

FONAMENTS DE LA PLANIFICACIÓ

- Què és
- Exemples
- De la resolució de problemes a la planificació
- Terminologia
- Un agent planificador simple
- Aproximacions

Què és

- Planificar: decidir sobre la línia d'acció abans d'actuar
- Problema: dissenyar un sistema que pugui descriure un conjunt d'accions (o pla) que permeti aconseguir un objectiu desitjat
- Aquest conjunt d'accions es passa a un robot, un sistema de producció (fàbrica) o algun tipus d'efector que, seguint el pla, produeix els resultats desitjats
- El disseny de planificadors
 - ← S'enriqueix de tècniques d'IA
 - Introdueix nous problemes en el camp de la IA com:
 - Representació i raonament del temps, causalitats, i intencions
 - Restriccions físiques
 - Incertesa en l'execució dels plans
 - Sensibilitat i percepció sobre el món real i creences sobre ell
 - Molts agents que cooperen (o interfereixen)

Què és

- Pla: línia d'acció
 - Pot ser ordenat
 - Ex. Les persones solen vestir-se abans d'anar al teatre
 - Pot ser desordenat
 - Ex. La llista de la compra.
- Molts plans tenen passos vagues que requereixen d'una posterior especialització.



Exemples

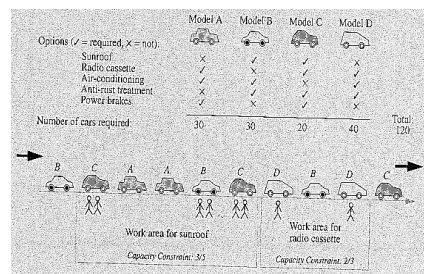
- Planificació d'un viatge
 - Requeriments: llocs de visita: visitar el Louvre, la Torre Eiffel, Notre Dame, ...
 - Opcions: bitllet d'avió, taxis, metros, lloguer de cotxer, ponts aeris
- ⇒ Planificar: trobar un itinerari que passi per tots els llocs que es volen visitar
- ⇒ Quines accions.

Exemples

- Scheduling (planificació del temps i dels recursos)
 - Restriccions:
 - Temporals: no marxar de casa abans de diumenge, no tornar més tard de dissabte
 - Cost: no superar les 100.000 ptes.
 - Temps meteorològic: quan faci bon temps
 - Exemple canònic: job-shop (planificació de les tasques d'un taller).
 - Conjunt de treballs: cada treball és una col·lecció de tasques parcialment ordenades, d'una determinada duració
 - Conjunt de màquines: cada màquina és capaç de dur a terme un subconjunt de totes les tasques possibles
 - Solució: assignar les màquines per intervals de temps, de manera que cap màquina estigui fent més d'una tasca al mateix temps i que cada tasca es completi abans de començar una altra que la segueixi en un ordre parcial establert.
 - Satisfacció vs. Optimització.

⇒ Quan.

Exemples



Exemples

- Grafs: (ex. El problema del viatjant de comerç): la solució que ofereix la IA és tenir en compte altres interaccions com:
 - Si sempre usem la mateixa línia aèria, és més barat
 - Viatjar en dissabte és més econòmic.
- Quant a la investigació operativa, aquesta treballa en variables, mentre que en IA el coneixement del domini és més ric, hi ha una dinàmica més complexa, un llenguatge més expressiu.

De la resolució de problemes a la planificació

- Beneficis de la planificació en la resolució de problemes:
 - Reduir la cerca
 - En cerca heurística anem aplicant operadors successivament.
 - En planificació:
 - Representem explícitament els objectius
 - » Podem dividir un problema en parts (divide & conquer, subplans)
 - » Podem relaxar els requeriments per construir solucions de manera successiva.
 - Resoldre conflictes entre objectius
 - Conflictes entre objectius
 - Els plans poden fallar més sovint que en la resolució de problemes
 - Ex.: Un pot anar a la biblioteca dos cops, perquè el pla de prendre un llibre en préstec i retornar un altre al mateix temps no s'ha pogut satisfer (conjunt independent d'objectius)
 - Ex. Construir una casa inclou els subobjectius d'instal·lar un radiador i la instal·lació elèctrica. Però primer s'ha de fer la instal·lació elèctrica (conjunt d'objectiu dependents). --> El planificador pot detectar aquesta situació abans d'actuar

De la resolució de problemes a la planificació

- (... beneficis...)
 - Proporcionar una base per la recuperació d'errors o fallides.
 - Els plans poden monitoritzar un procés de resolució de problemes i "caçar errors abans de que aquests causin cap mal".
 - Ex.:
 - » S'envia un vehicle a un planeta
 - » Aquest vehicle ha de ser capaç de planificar una ruta i re-planificar si les condicions canvien
 - » Hi ha un feedback (realimentació) de l'entorn cap el planificador que pot fer front a situacions inesperades

Terminologia

Què necessitem?

