FIT5047 Week 1

Agent, Search

授课老师：Joe

AGENTS

FIT5047平时班 – Week 1

An agent is anything that can be viewed as perceiving its environment through sensors, and acting upon that environment through actuators

Human agent:
– eyes, ears and other organs for sensors
– hands, legs, mouth and other body parts for actuators

Robotic agent:
– cameras and infrared range finders for sensors
– various motors for actuators

Rationality

For each possible percept sequence, a rational agent should select an action that is expected to maximize its performance measure, given the evidence provided by the percept sequence and the agent's built-in knowledge

FIT5047平时班 – Week 1

Task environment - PEAS

- **PEAS**
  - – Performance measure
  - – Environment
  - – Actuators
  - – Sensors

FIT5047平时班 – Week 1

Automated taxi driver:

- **Performance measure**

– Safe, fast, legal, comfortable trip, minimize fuel consumption, maximize profit
- **Environment**

– Road types, road contents, customers, operating conditions
- **Actuators**

– Control over the car, interfaces for informing other vehicles and informing passengers
- **Sensors**

– Cameras, sonar, speedometer, GPS, odometer, engine sensors, interface for receiving information from other vehicles and passengers (e.g., speech recognizer)

FIT5047平时班 – Week 1

Internet shopping agent:

- **Performance measure**

– cheap, good quality, appropriate product

- **Environment**

– current WWW sites, vendors

- **Actuators**

– display to user, follow URL, fill in form

- **Sensors**

– HTML pages (text, graphics, scripts)

FIT5047平时班 – Week 1

Environment Attributes

- **Fully (partially) observable – An agent's sensors give it access to the complete state of the environment at all times**
- **Known (unknown) – An agent knows the “laws” of the environment**
- **Single (multi) agent – An agent operating by itself in an environment**
- **Deterministic (stochastic) – The next state is completely determined by the current state and the action executed by the agent**
- **Episodic (sequential) – The agent's experience is divided into atomic *episodes*. The next episode does NOT depend on previous actions**
  – In each episode an agent perceives a percept and performs a single action
- **Static (dynamic) – The environment is unchanged while an agent is deliberating**
- **Discrete (continuous) – Pertains to number of states, the way time is handled, and number of percepts and actions**
  – E.g., state may be continuous, but actions may be discrete

FIT5047平时班 – Week 1

Agent Type	Action selected based on
Simple reflex	current percept
Model based	+ internal state (world model)
Goal based	+ goal
Utility based	+ utility function
Learning	performance element = above agent + critic + learning element + problem generator (exploratory)

FIT5047平时班 – Week 1

Agent Type	Action
Simple reflex	brake when brake-lights of car in front light up
Model based	+ remember the roads travelled, time, state
Goal based	+ make a plan to reach a destination
Utility based	+ quickest with least petrol consumption
Learning	
performance elem	above agent
+ critic	observes the world & informs learning elem
+ learning element	formulates new driving rules based on the feedback from the critic
+ problem generator	might suggest some driving exercises

Search

FIT5047平时班 – Week 1

Problem formulation

- Initial state
- Actions
- Constraints
- Goal test
- Path cost function

FIT5047平时班 – Week 1

1. **initial state**, **e.g., "at Arad"**
2. **actions**
   – e.g., {Go(Sibiu),Go(Timisoara), … }

   transition model
   – e.g., *Result(In(Arad),Go(Zerind))→In(Zerind)*
3. **constraints – nil**
4. **goal test can be**
   – explicit, e.g., *In(Bucharest)*
   – implicit, e.g., *Checkmate(x)*
5. **path cost (additive)**
   – e.g., sum of distances, number of actions executed
   – *c(s,a,s')* is the step cost of taking action *a* at state *s* to reach state *s'* (assumed to be ≥ 0)

FIT5047平时班 – Week 1

Start

5	4	
6	1	8
7	3	2

End

1	2	3
8		4
7	6	5

FIT5047平时班 – Week 1

- **States**

– Location of each of the 8 tiles in one of the 9 squares

- **Operators**

– Possible moves of blank tile

- **Constraints**

– A tile cannot move out of bounds

- **Goal test**

– Have we reached the goal configuration?

