CSIT113
Problem Solving

Week 2b

Brute Force vs. Finesse

- Often, the “easy” way to solve a problem is to list all possible answers and select the best (right) one.
- However, this will generally involve far more work than is strictly needed.
- The following problems will illustrate this idea.

Problem 1: Crossing the river - v1

- A **Farmer** has a **wolf**, a **goat** and a **cabbage** and must cross a river using a small boat.
- Only two things will fit in the boat at a time.
- If left alone, the wolf will eat the goat.
- If left alone, the goat will eat the cabbage.
- **Get everything safely across the river.**
- What are start state, final state, operators, constraints?

Problem 1

Brute Force approach:

1) list all possible configurations;
2) eliminate the illegal ones;
3) find a sequence of configurations starting at the initial state and ending at the goal state.

Invariants (constants, pattern)

- **Goat will eat cabbage**:
  - $F = G = C \vee G \neq C$
  - In other words either the farmer, goat and cabbage are all on the same bank ($f = g = c$) or the goat and cabbage are on different banks ($g \neq c$).
- **Wolf will eat goat**
  - $F = W = G \vee W \neq G$
  - In other words either the farmer, wolf and goat are all on the same bank ($f = w = g$) or the wolf and the goat are on different banks ($w \neq g$)

  F,G,C,W: variables

Brute Force

- List all possible configurations. How many?

F																
W																
G																
C																

Brute Force

- List all possible configurations.

F	l															
W	l															
G	l															
C	l															

- l for left bank, r for right bank

Brute Force

- List all possible configurations.

F	l	r														
W	l	l														
G	l	l														
C	l	l														

Brute Force

- List all possible configurations.

F	l	r	l													
W	l	l	r													
G	l	l	l													
C	l	l	l													

Brute Force

- List all possible configurations.

F	l	r	l	l												
W	l	l	r	l												
G	l	l	l	r												
C	l	l	l	l												

Brute Force

- List all possible configurations.

F	l	r	l	l	l											
W	l	l	r	l	l											
G	l	l	l	r	l											
C	l	l	l	l	r											

Brute Force

- List all possible configurations.

F	l	r	l	l	l	l										
W	l	l	r	l	l	l										
G	l	l	l	r	l	r										
C	l	l	l	l	r	r										

Brute Force

- List all possible configurations.

F	l	r	l	l	l	l	l									
W	l	l	r	l	l	l	r									
G	l	l	l	r	l	r	l									
C	l	l	l	l	r	r	r									

Brute Force

- List all possible configurations.

F	l	r	l	l	l	l	l	l								
W	l	l	r	l	l	l	r	r								
G	l	l	l	r	l	r	l	r								
C	l	l	l	l	r	r	r	l								

Brute Force

- List all possible configurations.

F	l	r	l	l	l	l	l	l	r							
W	l	l	r	l	l	l	r	r	l							
G	l	l	l	r	l	r	l	r	l							
C	l	l	l	l	r	r	r	l	r							

Brute Force

- List all possible configurations.

F	l	r	l	l	l	l	l	l	r	r						
W	l	l	r	l	l	l	r	r	l	l						
G	l	l	l	r	l	r	l	r	l	r						
C	l	l	l	l	r	r	r	l	r	l						

Brute Force

- List all possible configurations.

F	l	r	l	l	l	l	l	l	r	r	r					
W	l	l	r	l	l	l	r	r	l	l	r					
G	l	l	l	r	l	r	l	r	l	r	l					
C	l	l	l	l	r	r	r	l	r	l	l					

Brute Force

- List all possible configurations.

F	l	r	l	l	l	l	l	l	r	r	r	r				
W	l	l	r	l	l	l	r	r	l	l	r	r				
G	l	l	l	r	l	r	l	r	l	r	l	r				
C	l	l	l	l	r	r	r	l	r	l	l	l				

Brute Force

- List all possible configurations.

F	l	r	l	l	l	l	l	l	r	r	r	r	r			
W	l	l	r	l	l	l	r	r	l	l	r	r	r			
G	l	l	l	r	l	r	l	r	l	r	l	r	l			
C	l	l	l	l	r	r	r	l	r	l	l	l	r			

Brute Force

- List all possible configurations.

