Artificial Intelligence

- The field of **Artificial intelligence**:  
  - The design and study of computer systems that behave intelligently
- **AI:**  
  - Focus on nontrivial problems that require reasoning and are often solved by humans  
  - Goes beyond numerical computations and manipulations
- **Benefits of AI research**  
  - Engineering aspect  
    - solving of hard problems  
  - Cognitive aspect  
    - Understanding the nature of human intelligence
AI applications: Software systems.

- **Diagnosis of**: software, technical components

- **Adaptive systems**
  - Adapt systems to user needs
  - Adapt systems to specific tasks

- **Examples**:
  - Intelligent interfaces
  - Intelligent helper applications
  - Collaborative filtering
  - Target advertising

Search and information retrieval

**Web search engines**
- Improve the quality of search
- Rely on methods/algorithms developed in AI
- Add inferences and knowledge to search queries

**Semantic web (or web 2):**
- From information to knowledge sharing
- Ontology languages
Speech recognition

- **Speech recognition systems:**
  - Systems based on statistical models,
  - Hidden Markov models

- **Multi-user speech recognition**
- Voice command/voice activated devices
  - No training – works for many users

- **Adaptive speech systems**
  - Adapt to the user (training)
  - continuous speech
  - commercially available software – (Nuance, IBM)
    - http://www.nuance.com/

Space exploration

Autonomous rovers,  Analysis of sky
  intelligent probes  Survey data

CS 1571 Intro to AI  M. Hauskrecht

CS 1571 Intro to AI  M. Hauskrecht
AI applications: Medicine

• Medical diagnosis:
  – QMR system. Internal medicine.
• Patient Monitoring and Alerting:
  – Cerner
• Medical imaging
  – Classification of body structures and visualization
• Robotic surgeries

AI applications: Bioinformatics

• Genomics and Proteomics
  – Sequence analysis
  – Prediction of gene regions on DNA
  – Analysis of DNA micro-array and proteomic MS profiles: find genes, proteins (peptides) that characterize a specific disease
  – Regulatory networks
**AI applications: Transportation**

**Autonomous vehicle control:**
- ALVINN (CMU, Pomerleau 1993)
- Series of DARPA challenges (http://www.grandchallenge/)
  - 2004, 2005 Drive across Mojave
  - 2007 - DARPA Urban Challenge
- Google autonomous vehicles
  - Pedestrian detection
  - Traffic monitoring
  - Navigation/route optimizations

**Classification of images or its parts**
Game playing

- **Backgammon**
  - TD-backgammon
    - a program that learned to play at the championship level (from scratch).
    - reinforcement learning

- **Chess**
  - Deep blue (IBM) program
    - (defeated Kasparov in 1997)

- **Bridge, Poker**

Natural language processing

**understanding/annotation of free text**

- **Document analysis:**
  - Automatic classification of articles
  - Content extraction/inference
  - Email SPAM detection

- **IBM’s Watson project**
  - www.ibm.com/watson
  - Successfully competed against the top human players in Jeopardy
Robots

- Robotic toys
  - Sony’s Aibo

- Vacuum cleaners

- Humanoid robot
  - Honda’s ASIMO
    (http://world.honda.com/robot/)

- Military robots

Other application areas

- Handwriting analysis/detection
- Human face detection
- Video stream annotation
- Object tracking
- Music composition, picture drawing
- …
Topics

• Problem solving and search.
  – Formulating a search problem, Search methods, Combinatorial and Parametric Optimization.

• Logic and knowledge representations.
  – Logic, Inference

• Planning.
  – Situation calculus, STRIPS, Partial-order planners,

• Uncertainty.
  – Modeling uncertainty, Bayesian belief networks, Inference in BBNs, Decision making in the presence of uncertainty.

• Machine Learning
  – Supervised learning, unsupervised learning, Basic learning models

CS 1571 Introduction to AI
Lecture 2

Problem solving by searching

Milos Hauskrecht
milos@cs.pitt.edu
5329 Sennott Square
Example

- Assume a problem of solving a linear equation

\[ 3x + 2 = 11 \]

Do you consider it a challenging problem?

Example

- Assume a problem of computing the roots of the quadratic equation

\[ 3x + 2 = 11 \]

Do you consider it a challenging problem?

Hardly, we just apply the ‘standard’ formula or procedure to solve:

\[ ax + b = c \]
\[ x = (c - b) / a \]
\[ x = 3 \]
Solving problems by searching

• Some problems have a straightforward solution
  – Just apply a known formula, or follow a standardized procedure
    Example: solution of the linear or quadratic equations
  – Hardly a sign of intelligence

• More interesting problems do not have a straightforward solution, and they require search:
  – more than one possible alternative needs to be explored before the problem is solved
  – the number of alternatives to search among can be very large, even infinite

Search example: Route finding

• Find a route (path) from one city to another city
Example. Traveler problem

- Another flavor of the traveler problem:
  - find the route with the minimum length between S and T

Example. Puzzle 8.

