Problem-Solving Agents

Subclass of goal-based agents

goal formulation

problem formulation

example problems
  • toy problems
  • real-world problems

search
  • search strategies
  • constraint satisfaction

solution
Goal Formulation

Specify the objectives to be achieved

goal
  a set of desirable world states in which the objectives have been achieved

current / initial situation
  starting point for the goal formulation

actions
  cause transitions between world states
Problem Formulation

Actions and states to consider

states  possible world states
accessibility  the agent can determine via its sensors in which state it is
consequences of actions  the agent knows the results of its actions
levels  problems and actions can be specified at various levels
constraints  conditions that influence the problem-solving process
performance  measures to be applied
costs  utilization of resources
Example: vacuum world, restricted to two locations with two states (dirty, clean)
Problem Types

Not all problems are created equal

single-state problem

multiple-state problem

contingency problem

exploration problem
Single-State Problem

exact prediction is possible

state
  is known exactly after any sequence of actions

accessibility of the world
  all essential information can be obtained through sensors

consequences of actions
  are known to the agent

goal
  for each known initial state, there is a unique
goal state that is guaranteed to be reachable via
an action sequence

simplest case, but severely restricted
Example: Vacuum world, [?]p. 58
Limitations: Can’t deal with
incomplete accessibility
incomplete knowledge about consequences
changes in the world
indeterminism in the world, in actions
Multiple-State Problem

- semi-exact prediction is possible

**state** is *not* known exactly, but limited to a set of possible states after each action

**accessibility** of the world
- *not* all essential information can be obtained through sensors
- reasoning can be used to determine the set of possible states

**consequences** of actions
- are *not always or completely* known to the agent; actions or the environment might exhibit randomness

**goal** due to ignorance, there may be no fixed action sequence that leads to the goal

less restricted, but more complex
Example: Vacuum world, [?]p. 58, but the agent has no sensors. The action sequence right, suck, left, suck is guaranteed to reach the goal state from any initial state.

Limitations: Can’t deal with changes in the world during execution ("contingencies")
Contingency Problem

exact prediction is impossible

**state** unknown in advance, may depend on the outcome of actions and changes in the environment

**accessibility** of the world
- some essential information may be obtained through sensors only at execution time

**consequences** of actions
- may not be known at planning time

**goal** instead of single action sequences, there are trees of actions

**contingency** branching point in the tree of actions

**agent design** different from the previous two cases:
- the agent must act on incomplete plans

search and execution phases are interleaved

Franz J. Kurfess, Cal Poly SLO
Example: Vacuum world, [?]p. 58, The effect of a suck action is random. There is no action sequence that can be calculated at planning time and is guaranteed to reach the goal state.

Limitations: Can't deal with situations in which the environment or effects of action are unknown
Exploration Problem

- **state**
  - the set of possible states may be unknown

- **accessibility** of the world
  - some essential information may be obtained through sensors only at execution time

- **consequences** of actions
  - may not be known at planning time

- **goal**
  - can't be completely formulated in advance
  - because states and consequences may not be known at planning time

- **discovery**
  - what states exist

- **experimentation**
  - what are the outcomes of actions

- **learning**
  - remember and evaluate experiments
agent design
different from the previous cases: the agent
must experiment

search
requires search in the real world, not in an
abstract model

realistic problems, very hard
Well-Defined Problems

exact formulation of problems and solutions

initial state
  - current state / set of states, or the state at the
  - beginning of the problem-solving process
  - must be known to the agent

operator
  - description of an action

state space
  - set of all states reachable from the initial state
  - by a possible sequence of actions

path
  - in the search space
  - sequence of actions between two states

goal test
  - determines if the agent has reached a goal state

path cost
  - function that assigns a cost to a path
  - usually the sum of the costs of actions along
the path

data type  PROBLEM
  components: Initial-State, Operators,
              Goal-Test, Path-Cost

solution
  path from the initial state to a state that
  satisfies the goal test

search algorithm
  takes the problem data type and computes a
  solution

basis for a formal treatment
Performance Measuring

for problem solving

success
  Has a solution been found?

