Planning, Execution & Learning: Decision Theoretic Planning

Reid Simmons

Preference Models

Decision Theory

- Descriptive Theory: ***How to make choices among alternatives***
  - Does not describe how to ***generate*** the alternatives

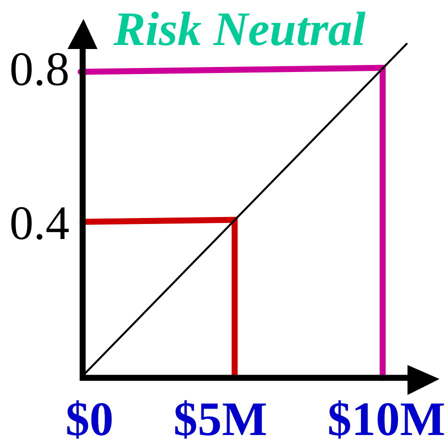
- Utility Models
  - Capture preferences for rewards and resource consumption
  - Capture risk attitude (risk-neutral, risk-averse, risk-seeking)
  - ***Expected Reward*** *vs*. ***Expected Utility***

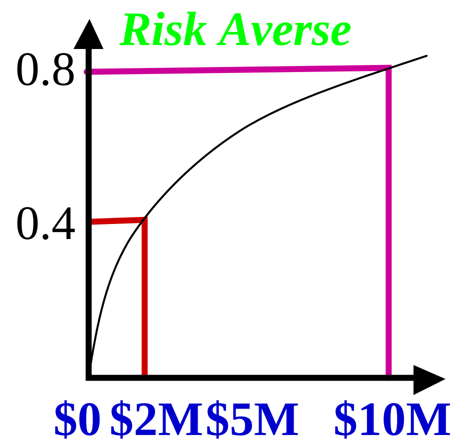
Lottery:

50%: $10,000,000
50%: $0

ER: $5,000,000

EU: ???





Utility Models for Planning

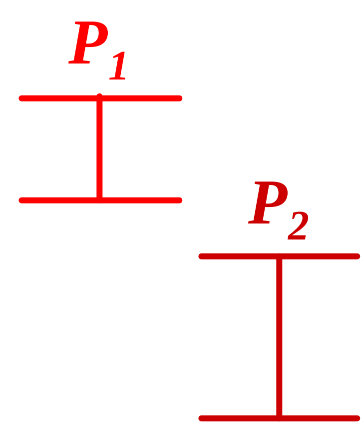
- Goal-Oriented Utility Models (Haddawy & Hanks, 1993)
  - $\mathrm{EU}(P) = \sum(\mathrm{U}(a) \cdot \mathrm{pr}(a \mid P))$
  - $\mathrm{U}(a) = k \cdot \mathrm{UG}(a) + \mathrm{UR}(a)$
    - UR: Resource consumption component
    - UG: Goal satisfaction component
      - Atemporal component (degree of goal satisfaction)
      - Temporal component (deadline)

- Deadlines
  - ***Soft***: Exponential drop-off
  - ***Hard***: Discontinuous drop-off (step function)

Plan Dominance

- What?
  - Plan can be eliminated from consideration if it is provably worse than another plan under consideration

- How?
  - For **complete** plans: P_1 dominates P_2 if $EU(P_1) > EU(P_2)$
  - For **partial** plans:
    - Maintain range of utilities: $[\underline{EU}, \overline{EU}]$
    - P_1 dominates P_2 if $\underline{EU}(P_1) > \overline{EU}(P_2)$



Pyrrhus (Williamson & Hanks, 1994)

- Decision-Theoretic Extension to UCPOP
  - ***Deterministic***, but plans evaluated with respect to ***utility***

- Utility Model:
  - $U(a) = k \cdot S_G(a) \cdot T(E_G(a)) + (1 - k) \cdot R(a)$
    - $S_G(a)$: Is goal satisfied (0/1)
    - $E_G(a)$: Earliest time goal is achieved
    - $R(a)$: Resource utilization
  - Can model classical planners
    - Set *T* and *R* to be constant functions

Pyrrhus Algorithm

1. Estimate utility of null plan (utility of initial resources)
2. Fix a flaw (based on UCPOP)
3. Estimate upper bound on utility of (partial) plan
   - $E_G(a)$: No earlier than current plan execution time
   - $R(a)$: No smaller than current resource utilization
     - Assumes ***asset-position monotonicity*** ("cannot can something for nothing")
4. When some plan becomes completely refined
   - Determine precise utility
   - Save plan, if best so far
   - Eliminate partial plans that are dominated

