy útkeresési problémában a problématér elemeit egy olyan élsúlyozott irányított gráf csúcsai vagy útjai szimbolizálják, amelyik gráf nem feltétlenül véges, de a

csúcsainak kifoka véges, és van egy közös pozitív alsó korlátja az élek súlyának (költségének) (δ-gráf).
A megoldást ennek megfelelően vagy egy célcsúcs, vagy egy startcsúcsból célcsúcsba vezető út (esetleg a legolcsóbb ilyen) megtalálása szolgáltatja.
Számos olyan modellező módszert ismerünk, amely a kitűzött feladatot útkeresési problémává fogalmazza át.

2.3 Gráf fogalmak

-csúcsok, irányított élek:$\rightarrow$ N, A $\subseteq$ NxN
-él n-ből m-be $\rightarrow$ (n,m) $\in$ A (n,m $\in$ N)
-n utódai $\rightarrow$ γ (n) = { m $\in$ N | (n,m) $\in$ A }
-n szülei $\rightarrow$ π(n) $\in$ Π(n) = { m $\in$ N | (m,n) $\in$ A }
-irányított gráf $\rightarrow$ R=(N,A)
-véges sok kivezető él $\rightarrow$ | γ (n) | $< \infty$ ($\forall$ n $\in$ N)
-élköltség $\rightarrow$ c:A $\rightarrow$ $\mathbb{R}$
-δ tulajdonság ($\delta \in \mathbb{R}^+$) $\rightarrow$ c(n,m) $\geq \delta > 0$ ($\forall$(n,m)$\in$A)
-δ-gráf $\rightarrow$ δ-tulajdonságú, véges sok kivezető élű, élsúlyozott irányított gráf
-irányított út $\rightarrow$ $\alpha = (n,n_1),(n_1,n_2),...,(n_{k-1},m) = <n,n_1,n_2,...,n_{k-1},m>$ $n \rightarrow^{\alpha} m$, $n \rightarrow m$, $n \rightarrow M$ ($M \subseteq N$) $n \rightarrow m$, $n \rightarrow M$
-út hossza $\rightarrow$ az út éleinek száma: $|\alpha|$
-út költsége $\rightarrow$ $c(\alpha)=c^{\alpha}(n,m) := \sigma_{j=1..k} c(n_{j-1},n_j)$ ha $\alpha = <n=n_0,n_1,n_2,...,n_{k-1},m=n_k>$
-opt. költség $\rightarrow$ $c^*(n,m):=\min_{\alpha \in \{ n \rightarrow m \}} c^{\alpha}(n,m)$ {δ gráfokban ez végtelen sok út esetén is értelmes. Értéke ∞, ha nincs egy út se.}
-opt. költség2 $\rightarrow$ $c^*(n,M):=\min_{\alpha \in \{ n \rightarrow m \}} c^{\alpha}(n,M)$
-opt. költségű út $n \rightarrow^* m := min_c\{\alpha |\ \alpha \in \{n \rightarrow m\}\}$
-opt. költségű út $n \rightarrow^* M := min_c\{\alpha |\ \alpha \in \{n \rightarrow M\}\}$

2.4 Gráfreprezentáció fogalma

Minden útkeresési probléma rendelkezik egy (a probléma modellezéséből származó) gráfreprezentációval, ami egy (R,s,T) hármas, amelyben:
-R=(N,A,c) δ-gráf az ún. reprezentációs 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 Tegyik csúcsába vezető irányított út, vagy s-ből T egyik csúcsába vezető legolcsóbb irányított út)
Az útkeresési problémák megoldásához azok a reprezentációs gráfjainak nagy mérete miatt speciális (nem determinisztikus, heurisztikus) útkereső algoritmusokra van szükség, amelyek:
-a startcsúcsból indulnak, amely az első aktuális csúcs;
-minden lépésben nem-determinisztikus módon új aktuális csúcso(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.

