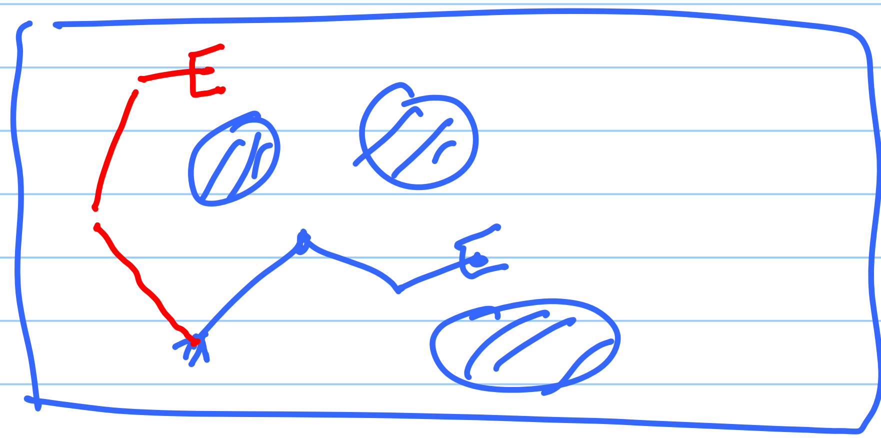
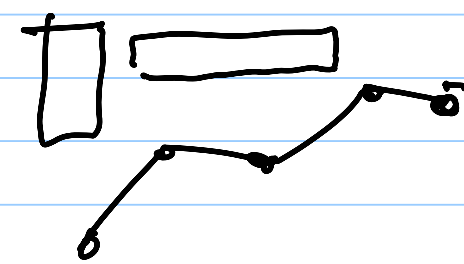
Note Title 1/12/2009

Lecture 2

Q: how difficult or complex is the basic motion planning problem?

KNOWN Env.





BACKGROUND: How do we measure complexity of a problem?

1) Complexity of an alg.

$O(n)$ notation

$O(\ )$

Ex **Sorting Alg:** $A[1:n]$ is given

Prob: arrange A[i] such that they are in descending order.

$a_1, a_2 \dots a_n$

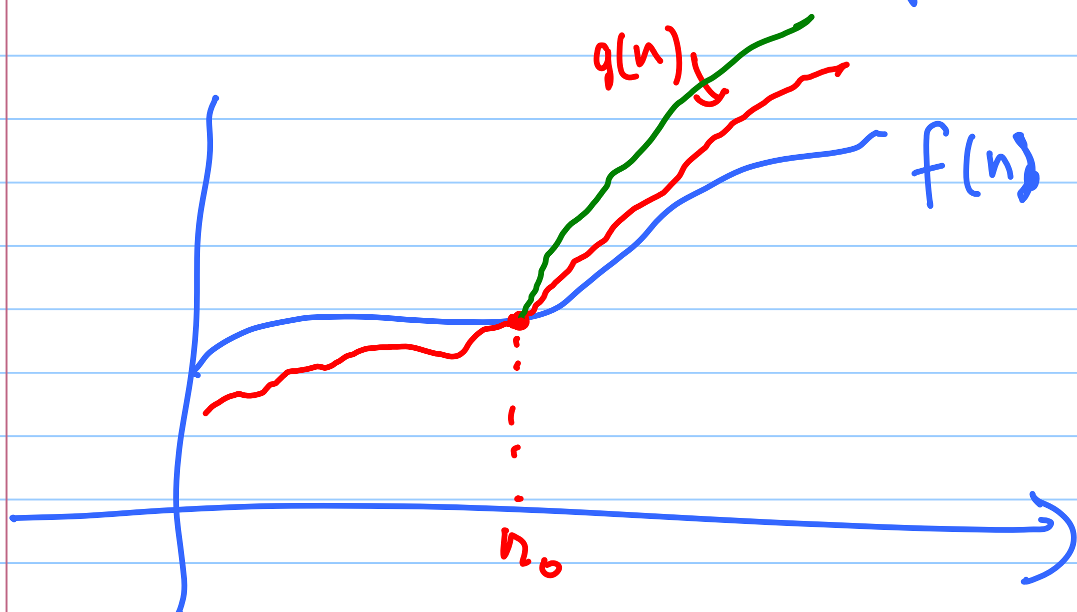
$\underset{i}{\uparrow}\ \underset{j}{\uparrow}$

```
for i ← 1, n do
  for j ← i+1, n do
    if A[i] < A[j]
        temp = A[j]
         A[j] = A[i]
         A[i] = temp
    end if
  end for
end for
```

