AI Session 3

Jan 29, 2021

Agenda

- Recap
- Continuing a bit on graph search algorithms etc.
- Using data and knowledge, reasoning

Recap

- Searching solutions to problems
  - Various strategies in
    - uninformed case
    - informed case (heuristics)
  - The environment, the problem, the time/space requirements, and the requirements for a solution affect the selection of the algorithm to be used
  - Local optimization methods can lead to good (enough?) results with modest space requirements

Situation

- Project topic developments or something else?
- Exercise situation (discussing the tasks)

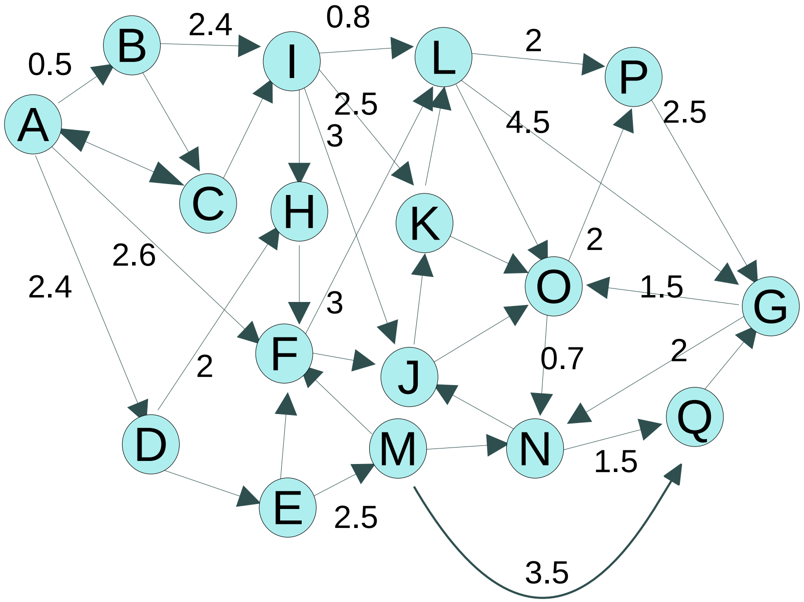
Uniform Cost Search (Dijkstra's)

Evaluation of state *n* by

$$f(n) = g(n)$$

Minimal path cost known so far

Always expand a node minimizing $f(n)$.



Find the shortest route from A to G.
The step costs / edge weights are indicated in the figure. A step cost without an explicit number indication defaults to one.

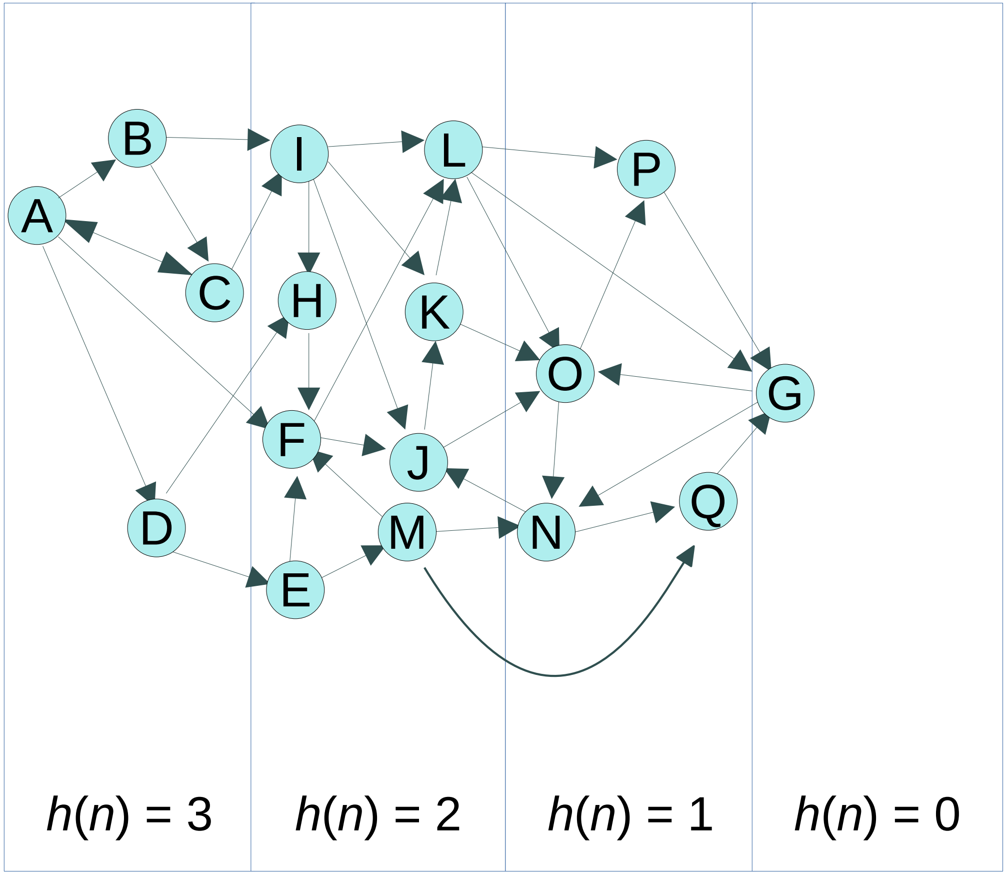
Heuristic-based Best-first Search

Evaluation of state *n* by

$$f(n) = \quad h(n)$$

Heuristic function value

Always expand a node minimizing *f*(*n*).