2.5 Kereső rendszer (KR)

Procedure KR
1. ADAT:= kezdeti érték
2. while !terminálási feltétel(ADAT) loop
3. SELECT SZ FROM alkalmazható szabályok
4. ADAT := SZ(ADAT)
5. endloop
end
AHOL:
-ADAT: globális munkaterület, tárolja a keresés során megszerzett és megőrzött ismeretet
-alkalmazható szabályok: keresési szabályok, megváltoztatják a globális munkaterület tartalmát
-SELECT: vezérlési stratégia, alkalmazható szabályok közül kiválaszt egy "megfelelőt"

2.6 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 programozási feladat megoldásához meg kell adnunk a feladat modelljét és készítenünk kell egy ehhez illeszkedő algoritmust, amely a feladat megoldását előállítja.
-Gépi tanulással a modell (reprezentáció és/vagy heurisztika), illetve a megoldó algoritmus (többnyire annak bizonyos paraméterei) állhatnak elő automatikusan.
-A tanuláshoz a megoldandó probléma néhány konkrét esetére, a tanító példákra van szükség.
-A gépi tanulási módszereket három csoportba szokás sorolni: felügyelt-, nemfelügyelt, és megerősítéses tanulásra attól függően, hogy a tanító példák input-output párok, csak inputok, vagy input-hasznosság párok.

4 Állapottér-reprezentáció

Állapottér: a probléma leírásához szükséges adatok által felvett érték-együttesek (azaz állapotok) halmaza
-az állapot többnyire egy összetett szerkezetű érték
-gyakran egy bővebb alaphalmazzal és egy azon értelmezett invariáns állítással definiáljuk
Múveletek: állapotból állapotba vezetnek.
-megadásukhoz: előfeltétel és hatás leírása
-invariáns tulajdonságot tartó leképezés
Kezdőállapot(ok) vagy azokat leíró kezdeti feltétel
Végállapot(ok) vagy célfeltétel

4.1 Hanoi tornyai probléma

Állapottér: AT=$\{1,2,3\}^n$
megjegyzés: a tömb i-dik eleme mutatja az i-dik korong rúdjának számát, a korongok a rudakon méretük szerint fentről lefelé növekvő sorban vannak.
Művelet:Rak(honnan,hova): AT→AT honnan, hova ∈ {1,2,3}
HA a honnan és hova létezik és nem azonos, és van korong a honnan rúdon, és a hova rúd üres vagy a mozgatandó korong (honnan rúd felső korongja) kisebb, mint a hova rúd felső korongja, AKKOR this[honnan legfelső korongja] := hova

4.2 Állapottér-reprezentáció gráf-reprezentációja

δ-gráf állapot-gráf
-csúcs: állapot
-irányított él: művelet hatása
-élköltség: művelet költsége
startcsúcs kezdőállapot
célcsúcsok végállapotok
irányított út egy műveletsorozat hatása

4.3 Állapottér vs. problématér

-Az állapottér-reprezentáció és a problématér között szoros kapcsolat áll fenn, de az állapottér többnyire nem azonos a problématérrel.
-A problématér elemeit többnyire nem az állapotok, hanem a startcsúcsból induló különböző hosszúságú irányított utak.
-A hanoi tornyai problémánál például egy megoldást egy irányított út szimbolizál, amelyik a startcsúcsból a célcsúcsba vezet.
-Van amikor a megoldás egyetlen álapot (azaz csúcs), de ebben az esetben is kell találni egy odavezető operátor-sorozatot (azaz irányított utat).

4.4 Állapot-gráf bonyolultsága

Állapot-gráf bonyolultsága → Problématér mérete → Keresés számításigénye

A bonyolultság elsősorban a start csúcsból kivezető utak száma az oda-vissza lépések nélkül, amely nyilván függvénye a
-csúcsok és élek számának
-csúcsok ki-fokának
-körök gyakoriságának, és hosszuk sokféleségének

Ugyanannak a feladatnak több modellje lehet: érdemes olyat keresni, amely kisebb problémateret jelöl ki.
-Az előző reprezentációnál a problématér mérete, azaz a lehetséges utak száma, óriási. Készítsünk jobb modellt!
-Bővítsük az állapotteret, és használjunk új műveletet!
-Műveletek előfeltételének szigorításával csökken az állapot-gráf átlagos ki-foka.