- Find the sequence of move of tiles from the initial game position to the designated target position

<table>
<thead>
<tr>
<th>Initial position</th>
<th>Goal position</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Initial Position" /></td>
<td><img src="image2" alt="Goal Position" /></td>
</tr>
</tbody>
</table>

Find a configuration of n queens on an n x n board such that queens do not attack each other

A goal configuration

A bad configuration

A search problem

is defined by:

- **A search space:**
  - The set of objects among which we search for the solution
  - **Example:** routes connecting two cities, or the different N-queen configurations

- **A goal condition**
  - What are the characteristics of the object we want to find in the search space?
  - **Examples:**
    - Path between cities A and B
    - Path between A and B with the smallest number of links
    - Path between A and B with the shortest distance
    - Non-attacking n-queen configuration
Search

- Search (process)
  - The process of exploration of the search space
- The efficiency of the search depends on:
  - The search space and its size
  - Method used to explore (traverse) the search space
  - Condition to test the satisfaction of the search objective
    (what it takes to determine I found the desired goal object)
Search

- **Search (process)**
  - The process of exploration of the search space

- **The efficiency of the search depends on:**
  - The search space and its size
  - Method used to explore (traverse) the search space
  - **Condition to test the satisfaction of the search objective**
    (what it takes to determine I found the desired goal object)

---

**Search**

- **Search (process)**
  - The process of exploration of the search space

- **Important**
  - We can often influence the efficiency of the search !!!!
  - We can be smart about choosing the *search space*, the *exploration policy*, and the *design of the goal test*
Graph representation of a search problem

- Search problems can be often represented using graphs
- **Typical example: Path finding**
  - Map corresponds to the graph, nodes to cities, links valid moves via available connections
  - **Goal:** find a path (sequence of moves) in the graph from S to T

![Graph representation of a search problem](image)

Graph search

- **Less obvious conversion:**
  **Puzzle 8.** Find a sequence of moves from the initial configuration to the goal configuration.
  - nodes corresponds to states of the game,
  - links to valid moves made by the player

![Graph search](image)
Graph search problem

**Four components:**
- **States** - game positions, or locations on the map that are represented by nodes in the graph
- **Operators** - valid moves
- **Initial state** – start position, start city
- **Goal state** – target position (positions), target city (cities)

```
start
```

```
S T A B C D E F G H I J
```

Graph search

- **More complex versions of the graph search problems:**
  - Find the minimal length path
    (= a route with the smallest number of connections, the shortest sequence of moves that solves Puzzle 8)

```
start
```

```
S T A B C D E F G H I J
```
Graph search

- More complex versions of the graph search problems:
  - Find the minimum cost path
    (= a path with the shortest distance)

N-queens

Some problems are easy to convert to the graph search problems

- But some problems are harder and less intuitive
  - Take e.g. N-queens problem.

Goal configuration

- Problem:
  - We look for a configuration, not a sequence of moves
  - No distinguished initial state, no operators (moves)
N-queens

How to choose the search space for N-queens?

• Ideas?
  
  Search space:
  – all configurations of N queens on the board

• Can we convert it to a graph search problem?
  
  We need states, operators, initial state and goal condition.

States are: N-queen configurations
Initial state: ?
Operators (moves)?
N-queens

Search space:
- all configurations of N queens on the board

- Can we convert it to a graph search problem?
- We need states, operators, initial state and goal condition.

Initial state: an arbitrary N-queen configuration
Operators (moves): change a position of one queen

N-queens

Is there an alternative way to formulate the N-queens problem as a search problem?

- Ideas?
N-queens

Is there an alternative way to formulate the N-queens problem as a search problem?

- **Search space:** configurations of 0, 1, 2, … N queens
- **Graph search:**
  - States: configurations of 0, 1, 2,…N queens
  - Operators: additions of a queen to the board
  - Initial state: no queens on the board

Graph search

N-queens problems

- This is a different graph search problem when compared to Puzzle 8 or Path planning:
  
  We want to find only the target configuration, not a path
Two types of graph search problems

- **Path search**
  - Find a path between states S and T
  - **Example**: traveler problem, Puzzle 8
  - **Additional goal criterion**: minimum length (cost) path

- **Configuration search (constraint satisfaction search)**
  - Find a state (configuration) satisfying the goal condition
  - **Example**: n-queens problem
  - **Additional goal criterion**: “soft” preferences for configurations, e.g. minimum cost design

Graph Search Problem

Search problems that can be often represented or converted into a graph search problems:

- **Initial state**
  - State (configuration) we start to search from (e.g. start city, initial game position)

- **Operators**:
  - Transform one state to another (e.g. valid connections between cities, valid moves in Puzzle 8)

- **Goal condition**:
  - Defines the target state (destination, winning position)

- **Search space** (the set of objects we search for the solution):
  - is now defined indirectly through:
    - **the initial state + operators**