Planning in the "Real World"

Why does STRIPS not suffice?

- Time impacts planning and especially concurrent plans
- Resources are difficult to handle in STRIPS
  - Unary resources, such as equipment (can be done)
  - Multiple resources, such as equipment (hard, but not impossible)
  - Numerical resources, such as fuel (very hard and expensive)
  - etc.
- Numerical calculations are impossible in STRIPS
- Complex conditions and state constraints are very hard in STRIPS
- Uncertainty and sensing is very hard to do in STRIP

But, can still use strips in real world, if we are clever…

Planning with Time and Resources

Planning techniques for complex domains

- Hierarchical Task Networks
- Constraint-based Planning

Planning techniques for uncertainty/sensing

- Conformant planning
- Conditional planning
- Execution and replanning
- Continuous planning

Hierarchical Task Networks

Basic Idea:

- Describe higher-level action as a set of lower-level actions
- Example: BuildHouse has actions:
  - Put in foundation
  - Put up walls
  - Add roof
  - Add interiors
  - Do exteriors
- along with conditions about the actions:
  - Foundation must come before walls
  - Roof comes before interiors
  - Adding roof requires resources "roofers", "crane", "access",…
  - etc.

Search technique:

- Select expansion for high-level action and resolve conditions

Hierarchical Task Networks

Advantages of HTN

- Can describe more complex conditions than in STRIPS
- Can utilize structure to separate subgoals

Disadvantages of HTN

- Requires specifying "how to" in planning domain
- Expensive to add completely new goals (e.g., build a rowhouse)

More realistic approach

- Mix HTN and STRIPS to permit both hierarchical breakdown and actions that have preconditions and effects.

Constraint-based Planning

Basic Idea:

- Describe each action, state, timepoint, resource allocation, etc., with variables
- Specify relations, timing, resources, etc. as constraints on variables
- When new actions are added, add new variables and constraints
- Use inferences to maintain consistency and identify complete plans
- Search: Select underspecified variable and make decisions

Planning with Uncertainty

Example:

- Rover is to take a picture but not sure about exact location

Conformant planning

- Take pictures in all directions and from all possible locations

Conditional planning

- Senses location and then either drive or take picture

Execution and replanning

- Drive and take picture, but track outcome and replan if needed

Continuous planning

- Drive and take picture, but continuously look for better plans, e.g., also grab sample that was to be done later

So what can we do with STRIPS in real world?

Rover domain (almost "real world")

- Actions:
  - Move to location
  - Pick up object
  - Drop object
  - …
- Sensor information:
  - Outcome of actions
  - Location
  - …

But STRIPS cannot handle sensing, execution, etc.

- However, can use STRIPS as part of that

So what can we do with STRIPS in real world?

Interleave planning and step execution

- Use sensing to determine current state
- Consider goals to be achieved next
- If current plan still works, continue with that
- Else, build plan to achieve goals from current state
- Execute one or more steps in plans
- Repeat

Possible implementation

- Core planner is simple STRIPS
- Domain must be specified
- Execution understands plans and executes them stepwise

Decisions under Uncertainty

Probability and inference

- Uncertainty and decisions
- Axioms of probability
- Probability tables and Bayes-nets

Decisions under uncertainty and adversity

- Markov decision problems
- Methods to solve Markov decision problems
- Game theory

Why worry about uncertainty?

Uncertainty is not randomness

- Probabilities impacted by decisions

Uncertainty may be in knowledge, not reality

- Information not available or not measurable

Can do better by considering uncertainty

- Likeliest outcome not necessary best
- Can impact uncertainty (e.g., by sensing or behavior)

Use probability theory to work with uncertainty

Use utility theory to maximize value overall

- Utility is sum over probability of result x value of result

Basic notions in probability theory

Random variables

- Variables taking on values according to probabilities

Probability distribution

Weather	Probability
sun	0.3
rain	0.5
snow	0.25

Joint probability distribution

Weather	Grass=dry	Grass=wet	Grass=frozen
sun	0.2	0.05	0.05
rain	0.05	0.4	0.05
snow	0.02	0.08	0.15

Conditional and unconditional probabilities

Unconditional probability

- Example: P(weather=rain) = 0.5

Conditional probability

- Example: P(weather=rain | grass=wet) = 0.9

Probability Axioms

Axioms

- $0 \leq P(a) \leq 1$
- $P(T) = 1$
- $P(F) = 0$
- $P(a \vee b) = P(a) + P(b) - P(a \wedge b)$

Conditional probability axioms

- $P(a|b) = P(a \wedge b) / P(b)$
- $P(a \wedge b) = P(a|b) P(b)$

Probability calculations

Let's calculate:

- P(Weather=rain)
- P(Weather=rain | Grass=wet)