- **Path cost**

– If we want to minimize the number of steps, then cost of 1 per step

Searching strategies

Backtracking – at any point in time, we keep one path only

– If we fail, we go back to the last decision point and **erase the failed path**

– Backtracking occurs when

1. we reach a DEADEND state OR
2. there are no more applicable rules OR
3. we generate a previously encountered state description OR
4. an arbitrary number of rules has been applied without reaching the goal

Searching strategies

Backtracking – at any point in time, we keep one path only

– If we fail, we go back to the last decision point and **erase the failed path**

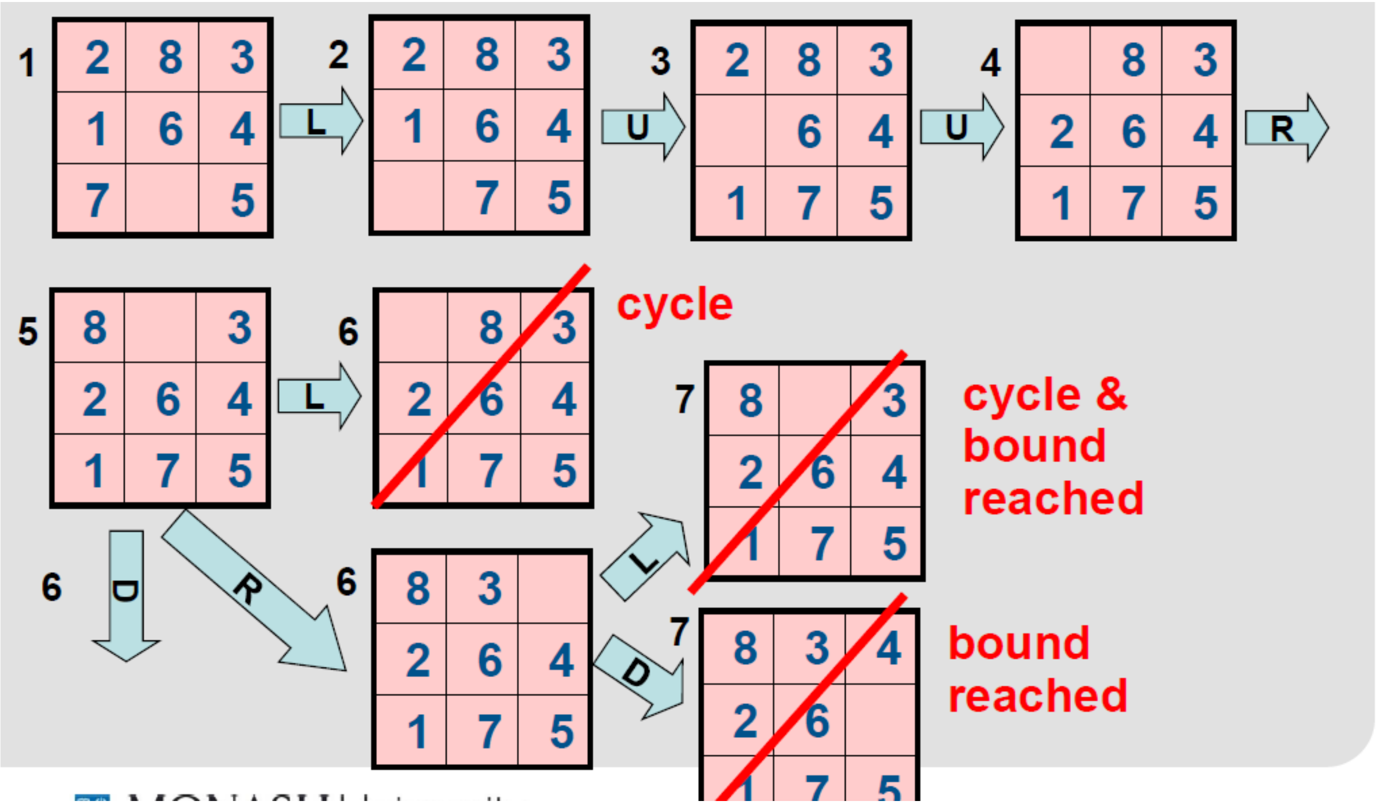
– Backtracking occurs when

1. we reach a DEADEND state OR
2. there are no more applicable rules OR
3. we generate a previously encountered state description OR
4. an arbitrary number of rules has been applied without reaching the goal