$$\underset{\text{assign}}{3n^2} + \underset{\text{comparison}}{n^2} = 4n^2$$

$$4n^2 = O(n^2)$$

Def: $f(n) = O(g(n))$ **iff** $\exists\, c, n_0 > 0$ such that $|f(n)| \le c\,|g(n)|$ $\forall\, n > n_0$



alg. → $O(n^m)$ m is a cons. (poly bound)

alg. → $O(2^n)$ exp. bound

problems → best known alg are poly. run time

problems → best known alg. are exp.

best known alg. are exp. → 1 Set Partition

best known alg. are exp. → 2 Boolean Satisfiability

1 Given a set $S = \{e_1, \dots e_n\}$ of numbers.

Can the set be partitioned into two disjoint subsets S_1 & S_2:

$$S_1 \cup S_2 = S$$

$$S_1 \cap S_2 = \phi$$

$$\sum_{e_i \in S_1} e_i = \sum_{e_j \in S_2} e_j$$

2

n literal : $x_1; x_2; \ldots x_n$ $\quad x_n \in \{0, 1\}$

Boolean Sent : $\bigwedge_{i=1}^{k} C_i$

$$C_i = \bigvee x_{ij}$$

<u>Sentence</u> $= (x_1 \vee \bar{x}_2 \vee x_3) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3)$

Prob: Does there exist an ~~too~~ asignment to x_i s.t. Sentence = true "polynomial verifiability"

Non-deterministic machine:

Choice (S) : arbitrarily choose a ~~the~~ correct soln.

Non-det. Polynomial : NP

open Q in CS

det. Polynomial : P

$P \subseteq NP$ $\quad$ $\boxed{NP \subseteq P\ ?}$

Notion of polynomial reducibility

L_1 & L_2 are two problems.

L_1 reduces to L_2 ($L_1 \propto L_2$) <u>iff</u> there is a way to solve L_1 by a deterministic poly. alg. using a det. alg. that solves L_2 in poly time.

$L_1 \propto L_2$

$L_2 \propto L_1$

$\Rightarrow$ L_2 is at least as hard as L_1.

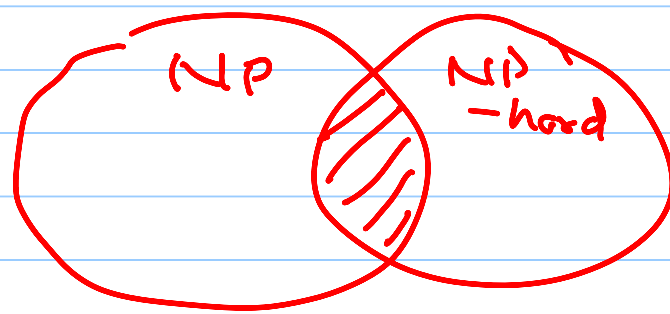
Cook showed: Boolean sat. is the hardest prob. in NP.

Cook's Result

A single Problem (Boolean Satisf.) is the hardest problem in NP.

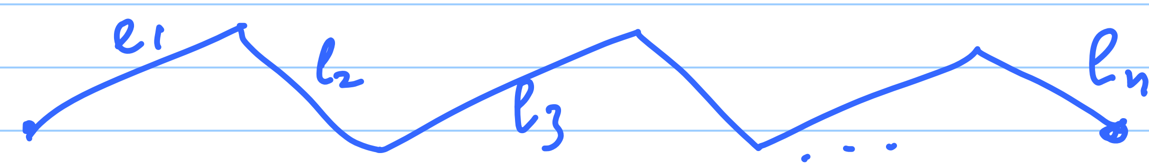
<u>NP-hard:</u> if Sat $\propto$ L, then L is NP-hard

$\wedge$ if $L \in NP$, then L is NP-complete



Ruler - folding problem

Carpenter's ruler:



Given this ruler, and a +ve integer k

Prob: Can you fold the ruler in length $\leq k$?

