Planning, Execution & Learning
1. Conditional Planning

Reid Simmons
Conditional Planning

- Create Branching Plans
  - Take *observations* into account when selecting actions

- Observations Used to Handle Uncertainty
  - Uncertainty arises from non-deterministic actions
  - Uncertainty arises from lack of knowledge

- Planners Differ With Respect To:
  - Representation of uncertainty (logic, probabilities)
  - Representation of plans (trees, graphs)
  - Representation of observations
  - Search control
**CNLP (Peot & Smith, 1992)**

- Extensions to SNLP to Create Conditional Plans with Observations

- Extensions to SNLP Representation
  - Three-valued logic (True, False, Unknown)
  - Observations actions
    - `Observe_Road (?loc1 ?loc2)`
      - Pre: Unknown (Clear(?loc1, ?loc2))
      - $+\alpha_1$: Clear(?loc1, ?loc2)
      - $+\alpha_2$: ~Clear(?loc1, ?loc2)
  - Contexts
    - Compatible observation labels
CNLP Extensions to SNLP

- “Conditioning”
  - Can remove threat by *separating contexts*
    (i.e., making them incompatible)

- Propagation of *context labels* and *reasons*
  - **Contexts**: What actions are incompatible
  - **Reasons**: what goals an action supports

- Tree-structured plan
  - Goal replication
Adding Conditional Operators

Start

Q, Unknown(P)

Q & Unknown(P)

Obs1

P α₁ α₂ ~P

P

A1

G r: Finish, Finish₂ c: {} c: {α₁}

G Finish c: {α₁}

~P

A2

G r: Finish₂ c: {α₂}

G Finish₂ c: {α₂}
Conditionally Planning a Ski Trip

Start
(at skis home), (at home), Unknown(Clear(b, s)), Unknown(Clear(c, p)),
Clear(home, b), Clear(b, c)

Get(skis)  Go(home, b)

Observe_road(b, s)
Clear(b, s)  ~Clear(b, s)

At(b)

Go(b, s)

Clear(b, s)

Go(c, p)

Observe_road(c, p)
Clear(c, p)

Fail

Finish
(c: {α₁})

Finish₂
(c: {α₂, β₁})

Start
(at skis home), (at home), Unknown(Clear(b, s)), Unknown(Clear(c, p)),
Clear(home, b), Clear(b, c)

Get(skis)  Go(home, b)

Observe_road(b, s)
Clear(b, s)  ~Clear(b, s)

Go(b, s)

Clear(b, s)

Go(c, p)

Observe_road(c, p)
Clear(c, p)

Fail

Finish
(c: {α₁})

Finish₂
(c: {α₂, β₁})
CNLP Summary

- Can Create Conditional Plans with Observation Actions
  - However, no explicit distinction between observations and causal effects

- Can Handle *Disjunctive Uncertainty*
  - No notion of which conditions more likely
  - Increases search space tremendously

- Can Plan with *Failure* as an Option
Buridan (Kushmerick, 1995)

- Plan to Achieve Goals with Probability Greater Than a Given Threshold

- Extensions to SNLP Representation
  - Probabilistic (not-deterministic) outcomes of actions
    - Conditioned on current state
    - Mutually exclusive and exhaustive “triggers”
  - No preconditions (!)
    - Action can occur anywhere

```
Pickup

Dry
  0.95 Holding
  0.05

~Dry
  0.5 Holding
  0.5
```
Buridan Extensions to SNLP

• Multiple Causal Links
  – Each link increases probability of achievement

• Confrontation
  – Reduce likelihood of threat by action $A_1$ by adding another action $A_2$ that makes it less likely for $A_1$ to have the undesired effect

• Plan Assessment
  – Estimate probability of plan success
  – NP-hard, in general
**Buridan (Partial) Plan**

Start

- Holding, Dry, Clean, ~Painted  
  p=0.7

- Holding, ~Dry, Clean, ~Painted  
  p=0.3

- Holding
  - Dry
    - Pickup
      - Holding
        - Paint
          - Holding
            - Painted
              - Holding
                - ~Clean
                  - Holding
                    - Painted
                      - Holding
                        - ~Clean
                          - Holding & Clean & Painted
                            Finish p > 0.85

Planning, Execution & Learning: Conditional  
Simmons, Veloso: Fall 2001
Increasing Probability of Success

~Holding, Dry, Clean, ~Painted

Dry

~Dry

1.0 0.8 0.2

Dry
~Dry

Pickup

Dry

~Dry

0.95 0.05 0.5 0.5

Holding

Dry

~Dry

1.0 0.8 0.2

Paint

Holding
~Holding

Painted
~Painted

Holding & Clean & Painted

Finish p > 0.85
Confronting a Threat

Start

~Holding, Dry, Clean, ~Painted

p = 0.7

~Holding, ~Dry, Clean, ~Painted

p = 0.3

Pickup

Dry

0.95

0.05

~Dry

0.5

0.5

Holding

Holding

0.9

0.1

Paint

Painted

~Clean

Holding

Holding

Painted

~Clean

Holding & Clean & Painted

Finish p > 0.85
Assessing the Plan

Initial: 
{(Dry, ∼Holding, Clean, Painted), 0.7)
(∼Dry, ∼Holding, Clean, Painted), 0.3}