Graphsearch – we keep track of several paths simultaneously

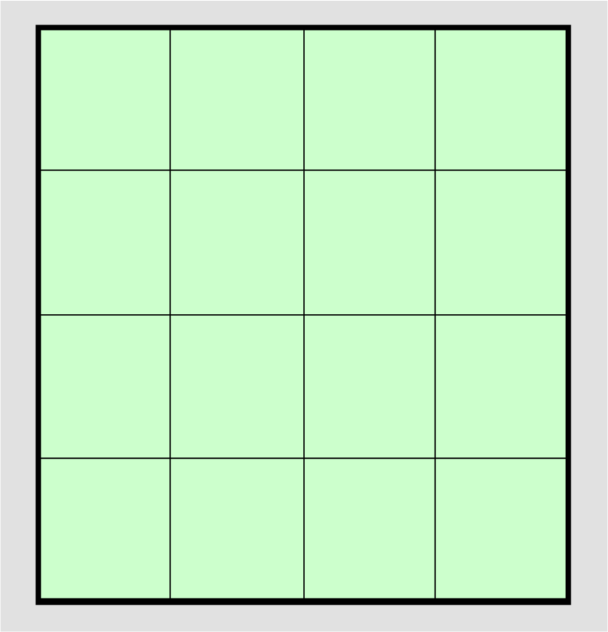
– Done using a structure called a ***search tree/graph***

FIT5047平时班 – Week 1



FIT5047平时班 – Week 1

4 Queens Problem



FIT5047平时班 – Week 1

<u>Tree Search</u>

• Initialize the frontier using the initial state of *problem*

• **Loop**

1. if the frontier is empty **then return** failure

2. choose a leaf node and remove it from the frontier

3. if the node contains a goal state **then return** the corresponding solution

4. expand the chosen node, ***adding*** the resulting nodes to the frontier

end

<u>Graph Search</u>

• Initialize the frontier using the initial state of *problem*

• Initialize the *explored set (**closed**)* to empty

• **Loop**

1. if the frontier is empty **then return** failure

2. choose a leaf node and remove it from the frontier

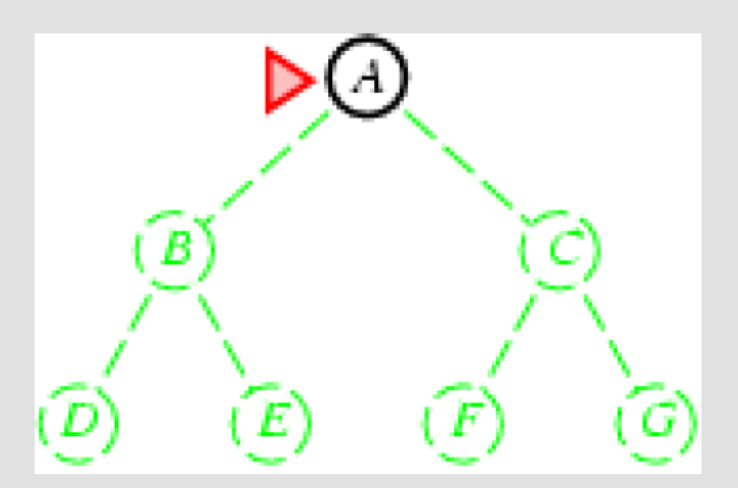
3. if the node contains a goal state **then return** the corresponding solution

4. add the node to the *explored set*

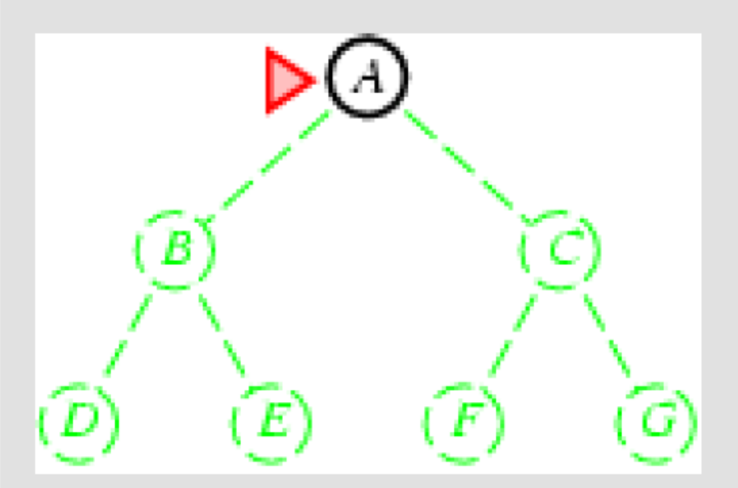
5. expand the chosen node, ***merging*** the resulting nodes
with the frontier or the *explored set*

