Reactive agents

Reactive policy (strategy): simple state → response mapping (may be *stateful*). Good for realtime. Subsumption architecture manages conflicting behaviors. Strategy is computed from the reward and transition functions to maximize immediate *and* future rewards. Decision process: describes knowledge of rewards & transitions. *Markov*: nondeterministic transitions; *partially observable MDP*: state is uncertain; *Q-learning*: transitions and rewards are not known. Value of a state: $V(s_i) = R(s_i, a^*) + \gamma V(T(s_i, a^*))$, a^* the best action, γ the *discount factor*. The *value iteration* algorithm (initialize V_i at random, pick best action by trying all, update V_i, stop when max update is small) converges to the true value. For MDP, transition function is a probability distribution: take expected value. The *policy iteration* optimizes over the policy directly by solving for the linear equations of values for the current policy. Q-learning: need to learn transitions and rewards (exploration / exploitation, use $\alpha(t)$). POMDP can be converted to deterministic *MDP* by replacing state by a *belief* function.

Deliberative agents

Deliberative architecture: explicit representation of goal states and plans. Considers only the subset of states that may be visited. Search algorithms: DFS (too much time spent in unpromising branches), BFS (too much memory usage), Iterative Deepening (DFS + depth limit), Branch and Bound (ignore nodes with higher cost than best found so far). With adversary: minimax, $\alpha - \beta$ pruning. Regret: difference in outcome between playing move i and the optimal move. In games with chance, minimize expected regret or worst-case regret. Big / chance games: evaluate horizon states with a *heuristic* and / or *Monte-Carlo sampling*. Upper Confidence Bound for game trees: minimax over $UCB_i = \frac{\text{wins}_i}{N_i} + c\sqrt{\frac{\log N}{N_i}}$.
Factored representation: Use *situation calculus*: operators transform predicates (preconditions, add-list, delete-list). *Frame axioms* carry over predicates that were not modified. *Policy* and *reward functions* should be factored. More difficult: factoring the estimate *value* of a state (use *basis functions*).
Leat-commitment principle: delay commitment to the order in which actions are taken (modulo dependencies). Determine policy by fixpoint iteration (basis functions: solve for weights that minimize the error of the value function recurrence ≡ least squares).
Partial-order planning: discover which actions can be carried out in parallel. *GraphPlan* builds layers (time) with nodes (proposition and actions) and directed edges (in: preconditions, out: add-list). Built in polynomial time, not exhaustive but complete. SAT: State = vector of state variables (each with a domain), add constraints for: pre and post-conditions for operators, incompatible propositions, exclusion for operators using the same resource. Time is broken up into $2n$ discreet points (states, actions).

Multiagent systems

Multiagent modalities: centralized with shared memory, centralized or distributed with message passing, decentralized (no communication, but observe common signals). Ontologies: provides a shared vocabulary for heterogeneous agents. Self-interested agents: always act to maximize their own interest. We need to create incentives. Centralized planning: can explicitly detect conflicts and synergy. E.g. *blackboard system* (shared memory), *publish-subscribe system* (notify on resource usage). Using reactive agents: optimize sum of rewards or each agent optimizes its own (find a NE using policy iteration, but probably not the best one). Using deliberative agents: find conflicting resource usages and similar goals. Partial-global planning: goal tree in which each agent inserts its partial plans (expressed with predicates). Can then reorder actions, exchange tasks. Contract net: cooperation protocol. Managers divide tasks, contractors place bids, contract is made for lowest bid (no negotiation). Tasks can be subcontracted. Market-based variation: managers increase prices until they obtain a solution.

Distributed multiagent systems

Distributed contract nets: make contracts asynchronously and contact agents directly. Agents try to sub-contract tasks with high *marginal cost*. Problems: incremental bidding can't work, impossible to resolve conflict, bidders must speculate on future tasks. Distributed SAT: variables (tasks), domains (resources that can carry out the task), constraints (timing, preconditions and resources), relations (inequality). Sync backtracking: each agent extends the partial solution of the previous, allows use of heuristics (async: all in parallel, but exponential number of messages needed). Distributed DP: organize agents in a rooted tree + backedges. Send `util` (constraint) messages up, parent decides the best value locally, get `value` messages down. Variables are collapsed, only local node knowns which value to set. Distributed local search: start with random assignment, make local change which most reduces the number of conflicts. Neighborhood = variables connected through constraints. Changes can be async as long as there is only 1 change per neighborhood. Breakout algorithm: extension of min-conflict, assigning *dynamic priority* to each constraint. Pick change that reduces most the sum of priorities. All remaining conflicts have priority increased when reaching local minimum, then restart. Termination: when time count is larger than the distance to the furthest agent.

Game theory

Games: zero / general sum, 2 / N players. Strategies: strictly / (very) weakly dominant, pure, mixed, *minimax* (for zero-sum: *value* of the game = expected gain v_A = expected loss v_B). Computing minimax for B: find set of p_i^B such that expected gain v of A is minimized and $\forall a_i^A, \sum_{a_j^B} p_j^B R_A(a_i^A, a_j^B) \leq v$. Utility function: can encode risk-profile. Support: set of actions with $p_a > 0$. Nash equilibria: no player gains from changing strategy, all other things being equal. Theorem: every game has at least a set of (mixed) NE. Compute them by removing dominated actions and going through possible subsets of actions (*supports*). For N players, NE exists but not necessarily unique or minimax. Uncertain utilities: *ex-ante* (no knowledge at all), *ex-interim* (own agent type is known), *ex-post* (strongest, knowledge of all types). Bayes-NE: NE over ex-ante expected utilities. Ex-post NE: strategies give highest utility no matter the value of the unknown information (not always possible).

