Introduction to AI

Rational Agents

- Percept sequence → action
- Exploitation vs exploration
- Autonomous: learn from itself

Specifying Task Environment

- PEAS: Performance Measure, Environment, Actuators (action), Sensors (percept)
- Fully-observerable? Deterministic? Episodic? Static? Discrete? Single-agent?

Agent Functions and Programs

- Table-lookup agent: large table, no autonomy
- Simple Reflex Agent: update state from percept (passive)
- Model-based
  - Goal-based: active in achieving goals
  - Utility-based: maximise utility

Uninformed Search

- Assume env = fully-obs, deterministic, discrete
- Solution = action seq, s_0 to g
  - Unique? Optimal?
  - Additive path cost ≥ ε
  - Goal test: s = g? Explicit states or boolean satisfiability
- Tree Search: Frontier = nodes seen, not explored (initial: s_0)
  - Explore, add neighbours
  - No search history from s to g
- Graph Search: explored nodes not revisited

Search Strategies

- *BFS*: Expand shallowest node
- *Uniform-Cost Search*
  - Expand from least path cost
  - Frontier = Priority queue, Equal step cost → BFS
- *DFS*: Expand deepest node
- *Depth-Limited Search*
  - DFS with limit l
- *Iterative Deepening Search*
  - DLS, increasing depth
  - \# nodes generated:

Property	BFS	UCS	DFS	DLS	IDS
Complete	Yes¹	Yes²	No	No	Yes¹
Optimal	No³	Yes	No	No	No³
Time	$O(b^d)$	$O\left(b^{1+\left\lfloor\frac{C^*}{\varepsilon}\right\rfloor}\right)$	$O(b^m)$	$O(b^\ell)$	$O(b^d)$
Space	$O(b^d)$	$O\left(b^{1+\left\lfloor\frac{C^*}{\varepsilon}\right\rfloor}\right)$	$O(bm)$	$O(b\ell)$	$O(bd)$

Informed Search

- Heuristics: optimise the order of node expansion
- *Best-First Search*: expand with lowest f(n)
- Admissible: h(n) ≤ h*(n)
  - Consider relaxed problems
  - Dominance: $h_2(n) \geq h_1(n)$
    - Dominant → lower cost
- Consistent
  - h(n) ≤ d(n, n') + h(n')

Greedy Best-First Search:

- Complete, not optimal
- Time: $O(b^m)$, Space: $O(b^m)$
- f(n) = h(n)
- BFS: all heuristics ∞ except g

A* Search

- f(n) = g(n) + h(n)
- h(n) admissible → Tree-S opt
- h(n) consistent → Graph-S opt
- Zero-heuristic: only consider g(n) → UCS

Local Hill-Climbing Search

- Irrelevant path to goal
- Local maxima: Sideway moves, random restarts

Adversarial Search

- Utility-maximising opponent
- Zero-sum utility, MAX and MIN
- Complete strategy: defined for every node, even suboptimal

Minimax Strategy (Optimal)

- MAX ↑ payoff, MIN ↓ payoff
- spne; every subtree ne

Complete?	Yes (if game tree is finite)
Optimal	Yes (optimal gameplay)
Time	$O(b^m)$
Space	Like DFS: $O(bm)$

α-β Pruning

- Can't expand entire game tree
- Prune if other player can guarantee better outcome
- Perfect ordering: $O(b^{m/2})$
  - Depends on expansion order heuristics, random = $O(b^{3m/4})$
- Unknown depth → cutoff limit
  - E(utility) using evaluation fn

Stochastic Games: chance layers

Constraint Satisfaction Problems

- Constraint scope: vars, relatns
- Find complete, consistent (satisfy all constr) assignment
  - Goal test: all vars assigned
- Graph: adjacency → constraint
- All solutions at depth n → DFS
- Large tree size ($n!d^n$) leaves, every level = (n-i)(d)

Backtracking Search

- Each level = assign one variable, order irrelevant
- DFS, inference on the domain of remaining variables
- Most Constrained Variable: minimum-remaining-values (MRV) heuristic
- Most Constraining Variable: degree heuristic (choose highest degree)
- Least Constraining Value