FIT5047平时班 – Week 1

Breadth first Search



Depth first search

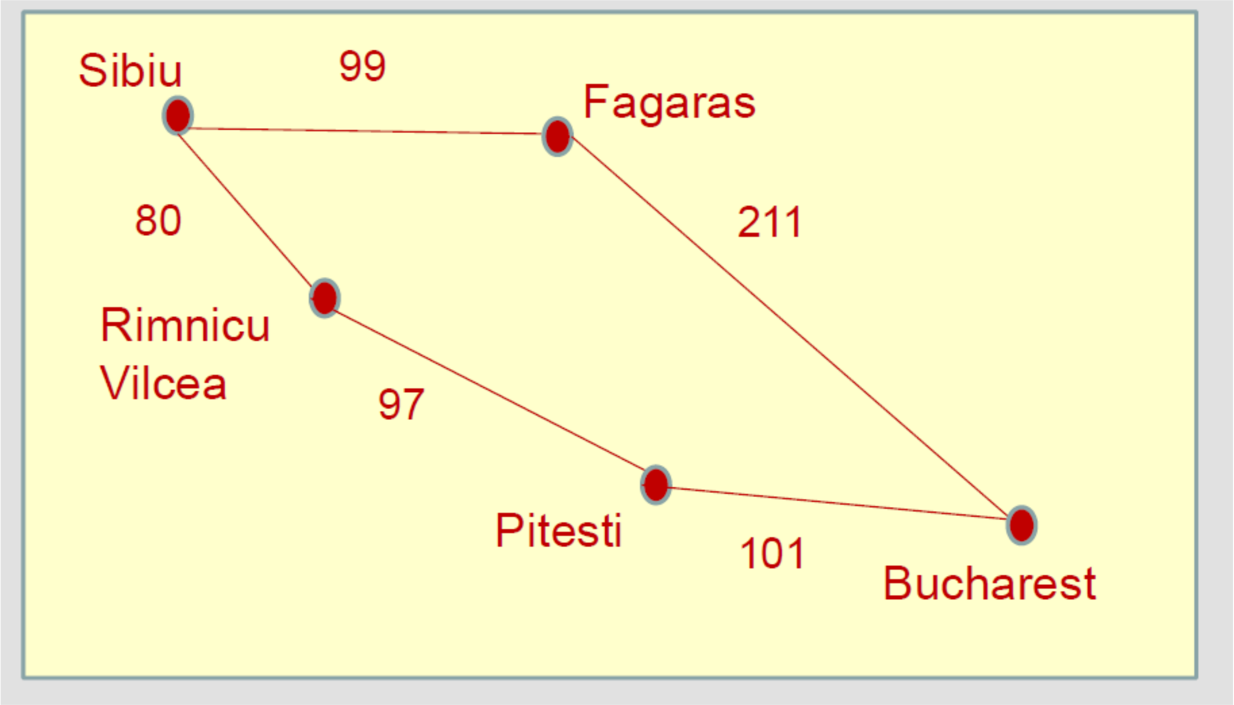


Uniform Cost Search

1. if the frontier is empty **then return** failure
2. choose the lowest-cost node in the frontier and remove it from the frontier
3. if the node contains a goal state **then return** the corresponding solution
4. expand the chosen node
a. if the resulting nodes are not in the frontier **then** **add** them to the frontier
b. else if the resulting nodes are in the frontier ***with higher path cost*** **then** **replace** them with the new nodes