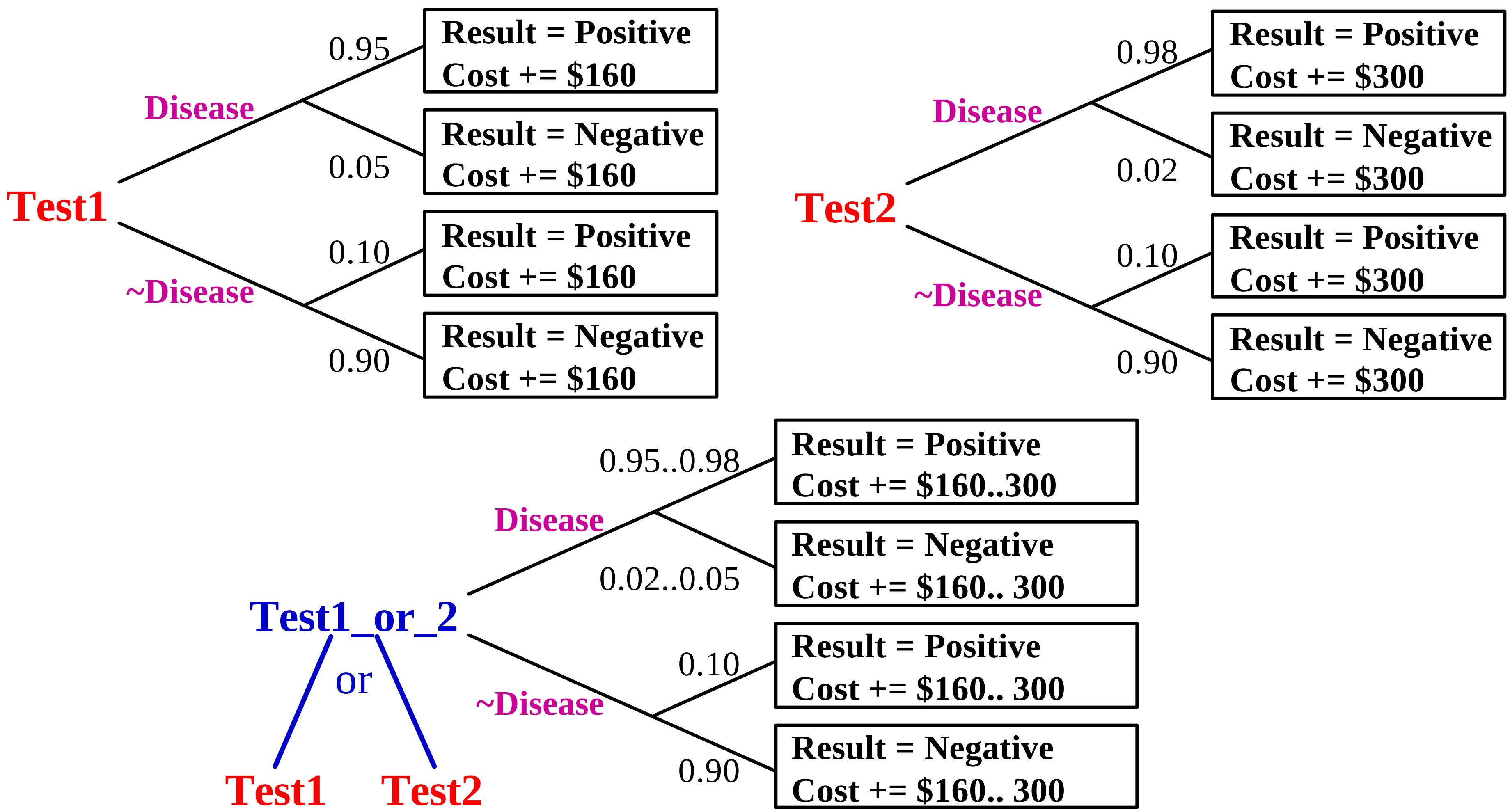
DRIPS (Haddawy & Hanks, 1994)

- Decision-Theoretic Hierarchical Refinement (HTN) Planner
  - Probabilistic; Maximizes expected utility
- Utility Model:
  - Similar to that used in Pyrrhus
- Action Representation:
  - ***Concrete*** Actions
    - Primitives, fully refined
  - ***Abstract*** Actions
    - ***Sequence*** of actions (e.g., “load and drive”)
    - ***Alternative*** actions (e.g., “drive slowly”, “drive fast”)
  - Planner works by refining abstract actions