Find the shortest route from A to G.
The step costs / edge weights are indicated in the figure. A step cost without an explicit number indication defaults to one.

A*

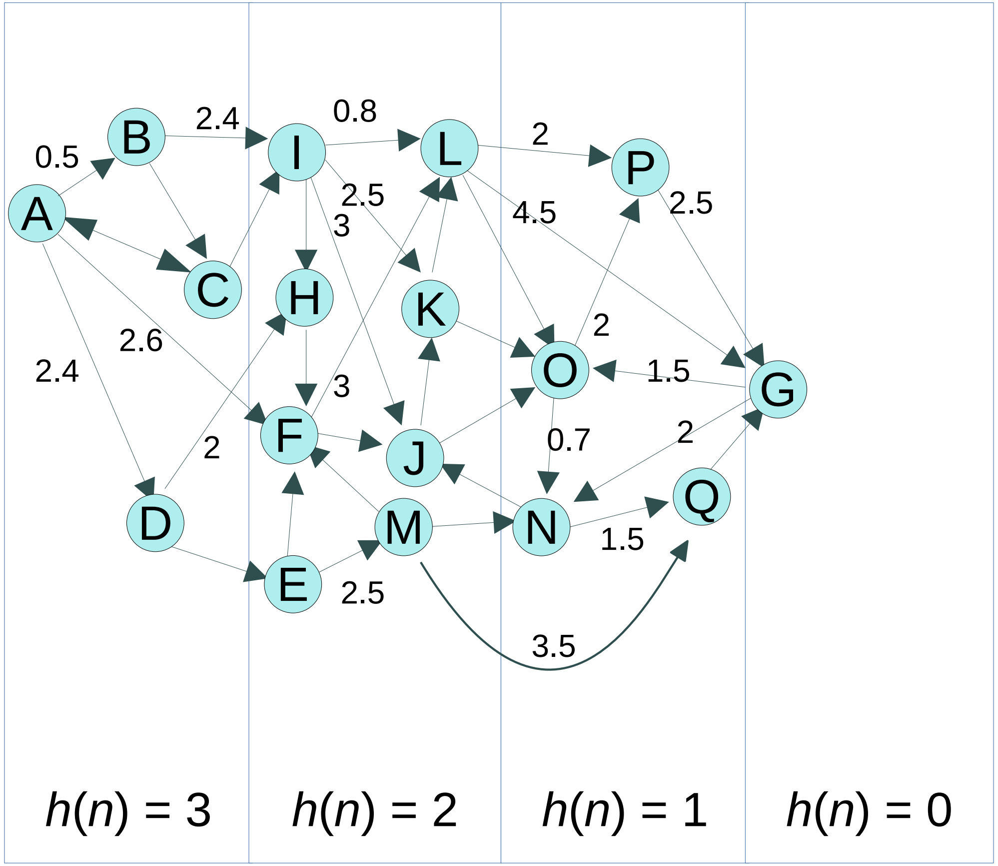
Evaluation of state n by

$$f(n) = g(n) + h(n)$$

Minimal path cost known so far + Heuristic function value

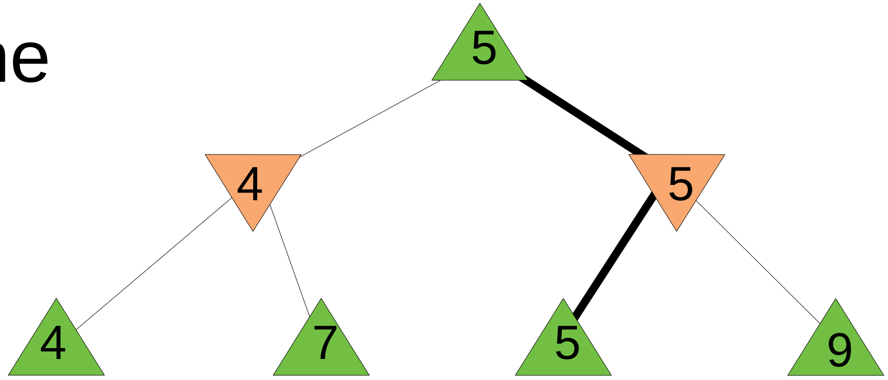
Always expand a node minimizing $f(n)$.

Find the shortest route from A to G.
The step costs / edge weights are indicated in the figure. A step cost without an explicit number indication defaults to one.