FIT5047平时班 – Week 1

Uniform Cost Search

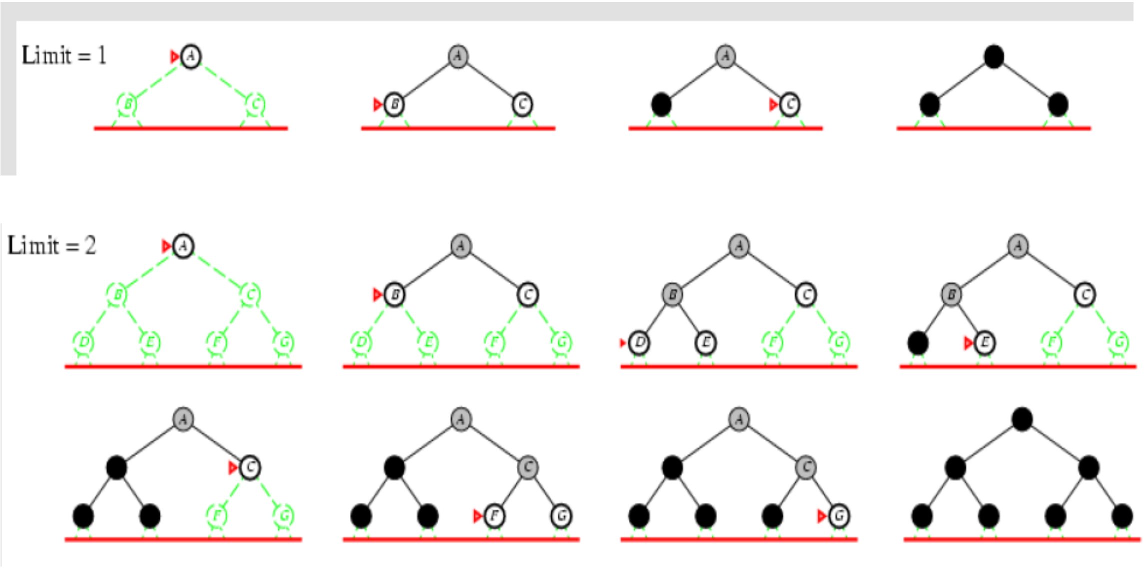


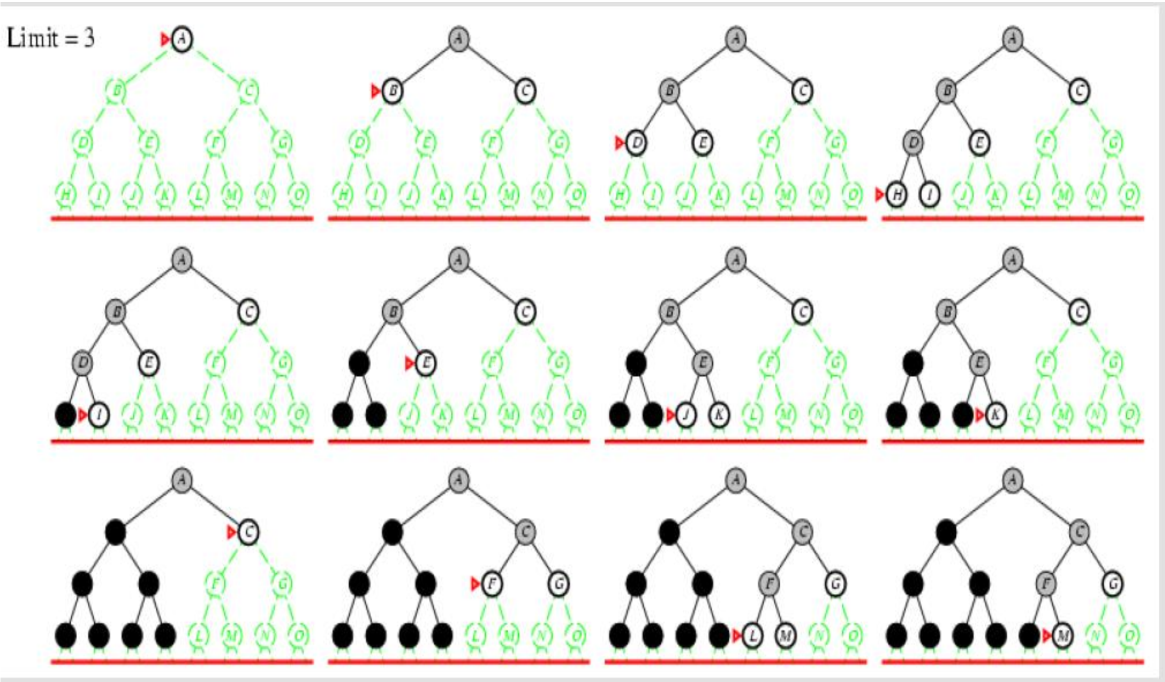
FIT5047平时班 – Week 1

Depth limited search = Depth first search with limited depth

FIT5047平时班 – Week 1

Iterative deepening search





Best First Search

- Heuristic function (prediction of how good that state is), denoted by **<u>h</u>**
- Do not care about the actual cost incurred, only looks at the heuristic when making decisions

Best First Search

- Heuristic function (prediction of how good that state is), denoted by **<u>h</u>**
- Do not care about the actual cost incurred, only looks at the heuristic when making decisions

Algorithm A

• Graphsearch using the evaluation function
f(n) = g(n) + h(n)
• Expands next the node in the frontier with the smallest value of *f(n)*