F	l	r	l	l	l	l	l	l	r	r	r	r	r	r		
W	l	l	r	l	l	l	r	r	l	l	r	r	r	l		
G	l	l	l	r	l	r	l	r	l	r	l	r	l	r		
C	l	l	l	l	r	r	r	l	r	l	l	l	r	r		

Brute Force

- List all possible configurations.

F	l	r	l	l	l	l	l	l	r	r	r	r	r	r	l	
W	l	l	r	l	l	l	r	r	l	l	r	r	r	l	r	
G	l	l	l	r	l	r	l	r	l	r	l	r	l	r	r	
C	l	l	l	l	r	r	r	l	r	l	l	l	r	r	r	

Brute Force

- List all possible configurations.

F	l	r	l	l	l	l	l	l	r	r	r	r	r	r	l	r
W	l	l	r	l	l	l	r	r	l	l	r	r	r	l	r	r
G	l	l	l	r	l	r	l	r	l	r	l	r	l	r	r	r
C	l	l	l	l	r	r	r	l	r	l	l	l	r	r	r	r

- Can you see the symmetry?

Brute Force

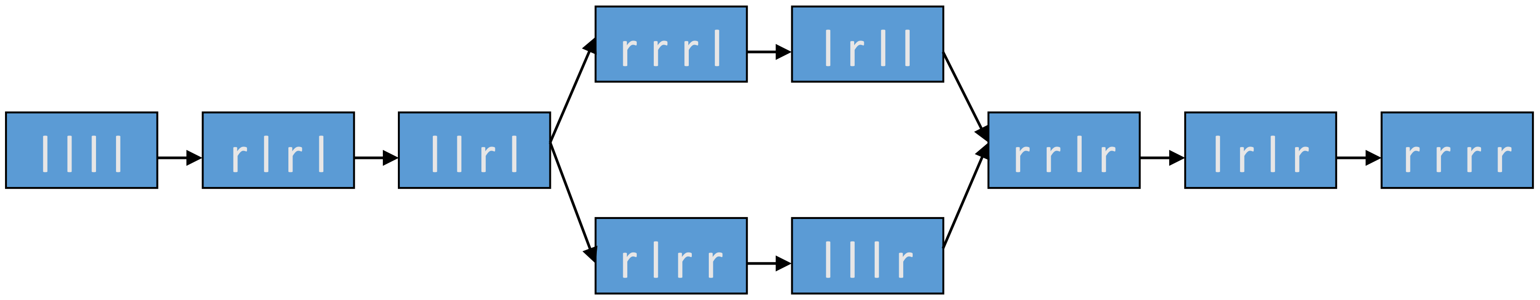
- Eliminate the illegal ones.

F	l	r	l	l	l	l	l	l	r	r	r	r	r	r	l	r
W	l	l	r	l	l	l	r	r	l	l	r	r	r	l	r	r
G	l	l	l	r	l	r	l	r	l	r	l	r	l	r	r	r
C	l	l	l	l	r	r	r	l	r	l	l	l	r	r	r	r

- 10 states are left

Brute Force

- Find a sequence of configurations starting at the initial state and ending at the goal state.



vs. Finesse

- We note that the wolf and cabbage are not a threat to each other.
- The goat is a problem for both.
- Rephrase the problem:

Abstraction: Crossing the river – ver1a

- A Farmer has two alphas and a beta and must cross a river using a small boat.
- Only two things will fit in the boat at a time.
- An alpha and a beta may never be left alone together.
- Get everything safely across the river.

Abstraction

- Note: the single invariant is now:
  - $f = \alpha = b \vee \alpha \neq b$
- The solution should now be obvious.

Dilbert: 27th November 2013

Problem 2: Crossing the river – ver2

- Three couples (husband and wife) wish to cross a river using a small boat.
- Only two people will fit in the boat at a time.
- Each husband is too jealous to leave his wife with another man.
- Get everyone safely across the river.

Jealous Husbands

Brute Force approach:

1) list all possible configurations;
2) eliminate the illegal ones;
3) find a sequence of configurations starting at the initial state and ending at the goal state.

Brute Force

- How many possible configurations are there?
- This is getting silly.
- We call this the “State Space Explosion”
- Clearly the brute force approach quickly becomes unattractive.
- So we need to find a finesse solution.