Agent cooperation, negotiation

Correlated equilibrium: act depending on an external factor (e.g. coin flip). Mediator: vehicle to enforce a contract. Plays depending on the number of agents which chose the mediator. Negotiation: make, accept or refuse offers to agree on a joint strategy. *Strategic* or *axiomatic*. Alternating offers: broken, first offer has more power. Can introduce time constraints, discount factor. Nash bargaining solution: maximize the product of utility gains (from *conflict deal*), concession is made by deal with lowest product. Mixed strategies: can bargain about probabilities. Monotonic concessions: reaches NBS. Agent with lowest risk tolerance $\frac{u_i(D_i)-u_i(D_j)}{u_i(D_i)-u_i(D_c)}$ must make an offer. Stackelberg games: decisions in sequence between a leader and a follower.

Mechanism design

Social choice: constructing a joint preference order reflecting individual, private orderings. Implementations: dominant equilibrium, Bayes-Nash equilibrium. Mechanisms: social choice function + payment rule. Goal: *incentive compatibility* (best strategy for agents leads to optimizing the social choice function). Revelation principle: for any mechanism, there is a truthful mechanism with the same outcomes and payments (proof by construction). VCG tax: maximizes the sum of declared valuations. $\text{tax}(A_i) = \sum_{A_{j\neq i}} v_j(d_{-i}) - v_j(d_{\text{all}})$. It is truthful, rational and incentive compatible. Generalization: *Groves mechanism* ≈ VCG with offset (e.g. refund part of the tax). Problems: collusions, not Pareto-efficient (tax must be wasted). Roberts' theorem: affine maximizer social choice functions are the only ones that can be implemented for unrestricted preference profiles with incentive compatibility. Median rule: IC for single-peaked preferences.

Auctions

Auctions: social choice, goals are *optimal allocation*, *individual rationality*. Values: *private*, *common* (entirely dependent on other's value), *correlated*. Auction protocols: open-cry (*Dutch*, *English*); sealed-bids (*Discriminatory*, *Vickrey*). Problems: *manipulability* (collusion, demand reduction lie), *risk-averse* or non-truthful / irrational bidding, *timing*, *side-constraints* (e.g. procurement), authenticity, privacy. Revenue equivalence theorem: all 4 settings yield equivalent revenue, but not optimal allocation (not true for *correlated values*). Multi-unit Vickrey: each agent pays the price of the bid it displaced from the set of winning bids. Double auctions: sort buy & sell bids in opposing orders, price is last match. M^{th} highest price is incentive-compatible for sellers, $(M+1)^{st}$ for the buyers. *McAffee*: price is average of last (sell, buy) bid, but block that one $\implies$ loss of 1 trade but gain incentive compatibility. May bid with *price-quantity graphs* (then, sum up curves and match demand). Bargaining: for 1 buyer and 1 seller, NE at price $0.5 \times (b_1 + b_2)$. Independent English Auctions (k run in parallel): bidding strategies include *straightfoward* (ignore complementarities, bid only for the best combination); *sunk-aware* (discout perceved price of won items by some factor since they cannot be returned); *price prediction*. GVA auction (≈ VCG tax): pay for the difference between the best allocations without and with your bid. GSP auction (internet ads): i^{th} highest bidder pays $(i+1)^{st}$ price to get i^{th} slot (not incentive compatible, but stable prices and high revenue).

Coalitions

Coalitions: when *side-contracts* (utility redistribution) are allowed, coalition utility geq others $\implies$ stable. Core of a game: set of payoff distributions for which the grand coalition is stable (often empty). Core is known to be nonempty for *Superadditive* and *Convex* games ($v(S \cup T) \geq v(S) + v(T) - v(S \cap T)$). Shapley value: unique vector (if core is nonempty, SV payoffs is in it). $SV(a_i)$ = average value of added payoff when added that agent to a sub-coalition (over all orders). Agents not in any carrier coalitions has $SV(a_i) = 0$. Weighted graph games: agents are nodes, self-edges are payoffs, edges are payoffs for 2-coalitions (not possible for all games). Value of a coalition is the sum of edge weights in the subgraph.

Voting protocols

Manipulability: non-truthful voting; removing a candidate can reverse the order; vote organizer can determine the winner by changing the order in which alternatives are presented. Condorcet winner: alternative that beats (or ties) all others in a pairwise majority vote (doesn't always exist; in a majority graph, it is the node with only outgoing edges). Plurality voting: vote for your single preferred alternative (variant: carry out $n-1$ rounds, eliminate the least preferred at each round). Borda count: give $(n-1)\dots 0$ points to alternatives. Slater ranking: vote between every pair of alternatives, pick the smallest transformation to obtain a majority graph. Gibbard-Satterthwaite theorem: any deterministic voting protocol (≥ 3 alternatives) has one of these properties: dictatorial, non-truthful, or $\exists$ a candidate that cannot win.

Credits

Most content taken from the lecture notes of Boi Falting's Intelligent Agents class at EPFL, 2015.