g(n) = Actual cost incurred so far
h(n) = prediction to goal

Algorithm A*

- *A*
- *Admissibility of h*
- *Monotonicity of h*

FIT5047平时班 – Week 1

<u>Algorithm A*</u>

- *A*
- *Admissibility of h*
- *Monotonicity of h*

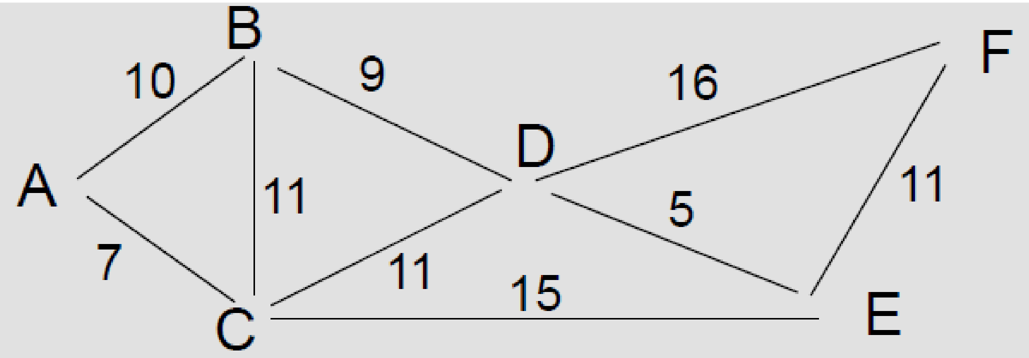
Admisibility of h

- *Never overestimated to true cost of going to the goal from that node.*
- *If h is not admissible, then the path found is not guaranteed to be optimal.*