What’s the Problem?

- We can look at the problem in more than one way.
  - 3 couples must cross.
  - 3 wives and 3 husbands must cross.
  - 6 people must cross.
- Some ways are more useful than others.

What’s the Problem

- Also, multiple possible strategies suggest themselves:
  - Get all the wives across first;
  - Get all the husbands across first;
  - Get one couple across at a time.
- Can we make use of symmetry and solve only half the problem?

State Representation

- We need a good notation to represent positions (states) in the problem.
- We do not need to identify individual people.
- We have three types of thing to deal with.
  - Couples.
  - Husbands.
  - Wives.

State Representation

Thus:

- 3c represents 3 couples on one bank.
- 2ch represents 2 couples and a husband (the wife is on the other bank).
- 2w represents two wives (the other bank will have 2ch).

State Representation

- We can represent a state in the problem using a notation of the form {l || r} where l and r represent the current contents of the left and right bank respectively.
- Thus:
  - {3c || } is the initial state;
  - { || 3c} is the goal state.
  - {2h || c2w} might be an intermediate state.

Move Representation

- We can represent moves by using a notation of the form {l | b | r} where l and r have the same meanings as before and b is the contents of the boat.
- The following are some valid moves:
  - {c | c | c}
  - {2ch | w | w}
  - {3h | 2w | w}

Invariants

- $0 \leq c \leq 3$
  - There are always between 0 and 3 couples present.
- $c + h = 3$
  - Law of conservation of husbands.
- $c + w = 3$
  - Law of conservation of wives.
- $\therefore\ h = w$

Illegal states and moves

- States in which couples are present with wives…
    - {2cw || h}, {cw || ch}, {c2w || 2h}…

  are all forbidden.
- Moves must obey these rules with the additional rule that the boat can only take the values h, c, w, 2h and 2w.

State Transitions

- We denote a transition between two states, the result of a move, by the notation:
  - {p} m {q}
- Where:
  - {p} is the state before the move;
  - {q} is the state after the move

  and
  - m is the move

State Transitions

- We can combine moves as follows:
  - If $\{p_0\}\ m_1\ \{p_1\}$ and $\{p_1\}\ m_2\ \{p_2\}$
  - then we can write $\{p_0\}$ m1, m2 $\{p_2\}$.
- In general we write:
  - {p} S {q}
  - where S is a sequence of individual moves.

Restating the problem

- The problem now becomes find a sequence of moves, S, so that.
  - {3c || } S { || 3c}.
- We can decompose this to the following symmetric sub-problems.
- Find S_1, S_2 and S_3 such that:
  - {3c || } S_1 {3h || 3w}
  - {3h || 3w} S_2 {3w || 3h}
  - {3w || 3h} S_3 { || 3c}.

Restating the problem

- We now have two smaller problems:
  - Get the wives across (S_1)
  - Swap the husbands and wives (S_2)
- Note that S_3 will simply be the reverse of S_1 so we do not have to find it as well.
- It turns out that finding S_1 (and hence S_3) is easy.

S_1

- Send over two wives:
- {3c||}{c2h|2w|}{c2h||2w}
- Bring one back:
- {c2h||2w}{c2h|w|w}{2ch||w}
- Send two wives over:
- {2ch||w}{3h|2w|w}{3h||3w}

Finding S_2

- If we keep assuming the we are going to have a symmetric solution we need to find two sequences T_1 and T_2 so that
  - {3h || 3w}
  - T_1
  - {c | c | c}
  - T_2
  - {3w || 3h}
- Again, T_2 will simply be the reverse of T_1.

Finding T_1

- The middle move, {c | c | c}, can be interpreted in two ways:
  - the couple is crossing left-to-right;
  - the couple is crossing right-to-left.
- We do not yet know which is the case.
- Either
  - {2c || c} {c | c | c} {c || 2c}

  or
  - {c || 2c} {c | c | c} {2c || c}
- We need to find a set of moves which gets from {3h || 3w} to one of these two states ({c || 2c} or {2c || c}).

T_1

- Send a wife back.
- {3h||3w}{3h|w|2w}{c2h||2w}
- Send two husbands over.
- {c2h||2w}{c|2h|2w}{c||2c}