4.5 Művelet végrehajtásának hatékonysága

A művelet kiszámítási bonyolultsága csökkenthető, ha az állapotokat extra információval egészítjük ki, vagy az invariáns szigorításával szűkítjük az állapotteret.

4.6 Hogyan "látja" egy keresés a reprezentációs gráfot?

Egy keresés fokozatosan fedezi fel a reprezentációs gráfot: bizonyos részeihez soha nem jut el, de a felfedezett részt sem feltétlenül tárolja el teljesen, sőt, sokszor torzultan "látja" azt: ha például egy csúcshoz érve nem vizsgálja meg, hogy ezt korábban már felfedezte-e, hanem új csúcsként regisztrálja, akkor az eredeti gráf helyett egy fát fog tárolni.

4.7 Reprezentációs gráf "fává egyenesítése"

Ha a keresés nem vizsgálja meg egy csúcsról, hogy korábban már felfedezte-e, akkor az eredeti reprezentációs gráf helyett annak fává kiegyenesített változatában keres.
Előny: eltűnnek a körök, de a megoldási utak megmaradnak.
Hátrány: duplikátumok jelennek meg, sőt a körök kiegyenesítése végtelen hosszú utakat eredményez.
A kétirányú (oda-vissza) élek szörnyen megnövelik a kiegyenesítéssel kapott fa méretét. De bármelyik keresésnél eltárolhatjuk egy csúcsnak azt a szülőcsúcsát, amelyik felől a csúcsot elértük. Így egy csúcsból a szülőjébe visszavezető él könnyen felismerhető és figyelmen kívül hagyható.

5 Probléma dekompozíció

Egy probléma dekomponálása során a problémát részproblémákra bontjuk, majd azokat tovább részletezzük, amíg nyilvánvalóan megoldható problémákat nem kapunk.
Sokszor egy probléma megoldását akár többféleképpen is fel lehet bontani részproblémák megoldásaira.

5.1 Dekompozíciós reprezentáció fogalma

A reprezentációhoz meg kell adnunk:
-a feladat részproblémáinak általános leírását
-a kiinduló problémát
-az egyszerű problémákat, amelyekről könnyen eldönthető, hogy megoldhatók-e vagy sem
-a dekomponáló műveleteket:
D: probléma $\rightarrow$ probléma$^+$
D(p)= $<p_1,...,p_n>$

5.2 A dekompozíció modellezése ÉS/VAGY gráffal

Egy dekompozíciót egy ún. ÉS/VAGY gráffal szemléltetjük: -egy csúcs egy részproblémát jelöl, a startcsúcs a kiinduló problémát, a célcsúcsok a megoldható egyszerű problémákat. -egy élköteg egy dekomponáló művelet hatását írja le, és a dekomponált probléma csúcsából a dekomponálással előállított részproblémák csúcsaiba vezet.

- egy élköteg élei mutatják meg, hogy a dekomponált probléma megoldásához mely részproblémákat kell megoldani. Az élköteg élei között ezért ún. "ÉS" kapcsolat van: hiszen minden részproblémát meg kell oldani.
- egy csúcsból több élköteg is indulhat, ha egy probléma többféleképpen dekomponálható. Ezen élkötegek élei között ún. "VAGY" kapcsolat áll fenn: hiszen választhatunk, hogy melyik élköteg mentén oldjunk meg egy problémát.

5.3 ÉS/VAGY gráfok

1. AZ R=(N,A) élsúlyozott irányított hiper-gráf, ahol az
   -N a csúcsok halmaza
   -A $\subseteq$ { (n,M)$\in$ NxN$^+$ | $0 \neq |M| < \infty$ } a hiper-élek halmaza, |M| a hiper-él rendje
   -(c(n,M) az (n,M) költsége)
2. Egy csúcsból véges sok hiper-él indulhat
3. $(0<\delta \leq c(n,M))$