Monotonicity fo h

- h of parent <= h of child + cost of going from parent to child

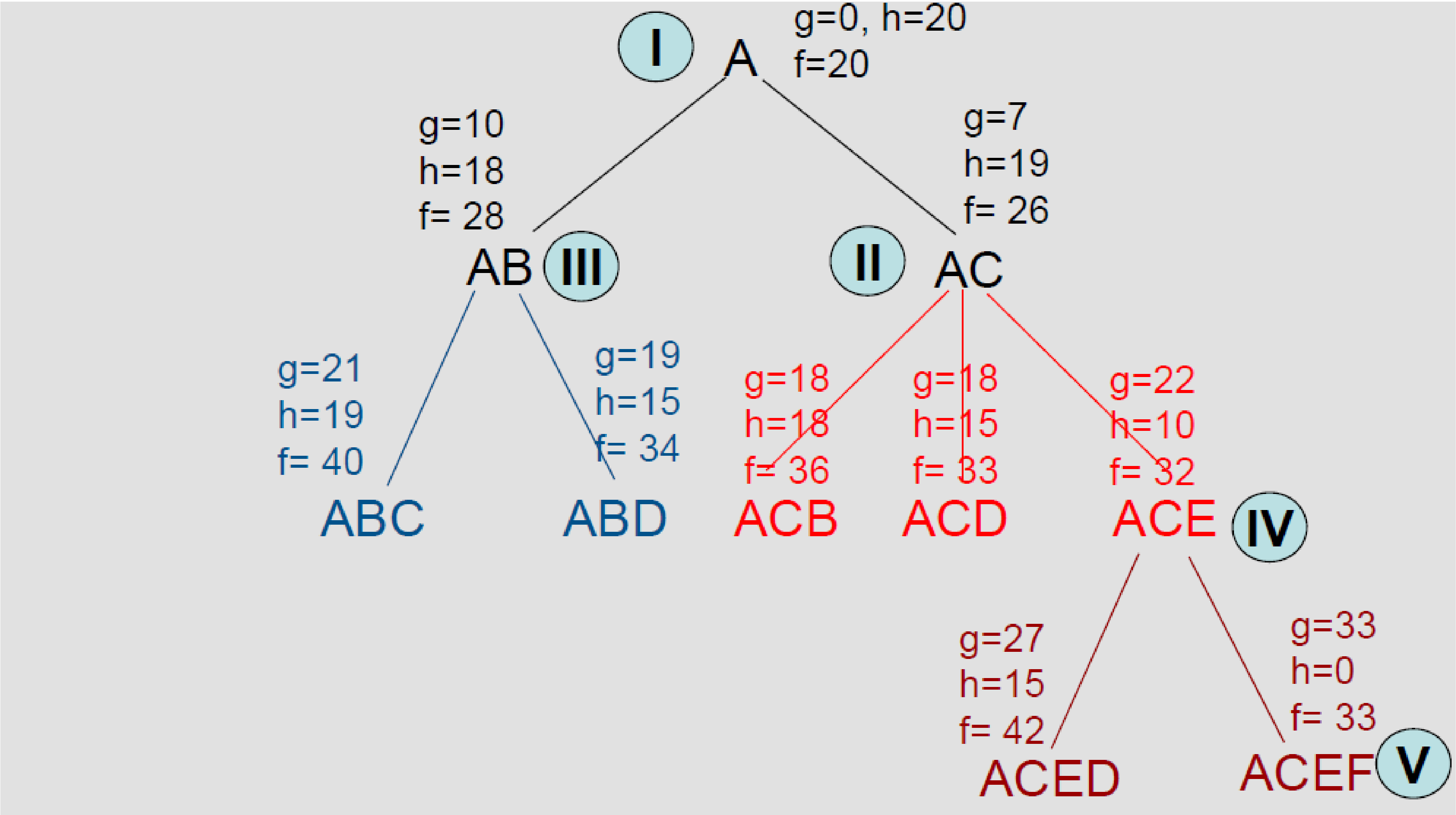
FIT5047平时班 – Week 1



ROAD DISTANCES						
	A	B	C	D	E	F
A		10	7			
B			11	9		
C				11	15	
D					5	16
E						11

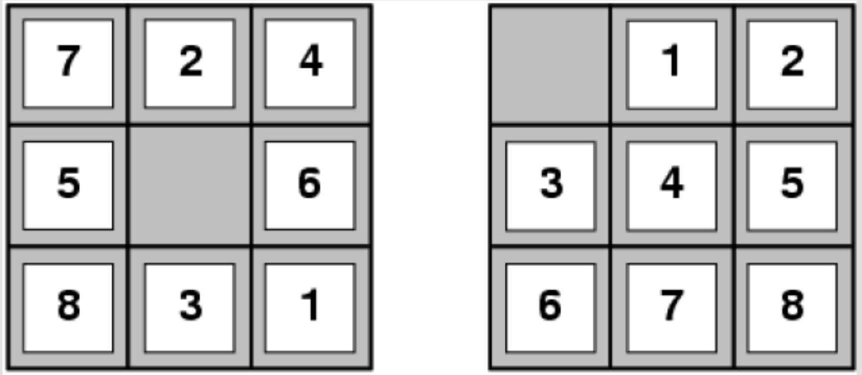
AIR DISTANCES						
	A	B	C	D	E	F
A		4	3	8	12	20
B			6	5	9	18
C				7	10	19
D					5	15
E						10

FIT5047平时班 – Week 1



Admissible heuristics: 8 Puzzle

- $h_1(n)$ = number of misplaced tiles
- $h_2(n)$ = total *Manhattan distance* (# of squares from desired location of each tile)



Start State Goal State

- $h_1(S) = ?$
- $h_2(S) = ?$

Dominance

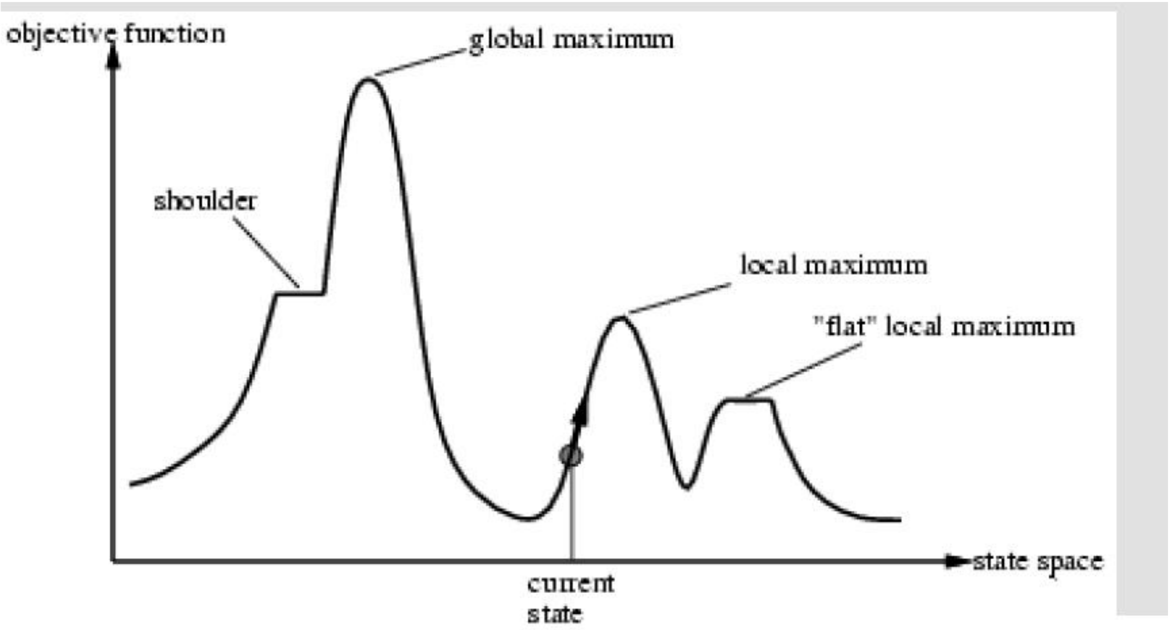
- **Given two admissible heuristics h_1 and h_2, if $h_2(n) \geq h_1(n)$ for all n then h_2 *dominates* h_1**

