Online Replanning

Section 11.3.3
Outline

- Contingency planning vs. replanning
- Replanning agent algorithm
- Execution monitoring
- Continuous planning
- (Multiagent planning)
Contingency planning vs. replanning

- **Contingency planning**: prepare in advance. Useful when some conditions needed for the contingency plan can be gathered before execution.

- **Execution monitoring**: ignore contingencies during planning, then handle them as they arise. Useful when planning time is a concern: not everything can be planned for.

- Basic idea: handle execution time failures at execution time.
Repairing a plan

whole plan

plan

continuation

repair

Sec. 11.3.3 – p.4/18
Chair and table example

Init(Init\(\text{Color}(\text{Chair}, \text{Blue}) \land \text{Color Table}(\text{Green}) \land \text{ContainsColor}(\text{BC}, \text{Blue}) \land \text{PaintCan}(\text{BC}) \land \text{ContainsColor}(\text{RC}, \text{Red}) \land \text{PaintCan}(\text{RC}))

Goal(Goal(\text{Color}(\text{Chair}, x) \land \text{Color}(\text{Table}, x))

Action(\text{Paint}(\text{object}, \text{color}), \\
\text{Precond}: \text{HavePaint}(\text{color}) \\
\text{Effect}: \text{Color}(\text{object}, \text{color}))

Action(\text{Open}(\text{can}), \\
\text{Precond}: \text{PaintCan}(\text{can}) \land \text{ContainsColor}(\text{can}, \text{color}) \\
\text{Effect}: \text{HavePaint}(\text{color}))
Chair and table example (cont’d)

Whole plan: [Start; Open(BC); Paint(Table,Blue); Finish]

What to do when

- it notices a missed green spot on the table just before finishing
- the agent plans to paint both red and it opens the can of red paint and finds there is only enough paint for the chair.
function REPLANNING AGENT (percept) returns an action

static: $KB$, a knowledge base (includes action descriptions)

  $plan$, a plan, initially []

  whole-plan, a plan, initially []

  $goal$, a goal


TELL ($KB$, MAKE-PERCEPT-SENTENCE (percept, t))

current ← STATE-DESCRIPTION ($KB$, t)

if $plan$ = [] then

  whole-plan ← $plan$ ← PLANNER (current, goal, $KB$)

if PRECONDITIONS (FIRST ($plan$)) not currently true in $KB$ then

  candidates ← SORT (whole-plan, ordered by distance to current)

find state $s$ in candidates such that

  failure $\neq$ repair ← PLANNER (current, $s$, $KB$)

  continuation ← the tail of whole-plan starting at $s$

  whole-plan ← $plan$ ← APPEND (repair, continuation)

return POP ($plan$)
What to monitor, what to ignore

- **Action monitoring**: Check the preconditions of the next action to execute
- **Plan monitoring**: Check the preconditions of all the actions to execute
- Monitor a selected set based on priority
- Look for opportunities (**serendipity**)
Other important questions

- Which contingencies to plan for, which ones to leave until execution
- Should replanning be a plan step
- learning/modifying actions
- side note: “don’t touch” conditions
Fixing plan flaws continually

- **Missing goal**: adding new goals
- **Open precondition**: close using causal links (POP)
- **Causal conflict**: resolve threats (POP)
- **Unsupported link**: remove causal links supporting conditions that are no longer true
- **Redundant action**: remove actions that supply no causal links
- **Unexecuted action**: return an action that can be executed
- **Unnecessary historical goal**: if the current goal set has been achieved, remove them and allow for new goals
function \textsc{Continuous-POP-Agent}(\texttt{percept}) returns an action

\begin{itemize}
  \item \texttt{action} $\leftarrow$ \texttt{NoOp} (the default)
  \item \texttt{Effects}[\texttt{Start}] = \texttt{Update}(\texttt{Effects}[\texttt{Start}], \texttt{percept})
  \item \texttt{Remove-Flaw} (\texttt{plan})  // possibly updating action
\end{itemize}

return \texttt{action}
Example - start

Start

A B C D

B E C F

D G

Someone moved D

A B C D

B E C F

D G

dropped C

C A B

D E F G

tried again

Move(C,D)

Move(D,B)

Start

Finish

Start

A B C D

B E C F

D G

Ontable(A)

On(B,E)

On(C,F)

On(D,G)

Clear(A)

Clear(C)

Clear(D)

Clear(B)

On(C,D)

Clear(C)

Clear(D)

Clear(B)

Sec. 11.3.3 – p.12/18
Example - after D is moved onto B

Start Someone moved D dropped C tried again

Start

Move(D,B)

Move(C,D)

Finish
Example - Move(D,B) was redundant

Start

Someone moved D

dropped C

tried again

Start Someone moved D dropped C tried again

Ontable(A)
On(B,E)
On(C,F)
On(D,B)
Clear(A)
Clear(C)
Clear(D)
Clear(G)

Move(C,D)

On(C,F)
Clear(C)
Clear(D)

On(C,D)
On(D,B)

Finish

Sec. 11.3.3 – p.14/18
Example - Move(C,D) was executed

[Diagram showing the sequence of moves:

Start: A B C D E F G

Someone moved D and dropped C:

Start: A B C D E F G

Dropped C:

Start: C A D B E F G

Tried again:

Start: A B C D E F G

Instructions:

- Ontable(A)
- On(B,E)
- On(C,A)
- On(D,B)
- Clear(F)
- Clear(C)
- Clear(D)
- Clear(G)

Finish:

- On(C,D)
- On(D,B)
Example - put Move(C,D) back in

Start

Someone moved D

dropped C

tried again

Start

Start

Move(C,D)

Finish

Sec. 11.3.3 – p.16/18
Example - plan complete

Start

- Ontable(A)
- On(B,E)
- On(C,D)
- On(D,B)
- Clear(F)
- Clear(C)
- Clear(D)
- Clear(G)

Someone moved D

- Drop(C)

Dropped C

Tried again

Finish

Sec. 11.3.3 – p.17/18
Multiagent planning

- **Cooperation**: Joint goals and plans
- **Multibody planning**: Synchronization, joint actions, concurrent actions
- **Coordination mechanisms**: convention, social laws, emergent behavior, communication, plan recognition, joint intention
- **Competition**: agents with conflicting utility functions