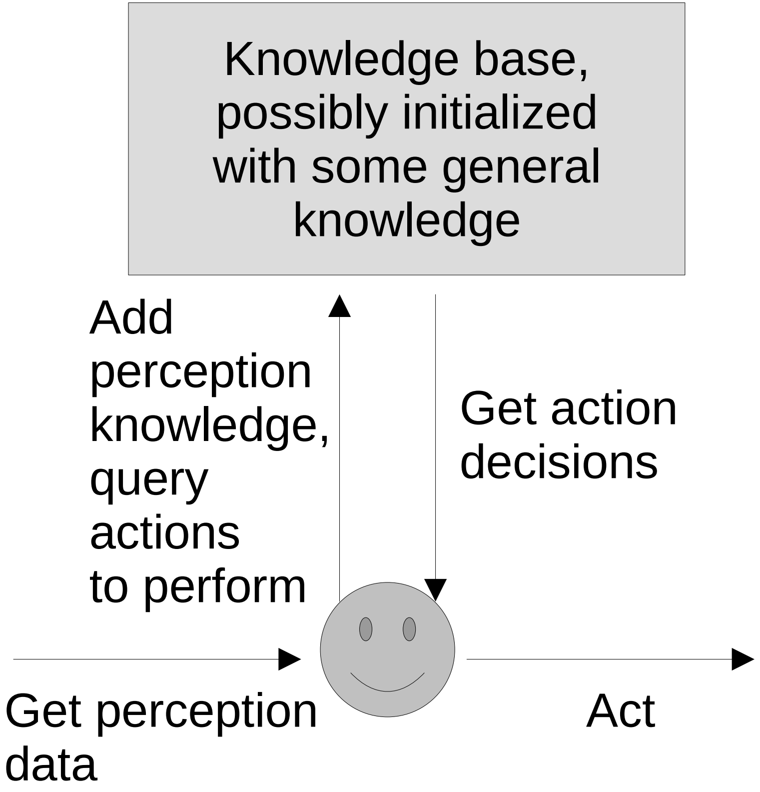
Adversarial Search (Simple Case)

- Minimax algorithm
- alpha–beta pruning
- Typically, cannot evaluate the real utilities (leaf nodes)
  - heuristics



Data and Knowledge

- Often, our agents have lots of *data* available. In order to take advantage of it, it can be refined to *knowledge*.
- Logical agents may use knowledge in *knowledge base*, described as *sentences* in some *knowledge representation* language, to draw conclusions and derive more knowledge.
  - Based on the knowledge, an agent may be able to derive the solution to the problem it should solve
  - Inferring hidden features of states and options may help in partially observable environments



Reasoning: Propositional Logic

- Proposition symbols, logical connectives, parentheses
- In order to determine, whether the knowledge entails some sentence, we can check models taking advantage of logical equivalences and tautologies.
- Are there models evaluating to true (satisfiability problem)?
- Checking entailment by enumerating all the possible models is not viable with large propositional vocabularies
- Reasoning patterns like modus ponens or and-elimination can be used (no need to evaluate all the models)
- So-called *resolution* inference rule combined with a complete search algorithm gives us a complete inference algorithm (requires conjunctive normal form)
- With Horn clauses, forward and backward chaining can be used
- Local search can be useful, when trying to prove satisfiability

A	B		$\neg A$	$A \wedge B$	$A \vee B$	$A \Rightarrow B$	$A \Leftrightarrow B$
false	**false**		**true**	**false**	**false**	**true**	**true**
false	**true**		**true**	**false**	**true**	**true**	**false**
true	**false**		**false**	**false**	**true**	**false**	**false**
true	**true**		**false**	**true**	**true**	**true**	**true**

$\boldsymbol{\alpha}$ entails $\boldsymbol{\beta}$ iff
$(\boldsymbol{\alpha} \wedge \neg\boldsymbol{\beta})$ is unsatisfiable!

We know:
$R \Rightarrow W$
W
$D \vee W$

Can we, based on this, deduce R?

Reasoning: First-order Logic

- More expressive than propositional logic
  - world has objects with *relations* (that hold or do not hold), not just “facts”

Cat(Garfield)

Normal(Garfield)

$\forall$ x [Cat(x) $\wedge$ Normal(x) $\Rightarrow$ $\exists$ y Tail(y, x)]

Reasoning: First-order Logic

- Quantifiers can be got rid of by *universal instantiation* (substitutions by ground terms) end *existential instantiation* (skolemization)
  - We can apply propositional logic inference (slow)
- Unification (substitutions to variables) can boost performance
  - Generalized modus ponens, forward and backward chaining
- If knowledge base in conjunctive normal form, the generalized resolution rule can be used to achieve a complete proof system

Logic Programming

- A formal logic-based programming paradigm
- A program: facts and rules over some domain
  - can be queried
- Prolog
  - (Still) the most popular logic programming language
  - Based on (headless and headed) Horn clauses
  - Used in, e.g., IBM Watson
  - https://en.wikipedia.org/wiki/Prolog
  - http://www.gprolog.org/
  - https://www.swi-prolog.org/ (try online: https://swish.swi-prolog.org/)

```
cat(nermal).
likes(garfield, odie).

?- likes(X, odie).
```

Exercises

It is time to embark upon the challenges of this week (check Moodle)!