Forward Checking

- Keep track of remaining legal values for unassigned vars
- Terminate search when any var has no legal values
- Propagates information from assigned → unassigned vars

Inference

- Unary constr: node consistency
- Binary constr: arc consistency
  - Directed, binary = two arcs
  - Start with arcs of neighbours which are not assigned
  - Propagate to neighbours
  - Time complexity $O(n^2d^3)$
- Maintaining AC (MAC)
  - Establish AC at root
  - Assign variable, repeat
  - Backtrck? Undo domn redux?
- k-consistency → split to binary

Local Search

- Arbitrary assignment, change values w/ min-conflicts heuristic (hill-climbing search)

Tree (no cycles): CSP in $O(nd^2)$

Logical Agents

- Representation in KB
- $\alpha \vDash \beta \Leftrightarrow M(\alpha) \subseteq M(\beta)$
- Inference Algorithm
  - Sound: KB $\vDash \alpha$
  - Complete: KB $\vdash \alpha$

Propositional Logic

- Sentences = symbols, Model = truth assignment to variables
- Truth table enumeration
  - $O(2^n)$ time, O (n) space
- KB $\vDash \alpha \Leftrightarrow$ KB ^ $\neg\alpha$ unsatisfiable

CNF Resolution: sound, complete

- $(P \vee x) \wedge (Q \vee \neg x) \rightarrow P \vee Q$

Horn Form: FC, Modus Ponens

- $[A \wedge (A \rightarrow B)] \rightarrow B$
- Count # symbols in premise
- Complete: every clause in KB is true in model w/ vars assigned

Backward Chaining

- Given q, recursively prove premises to conclude q
- Avoid loops in recursion

DPLL: Complete, backtracking

- Checks if CNF is satisfiable
- Early termination
  - Literal true → clause true
  - Clause false → CNF unsatisf
- Pure symbols: assign true
  - Least constraining value
- Unit clause: assign true
  - Most constrained variable

WalkSAT: ↓ # unsatisf clauses

- Incomplete
- Unsatisf clause → random flip/ flip to maximise Φ

First Order Logic

- More expressive w/ objects and relations, sentences = true
- $\exists$:^, $\forall$:→, equality: same obj.
- KB: percept/ reflex rules

Inference

- Propositionalise: $\exists$ → Skolem
- Standardising apart: no clash
- Apply resolution
- Entailment: semi-decidable
- Unification: returns θ
  - SUBST(θ, p) = SUBST(θ, q)
  - Unique MGU upto rename
  - x appears in Q: cannot unify

GMP

- Under subst, conclusion true
- Sound: universal instantiation
  - P $\vDash$ SUBST(θ, p) $\forall\theta$

FC: sound, complete

- Rule matching expensive
  - Conjunct ordering, MRV
- Redundant rule matching
  - Incremental FC: premise has new fact from t-1
- Generate irrelevant facts

BC: incomplete (loops)

- Check against goal stack
- Cache solutions to subgoals

Resolution → CNF

- Drop $\forall$, Skolemnise $\exists$ (external $\forall$: use F(x))
- Under SUBST, p unifies with $\neg$q
  - Resolve under SUBST
- First-order factoring: unify two literals $G(a) \vee G(k) \rightarrow G(k)$

Uncertainty

Probability

- Joint Probability
- Conditional Probability
- Independence

Bayesian Inference

- $P(X|Y_1, Y_2 \ldots)$
- Conditional Independence
  - $P(A \wedge B|C) = P(A|C) * P(B|C)$

Bayesian Networks

- Representing joint distr.
- Conditional Probability Tables
- Variables are only conditionally dependent on parents
- Independent Causes/ Conditionally Independent Effects
- Choose min. set of parents available
- Markov Blanket: node conditionally independent given parents, children, children's parents

Bayesian Networks

d-separation

- Determine conditional independence given known ε
- Conditionally independent $\Leftrightarrow$ all undirected paths not active
- Path active $\Leftrightarrow$ all triplets active