Goal: pr(Holding & Clean & Painted) > 0.85

Paint: 
{(Dry, ∼Holding, Clean, Painted), 0.63)
(Dry, ∼Holding, ∼Clean, Painted), 0.07)
(∼Dry, ∼Holding, Clean, Painted), 0.27)
(∼Dry, ∼Holding, ∼Clean, Painted), 0.03}

Dry-It: 
{(Dry, ∼Holding, Clean, Painted), 0.63)
(Dry, ∼Holding, ∼Clean, Painted), 0.07)
(Dry, ∼Holding, Clean, Painted), 0.216)
(∼Dry, ∼Holding, Clean, Painted), 0.054)
(Dry, ∼Holding, ∼Clean, Painted), 0.024)
(∼Dry, ∼Holding, ∼Clean, Painted), 0.006)

Pickup: 
{(Dry, Holding, Clean, Painted), 0.8037)
(Dry, ∼Holding, Clean, Painted), 0.0423)
(Dry, Holding, ∼Clean, Painted), 0.0893)
(Dry, ∼Holding, ∼Clean, Painted), 0.0047)
(∼Dry, Holding, Clean, Painted), 0.0513)
(∼Dry, ∼Holding, Clean, Painted), 0.0027)
(∼Dry, Holding, ∼Clean, Painted), 0.0057)
(∼Dry, ∼Holding, ∼Clean, Painted), 0.0003)}
Buridan Summary

• Handles Probabilistic Actions
  – Outcomes conditioned on current state and random chance

• Different Notion of Plan Success
  – Probability of achieving goal greater than threshold
  – Adds multiple actions to increase probability

• No Observational Actions
  – *Not a conditional planner*
C-Buridan (Draper, 1994)

- Conditional, Partial-Order Planner

- Extensions to Buridan
  - Representation: Observation labels on actions
    - Clear distinction between effects and observations
    - Models noisy sensors
  - Algorithm: *Conditioning (branching)* to remove threats
    - Add observation actions to separate contexts
    - Propagate context labels


**Differences from CNLP**

- Branches can Rejoin
  - Plans are DAG’s

- Branch Added Only to Remove Threat
  - Not really “planning to observe”

- No a priori Relationship Between Observation Labels and Propositions
  - Planner must “discover” correlations
**Conditionally Processing Widgets (I)**

- Start
  - Flawed, Blemished, ~Processed, ~Painted (p=0.3)
  - ~Flawed, ~Blemished, ~Processed, ~Painted (p=0.7)

- Paint
  - Painted
    - ~Blemished
      - Ship
        - Processed
          - ~Processed, ~Flawed
          - Processed
            - Processed & Painted
              - Finish p > 0.85
**Conditionally Processing Widgets (II)**

- Start
  - \( p = 0.7 \)
  - \( \neg \text{Flawed}, \neg \text{Blemished}, \neg \text{Processed}, \neg \text{Painted} \)
  - \( \text{Flawed}, \neg \text{Blemished}, \neg \text{Processed}, \neg \text{Painted} \)

- Paint
  - \( p = 0.3 \)
  - Processed
  - \( 1.0 \)
  - \( 0.9 \)
  - \( 0.1 \)
  - Painted
  - \( \neg \text{Blemished} \)

- Ship
  - Processed
  - \( \neg \text{Processed}, \neg \text{Flawed} \)
  - Processed
  - \( \neg \text{Processed}, \neg \text{Flawed} \)

- Reject
  - Processed
  - \( \neg \text{Processed}, \neg \text{Flawed} \)
  - Processed
  - \( \neg \text{Processed}, \neg \text{Flawed} \)

- Processed & Painted
  - Finish \( p > 0.85 \)
Conditionally Processing Widgets (III)

- Flawed, ~Blemished, ~Processed, ~Painted
- Flawed, Blemished, ~Processed, ~Painted

Start

p=0.7

~Flawed, ~Blemished, ~Processed, ~Painted

Paint

~Processed

1.0

0.9

0.1

Painted

~Blemished

Inspect

~Blemished

1.0

0.9

Blemished

α₁

α₂

Reject

~Processed, c: {α₁}

~Processed, Flawed

c: {α₁}

Processed

~Processed, Flawed

~Processed

~Flawed

Processed

~Processed

Flawed

~Flawed

Processed & Painted

Finish p > 0.85

c: {α₁, α₂}

Ship

Processed

~Processed

Flawed

~Flawed

Processed

~Processed

~Flawed

Processed