Weather	Grass=dry	Grass=wet	Grass=frozen
sun	0.2	0.05	0.05
rain	0.05	0.4	0.05
snow	0.02	0.08	0.15

Independent variables

If variables a og b eru independent:

- $P(a \wedge b) = P(a)\,P(b)$

Example:

- Stocks is independent of Weather andGrass

	Stocks=up	Stocks=up	Stocks=up	Stocks=down	Stocks=down	Stocks=down
Weather	Grass=dry	Grass=wet	Grass=frozen	Grass=dry	Grass=wet	Grass=frozen
sun	0.1	0.025	0.025	0.1	0.025	0.025
rain	0.025	0.2	0.025	0.025	0.2	0.025
snow	0.01	0.04	0.075	0.01	0.04	0.075

Result:

- Can make two smaller tables

Weather	Grass=dry	Grass=wet	Grass=frozen
sun	0.2	0.05	0.05
rain	0.05	0.4	0.05
snow	0.02	0.08	0.15

Stocks	
up	0.5
down	0.5

Bays Rule

Bays Rule:

- P(b|a) = (P(a|b) P(b) / P(a)

Use:

- Often we know relation in one direction, but not other
- Example: 80% of people with flue have fever
- Want to know: If person has fever, what are odds of having flue
- Let's say odds of fever are 2% overall and odds of flue 0.1%
- Then, odds of patient having flue, given fever, is:
  (0.8*0.001) / 0.02 = 0.04 = 4%

Conditional independence

Basic idea:

- Independent variables are good™
- But, most variables end up being dependent
- However, many are dependent "through another variable

Example:

Weather	Grass=dry	Grass=wet	Grass=frozen
sun	0.2	0.05	0.05
rain	0.05	0.4	0.05
snow	0.02	0.08	0.15

Weather	BBQ=yes	BBQ=no
sun	0.7	0.3
rain	0.05	0.95
snow	0.01	0.99

Conditional Independence

Skilgreining:

- Breytur a og b eru óháðar ef gildi C er gefið, ef
  $P(a \wedge b \mid C=c) = P(a \mid C=c)\, P(b \mid C=c)$

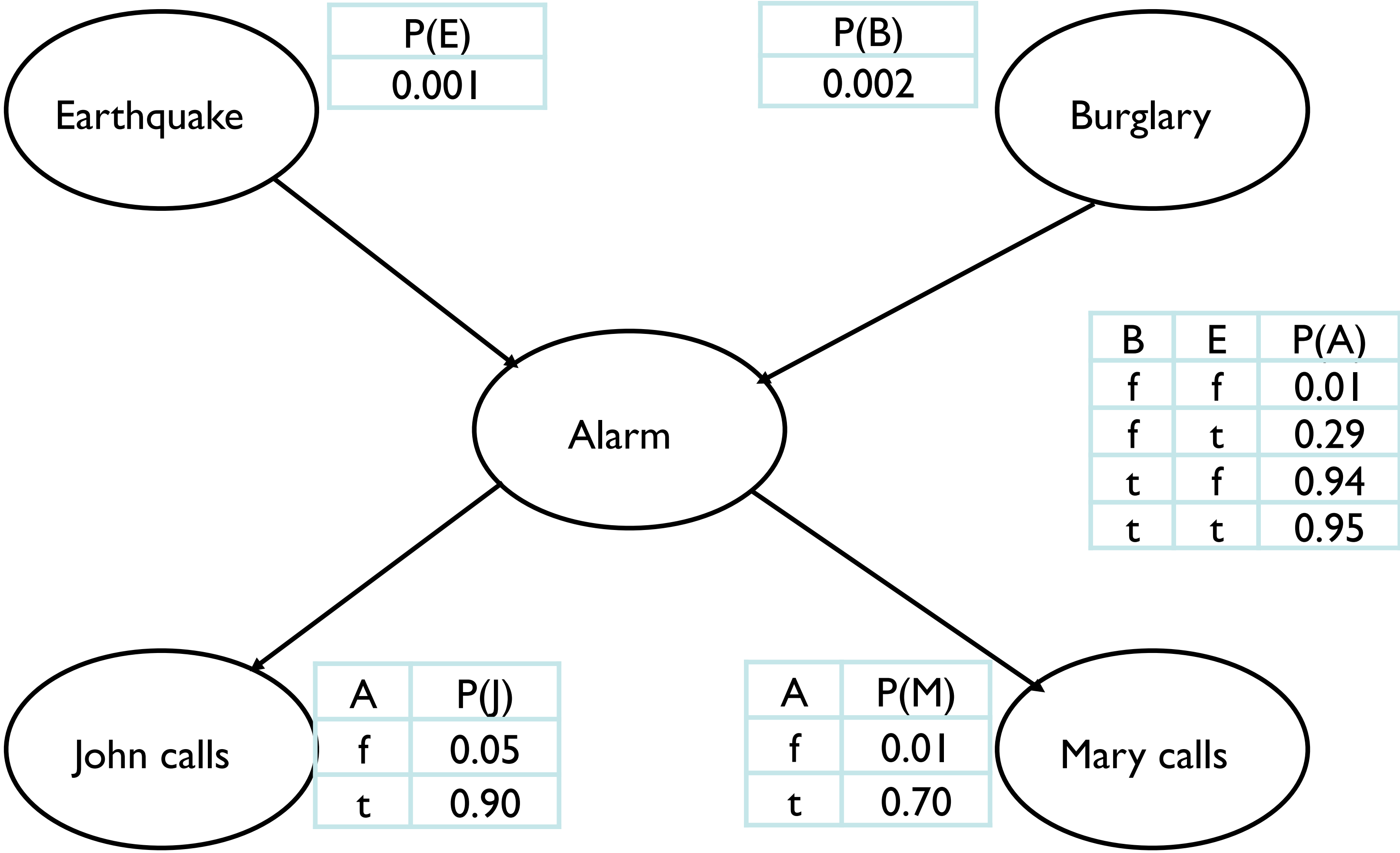
Notkun með Bays reglu:

- $P(\text{Weather=rain} \mid \text{Grass=wet} \wedge \text{BBQ=no})$
  $= \alpha\ P(\text{Grass=wet} \mid \text{Weather=rain})\, P(\text{BBQ=no} \mid \text{Weather=rain})$

Nytsemi:

- Getum notað Bays net til að lýsa líkindum og venslum líkinda á þjappaðan hátt
- Það gerir okkur kleyft að draga ályktanir út frá líkindaupplýsingum

Example of Bays net



P(E)
0.001

P(B)
0.002

B	E	P(A)
f	f	0.01
f	t	0.29
t	f	0.94
t	t	0.95

A	P(J)
f	0.05
t	0.90

A	P(M)
f	0.01
t	0.70

Planning in uncertain World

Problem

- How to decide what to do when outcomes are uncertain, but still do the right thing at each point

Basic idea

- Build a policy that says what to do in each possible position

Markov Decision Problems

- Probability of outcomes, given a state and an action
- Probability independent of earlier states and actions

Game Theory

- Possible states and outcomes, own actions and adversary actions

Markov ákvörðunarvandamál

Færslur í óvissu

- Fyrir hverja stöðu og hverja aðgerð, fáum líkindadreifingu á hvaða stöðu við lendum í

Markov færslur (Markov transitions)

- Færslur þar sem líkindi á niðurstöðu færslu eru einungis háð stöðu og aðgerð, en ekki fyrri stöðum og aðgerðum

Markov ákvörðunarvandamál

- Upphafsstaða s_0
- Færslulíkan $T(s, a, t) = p$
  - Fyrir hverja stöðu og hverja aðgerð, fáum líkindadreifingu fyrir stöður sem við getum lent í - ef a er framkvæmt í stöðu s, lendum í með líkindum p
- Ávinningsfall $R(s)$
  - Fyrir hverja stöðu, fáum ávinning af því að vera í þeirri stöðu

Lausnaraðferðir

Markmið

- Ákvarða fyrir hverja stöðu hvað sé best að gera
- Ef við vitum gildi hverrar stöðu er auðvelt að ákveða aðgerðir
  - veljum þá aðgerð sem gefur hæsta viðbúna gildi (expected value)

Gildisítrun (value iteration)

- Reiknum út gildi hverrar stöðu, miðað við að fylgjum stefnu sem leitast við að fara í stöður með hærri gildi
- Raunverulegt gildi stöðu, miðað við samsavarandi stefnu er
  $U_\pi(s) = E\,[\,\Sigma\,\gamma^t\,R(s_t)\mid\pi\,]$
- En getum notað auðveldari jöfnu sem lýsir venslum staða í lausn

Stefnuítrun (policy iteration)

Uppfærsla stöðugilda

Bellman jafnan

- $U(s) = R(s) + \gamma \max_a \Sigma_t \; T(s,a,t) \, U'(t)$

þar sem

- U' er núverandi gildismat
- R er fall sem gefur ávinning fyrir að vera í stöðu s
- γ er hlutfall gildis framtíðarávinnings og fengins ávinnings
  - betri er einn fugl í hendi en tveir í skógi

Notum hana til að uppfæra stöðugildi í ítrun

- $U(s) = R(s) + \gamma \max_a \Sigma_t \; T(s,a,t) \, U'(t)$

Gildisítrun

U(s)=0 fyrir öll s

Endurtökum

- U' = U
- fyrir hverja stöðu s
    - $U(s) = R(s) + \gamma \max_a \Sigma_t\ T(s,a,t)\ U'(t)$

þar til nægilegt jafnvægi næst