quality
  Is it a good solution?
  What are the criteria?

optimal solution
  may be difficult to find and not necessary

cost
  sum of
  • search cost (time, resources to find a solution)
  • path cost (as defined above)

Franz J. Kurfess, Cal Poly SLO
Vacuum World

simplified version

two squares, either dirty or clean, vacuum has

sensors

states
  location of vacuum, squares dirty or clean

operators
  move left, move right, suck

goal test
  all squares clean

path cost
  1 unit per action
see Figure 3.2, 3.6 in [?], p. 66
8-Queens

no queen attacks any other

states
arrangement of 8 queens on the board

operators
add a queen

goal test
no queen attacked

path cost
zero (irrelevant, all solutions are equally good)

restrictions on the states and operators can lead to vastly different search spaces
incremental version; complete-state formulation moves queens around
[?]page 64
Real-World Problems

route finding
   travel advisory, computer networks, airline travel

travelling salesperson
   each city must be visited exactly once
   more complex than route finding

VLSI layout
   positioning of gates and connections
   too complex for humans
   crucial for successful operation and costs

robot navigation
   generalization of route finding to continuous space, possibly multi-dimensional (actions involving arms)
Examine possible sequences of actions

**input**
- problem description, initial state

**output**
- solution as an action sequence

**search space**
- set of all possible action sequences
Search

in Artificial Intelligence

search of a problem space
for a solution to a problem
not: search through data structures

basic idea:
find a path from the initial description of a
problem to a description of the solved problem

problem space is created incrementally,
not predefined and already in existence

problem-solving method
powerful technique for many different areas
Problem Space

Representation

**Network**
- graph with nodes as states and arcs as possible steps
- unique representations of states, multiple incoming arcs

**Tree**
- multiple representations of states
Search

different ways to search

**random search**
next step is selected randomly from the possible ones
non-systematic; can’t guarantee complete coverage of the search space; paths may be selected multiple times; may take infinite time

**blind search**
systematic approach; no knowledge about closeness to the solution; complete coverage; ineffective if closeness to solutions can be measured

**directed search**
systematic approach; paths leading towards the solution are preferred
Search Methods

used in AI problems

**depth-first**
- blind, systematic
- expands each path to the end, backtracking
- when a dead end is encountered

**breadth-first**
- blind, systematic
- all nodes at one level are expanded
- finds the shortest path

**beam search**
- directed, heuristic variation of breadth-first
- only a limited number of nodes are expanded
- all successor nodes are evaluated, the best ones are selected for expansion
hill-climbing
   directed variation of depth-first
   successor node with the greatest progress
   towards the goal is selected
   problems: local maxima, plateaus, ridges

branch and bound
   directed search
   most promising node in the tree is selected
   finds the shortest path
   problem: significant portion of the search tree
   must be expanded

best-first
   directed, heuristic search algorithm
   requires estimate of the distance to the solution
   selects the node with the smallest estimate
   problem: does not take into account the length
   of already expanded parts of the paths
A* (A-Star) combination of best-first and branch and bound requires estimate of the distance to the solution uses estimate and previous path length to calculate the cost if estimates are always greater than zero but never greater than the actual cost, the lowest cost path will be found reduces the number of nodes expanded by best-first
Solution

Action sequence that satisfies the goal

validation
Does the solution really achieve the goal?

verification
Is the sequence of actions admissible?

feasibility
With the available resources, can the sequence of actions be carried out?

execution
actions are carried out
# Summary

**Problem-Solving Agents**

- **goal formulation**
  - objectives that need to be achieved

- **problem formulation**
  - actions and states to consider

- **problem types**
  - single-/multiple state, contingency, exploration

- **example problems**
  - toy problems real-world

- **search**
  - strategies

- **solution**
  - execution of the action sequence