- Representacions obertes
 - Accions: descripcions lògiques de condicions i efectes
 - Permetre al planificador establir connexions entre estats i accions
 - La lògica de predicats és una solució teòrica
 - Inici: $En(casa, S0), \neg Hi_ha(llet, S0), \neg Hi_ha(plàtans, S0)$
 - Objectiu: $\exists s, En(casa, s), Hi_ha(llet, s), Hi_ha(plàtans, s)$
 - Operador: $\forall a, s \ T\epsilon(llet, resultat(a, s)) \Leftrightarrow [a = comprar(llet), En(supermercat, s), Hi_ha(llet, s), a \neq alluirar(llet)]$
- ... però
 - Temps exponencial a l'extensió de la solució
 - La inferència lògica és semidecidible
 - Plans irrelevantes
- ⇒ Restringir el llenguatge i usar algorismes especials (planificador) enlloc d'un demostrador de teoremes
- Incorporació d'accions sempre que sigui necessari
- Independència de les accions

Terminologia

El problema de la planificació

- Entrada
 - Una descripció del món en un llenguatge
 - Una descripció de l'objectiu de l'agent
 - Una descripció de les accions possibles (teoria del domini)
- Sortida
 - Seqüència d'accions.



- Generació de plans
- **Síntesi** de plans
- Construcció de plans

Terminologia

Llenguatge

- Hi ha efectes determinístics que es poden representar mitjançant càlcul de predicats.
 - Ex. $rm * (esborrar\ tots\ els\ fitxers)$
- Però hi ha que no
 - Ex.: l'efecte de subministrar un medicament a un pacient
- Cal fer una sèrie de suposicions per definir el llenguatge.
 - Temps atòmic:
 - L'execució d'una acció és indivisible i ininterrompible
 - 2 accions no es poden executar simultàniament
 - Ommiscència:
 - L'agent té un coneixement complet de l'estat inicial i la naturalesa de les seves accions
 - Única causa d'acció: el món només canvia per l'actuació de l'agent.

Terminologia

STRIPS: el llenguatge més simple

- STRIPS: planificador que controla el robot SHAKEY
- Representació basada en proposicions: (damunt A taula)
- Accions:
 - Condicions
 - Efectes: Add-delete lists.

Agafar(x)

Precondicions: sobreTaula(x) ∧ braçLliure ∧ clar(x)

Esborrar: sobreTaula(x)

braçLliure

clar(x)

Afegir: agafat(x)

- Suposició del món tancat: tot allò que no figura descrit, és fals.

Un agent planificador simple

funció agent_planificador_simple(percepció) retorna acció

estàtic: BC: base de coneixements
(inclou descripció de les accions)
p: pla (inicialment no_hi_ha_pla)
t: comptador de temps (inicialment 0)

variables locals:
G: objectius
actual: estat_actual

Un agent planificador simple

```
inici
actualitza(BC, informació(percepció,t))
actual:= descripció_estat(BC,t);
si p=no_hi_ha_pla
llavors G:= pregunta(BC,fer_consulta(t));
p:= planificador(actual,G,BC,pla);
fsi;
si p=pla_nul o p=no_hi_ha_pla
llavors acció:=no_op;
sinó acció:=primer(p);
p:=resta(p);
fsi;
actualitza(BC,informa(acció,t));
t:=t+1;
retorna(acció);
fi agent_planificador_simple;
```

Aproximacions segons la dependència amb el domini

- Planificadors independents del domini: s'espera que funcionin en una ampla gamma d'aplicacions
- Planificadors dependents del domini: Utilitza heurístiques pròpies d'un domini per controlar les operacions del planificador

STRIPS	Control de robots
HACKER	Generació de programes
NOAH	Supervisió d'aprenents d'enginyers mecànics
NONLIN	Revisió turbines elèctriques
ISIS	Producció de turbines
MOLGEN	Experiments en genètica molecular
SIPE	Planificació de missions aeronàutiques
NONLIN+	Logística Naval
DEVISER	Seqüenciació de missions
FORBIN	Control de fabricació
.....	

Aproximacions segons l'esforç inferencial

- Planificadors deliberatius
 - Quan un conjunt d'accions s'han d'executar d'acord amb un patró coherent
 - Quan les accions interaccionen
 - Conjunt d'accions complexes: un agent ha de dedicar n temps raonable per raonar sobre elles
 - Raonar sobre les accions
- Planificadors reactius
 - La millor acció es calcula a partir de l'instant actual