Discussion of feedback

• Pace
  – 2 “fine”, 1 “could go faster”
  – 2: recap too long, 3: “sometimes rushed later (as a consequence)”

• Coloured card questions
  – Some more explanation would be good
  – More consistent: get everyone to vote”

• Which parts are most important?
  – Definitions + algorithms. Examples are for illustration

• Hard concepts:
  – Arc consistency: today + work in Alspace + practice exercise
  – Alternative formulation of CSP as graph search: after class
Discussion of feedback

• Midterm: review & sample questions?
  – Midterm date confirmed: Mon, Feb 28, 3pm (1 to 1.5 hours, TBD)
  – Sample midterm has been on WebCT for ~2 weeks
    • Topics: everything up to (including all of) CSP, plus planning (partially or all of it, TBD closer to midterm)
    – Should we do a midterm review session?

• More explanation of practice exercises?
  – I’ll show where they are in WebCT
  – If you have trouble with them, please come to office hours

• How will what we learn eventually be applied in making an intelligent agent?
  – Game AI: lots of search
  – Reasoning under constraints is core to making intelligent decisions
    • With CSPs, we’re right in the middle of it!
We’ll now focus on CSP.
Lecture Overview

Arc consistency
- Recap
- Complexity analysis
- Domain Splitting

• Intro to Local Search
Definition:
An arc \( <x, r(x,y)> \) is **arc consistent** if for each value \( x \) in \( \text{dom}(X) \) there is some value \( y \) in \( \text{dom}(Y) \) such that \( r(x,y) \) is satisfied.

A network is arc consistent if all its arcs are arc consistent.
Arc Consistency

Arc consistent:
For each value in \( \text{dom}(C) \), there is one in \( \text{dom}(A) \) that satisfies \( A > C \) (namely \( A = 3 \))

Not arc consistent:
No value in domain of \( B \) that satisfies \( A < B \) if \( A = 3 \)
Arc Consistency

Not arc consistent anymore:
For C=2, there is no value in dom(A) that satisfies A>C
Which arcs need to reconsider?

- When we reduce the domain of a variable $X$ to make an arc $\langle X, c \rangle$ arc consistent, which arcs do we need to reconsider?

  - You do not need to reconsider other arcs
    - If an arc $\langle X, c' \rangle$ was arc consistent before, it will still be arc consistent
    - Nothing changes for arcs of constraints not involving $X$

- every arc $\langle Z, c' \rangle$ where $c' \neq c$ involves $Z$ and $X$: 

\begin{itemize}
  \item $Z_1 \rightarrow c_1 \rightarrow X \rightarrow c \rightarrow Y \rightarrow c_4 \rightarrow A$
  \item $Z_2 \rightarrow c_2 \rightarrow X \rightarrow c \rightarrow Y$
  \item $Z_3 \rightarrow c_3 \rightarrow X \rightarrow c \rightarrow Y$
\end{itemize}
Lecture Overview

• Arc consistency
  - Recap
  - Complexity analysis
  - Domain Splitting

• Intro to Local Search
Arc Consistency Algorithm: Complexity

- Worst-case complexity of arc consistency procedure on a problem with N variables
  - let $d$ be the max size of a variable domain
  - let $c$ be the number of constraints

- How often will we prune the domain of variable $V$? $O(d)$ times
- How many arcs will be put on the ToDoArc list when pruning domain of variable $V$?
  - $O(\text{degree of variable } V)$
  - In total, across all variables: sum of degrees of all variables = …
    - $2^{\text{number of constraints, i.e. } 2c}$
  - Together: we will only put $O(dc)$ arcs on the ToDoArc list
  - Checking consistency is $O(d^2)$ for each of them

- Overall complexity: $O(cd^3)$
- Compare to $O(d^N)$ of DFS!! Arc consistency is MUCH faster
Lecture Overview

• Arc consistency
  - Recap
  - Complexity analysis
  Domain Splitting

• Intro to Local Search
Can we have an arc consistent network with no solution?

YES

NO

• Example: vars A, B, C with domain \{1, 2\} and constraints A \neq B, B \neq C, A \neq C

• Or see Alspace CSP applet Simple Problem 2
Domain splitting (or case analysis)

• Arc consistency ends: Some domains have more than one value → may or may not have a solution
  A. Apply Depth-First Search with Pruning or
  B. **Split the problem** in a number of disjoint cases:

  CSP with \( \text{dom}(X) = \{x_1, x_2, x_3, x_4\} \) becomes

  \[\text{CSP}_1 \text{ with } \text{dom}(X) = \{x_1, x_2\} \text{ and } \text{CSP}_2 \text{ with } \text{dom}(X) = \{x_3, x_4\}\]

• Solution to CSP is the **union** of solutions to \( \text{CSP}_i \)
Whiteboard example for domain splitting

- ...
Domain splitting

• Each smaller CSP is easier to solve
  – Arc consistency might already solve it
• For each subCSP, which arcs have to be on the ToDoArcs list when we get the subCSP by splitting the domain of X?

- arcs <Z, r(Z,X)>
- arcs <Z, r(Z,X)> and <X, r(Z,X)>
- All arcs
Domain splitting in action

- Trace it on “simple problem 2”
Searching by domain splitting

CSP, apply AC

If domains with multiple values

Split on one

CSP₁, apply AC

If domains with multiple values
Split on one

CSPₙ, apply AC

If domains with multiple values.....Split on one

How many CSPs do we need to keep around at a time?
With depth m and b children at each split: \(O(bm)\). It's a DFS
Learning Goals for today’s class

• Define/read/write/trace/debug the arc consistency algorithm. Compute its complexity and assess its possible outcomes

• Define/read/write/trace/debug domain splitting and its integration with arc consistency

• Assignment 1 is due on Monday
• Local search practice exercise is on WebCT
• Programming assignment (part of assignment #2) is available on WebCT (due Wednesday, Feb 23rd)

• Coming up: local search, Section 4.8