  → h_2 is better for search

FIT5047平时班 – Week 1

Hill Climbing

- Look around, find the next best move , go to that place
- If there is none, then terminate
- Greedy in nature, since it always find something better



FIT5047平时班 – Week 1

Local Beam Search

- Running multiple hill climbing simultaneously

FIT5047平时班 – Week 1

Simulated Annealing

- Similar to hill climbing, but have a chance of going downhill (accept a worse state)
- The probability of accepting a worse state is controlled by temperature and annealing schedule, which are defined by you
- Pr=e-ΔE/***T***
- When temperature decreases to 0, simulated annealing is just hill climbing

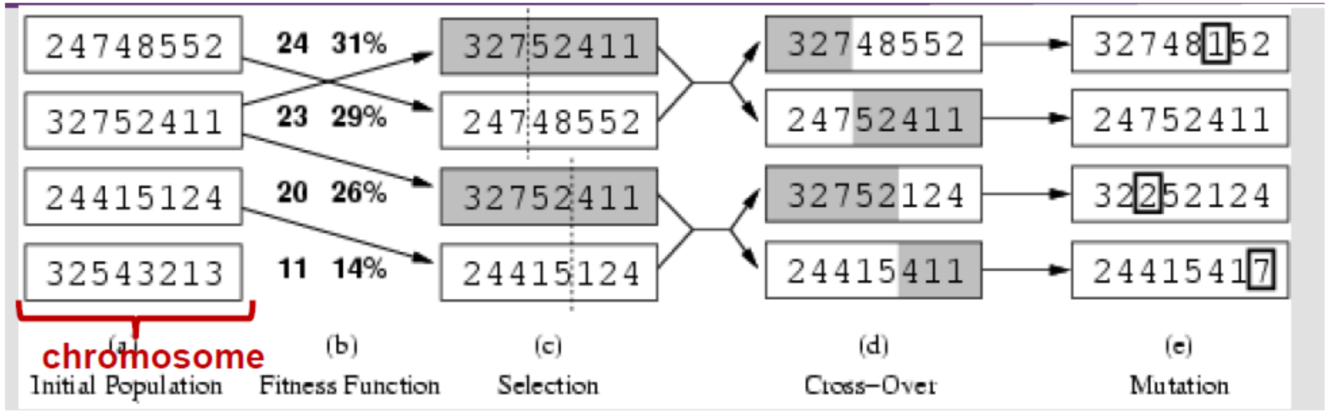
FIT5047平时班 – Week 1

Genetic Algorithm

- The problem is so large and complex, we have no idea where to go

• Start with a population of k randomly generated states (parent)
• A state (chromosome) is represented as a string over a finite alphabet of genes (often a string of 0s and 1s)
• A successor state is generated by combining two parent states
• Evaluation function (fitness function):
– Higher values for better states
• Produce the next generation of states by selection, crossover and mutation

FIT5047平时班 – Week 1



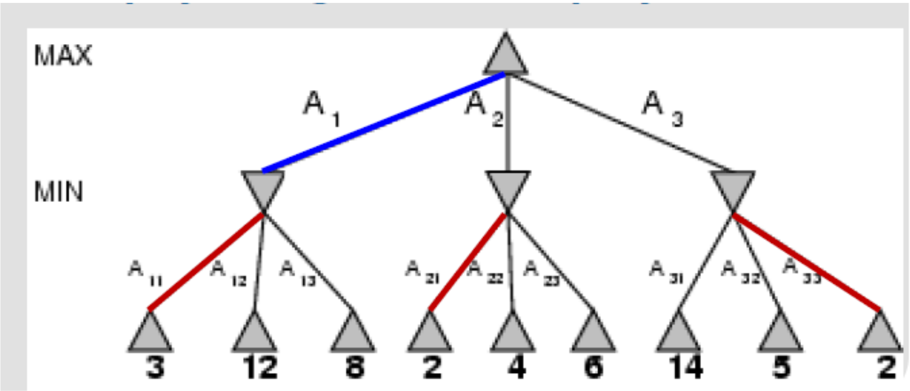
FIT5047平时班 – Week 1

Adversarial Search

Two players

- Min and Max
- Min wants to minimize the score while Max wants to maximize the score

Minimax ideas



FIT5047平时班 – Week 1

Minimax optimization – Alpha and beta pruning