DRIPS Algorithm

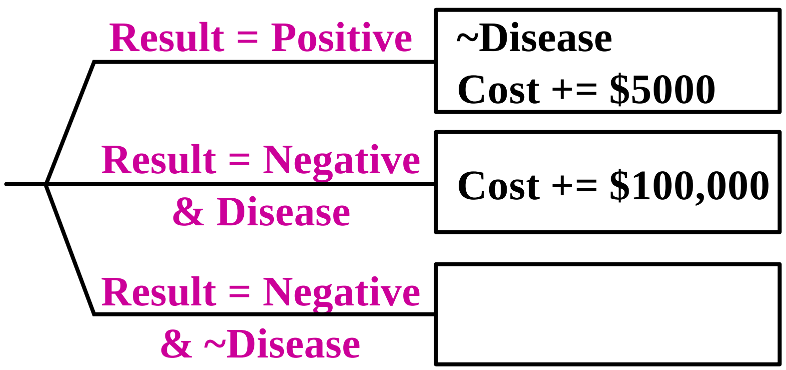
1. Evaluate Abstract Plan
   - Evaluate each ***chronicle*** explicitly
   - Can yield ranges of probabilities, durations, and resources
   - Compute upper and lower bound on utility
     - Assumes utility function is monotonic
2. Eliminate Dominated Plans
   - Stop when only fully instantiated plans remain
3. Refine Plan
   - **Choose**: Expand action *or* Create alternative plans
4. Go to step 1

DRIPS Example (I)

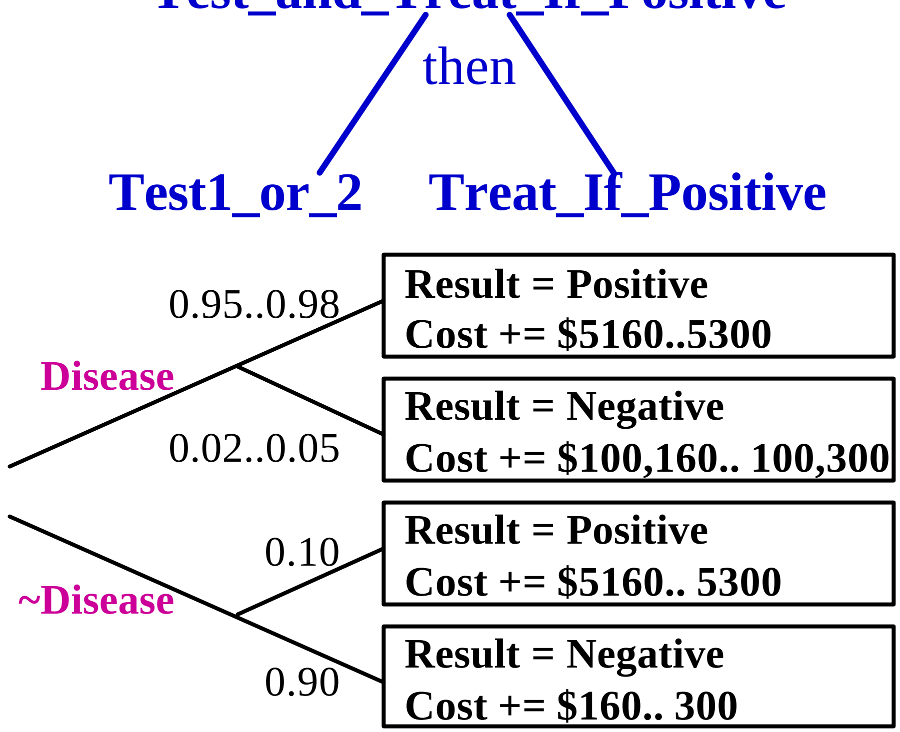


DRIPS Example (II)

Treat_If_Positive



Test_and_Treat_If_Positive



Evaluating Decision-Theoretic Plans

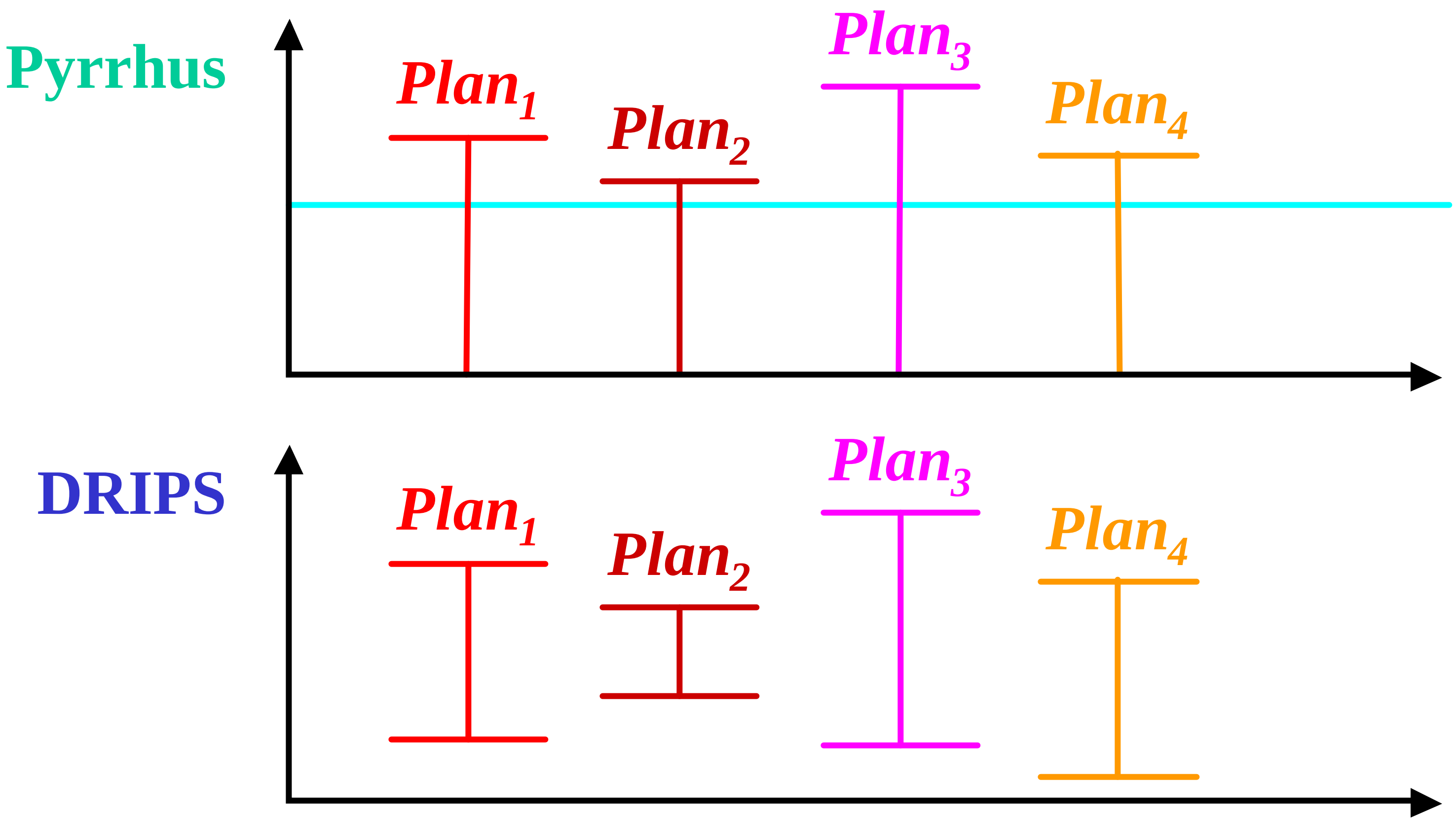
- Determining Expected ***Reward*** is Linear
  - $ER(P) = \sum(R(a) \cdot pr(a \mid P))$
- Determining Expected ***Utility*** is NP-Complete, in general
  - Marginal utility of an action depends on current "wealth":
    $EU(P) = \sum(U(a, P) \cdot pr(a \mid P))$
    $U(a, P) = f(a + \sum_{a'=\text{"all actions preceding a"}} U(a', P))$
- For Certain Types of Utility Functions, Can do Much Better
  - For ***linear*** (risk-neutral) utility functions, $U(a, P) \propto R(a)$
  - For ***exponential*** ($e^{k(r_1 + r_2)}$) utility functions, one can transform the problem: $\ln(EU(P)) \approx k \sum R(a)$
  - Linear and exponential are the *only* such ***decomposable*** utility functions

Search Control for Probabilistic Planners

Which Plan to Work On?

Which Refinement to Apply?

Choosing a Plan to Work On



- ***Provably Optimal*** to Select Plan with Greatest Upper Bound (Goodwin, 1996)

Choosing Refinement to Apply

- Optimal to Choose Refinement That Gives ***Maximal Impact*** on Upper Bound of Utility for ***Least Amount of Work*** (Goodwin, 1996)
  – Estimate average change in utility
  – Estimate work needed to totally refine/complete plan

- Example: Application to Pyrrhus
  – Current lower bound: EU_{opt}
  – To eliminate plan, must decrease bound ($EU < \overline{EU}_{opt}$)
  – Estimate work needed: $\sum_{i=1}^{n} B^{(\overline{EU}_i - EU_{opt})/\delta}$
    - n: Number of sub-plans; B: average branching factor
    - δ: Average change in $\overline{EU}$
  – Choose refinement that minimizes estimated work