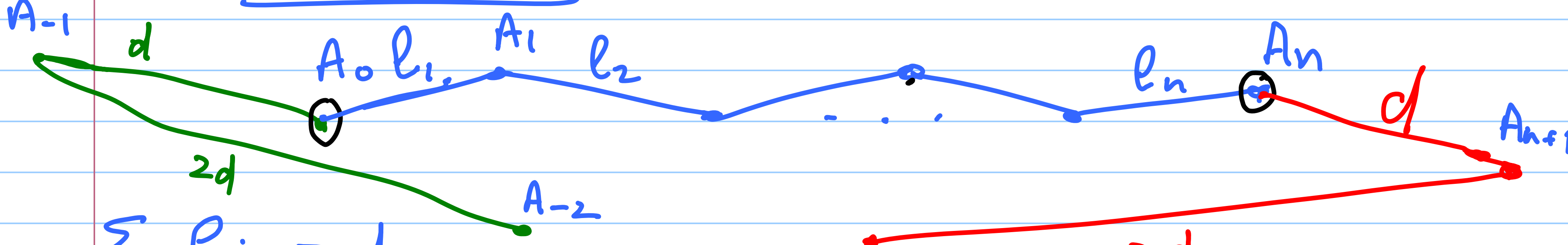
Set Partitioning $\propto$ Ruler folding

Assume we are given an instance of set part. problem.

$$S = \{ l_1, \dots l_n \}$$

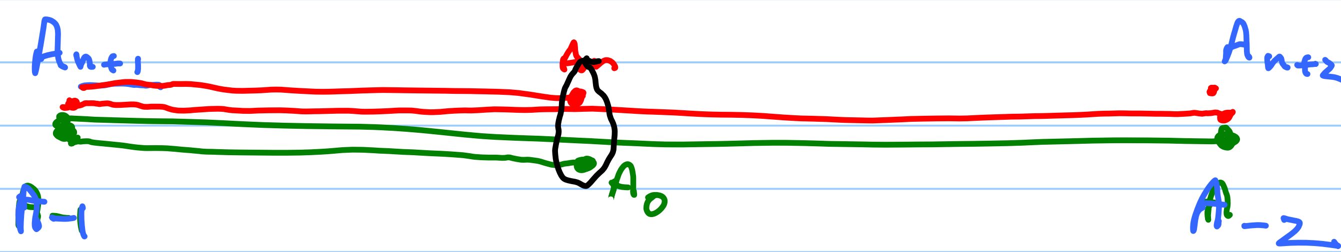
find S_1, S_2 : $S_1 \cup S_2 = S$, $S_1 \cap S_2 = \phi$, $\sum_{l_i \in S_1} l_i = \sum_{l_j \in S_2} l_j$

~~E1~~. <u>Ruler Const:</u>

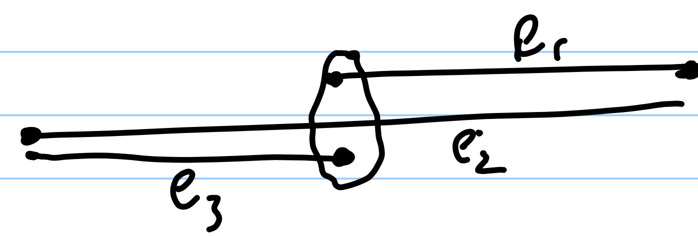


$$\sum_{l_i \in S} l_i = d$$

Can you fold the ruler in length $\leq 2d$.



$\ell_1 + \ell_3 = \ell_2$



ℓ_1 ✓

ℓ_2 $\ell\cdot$

ℓ_3 ✓

any alg. exponential in the number of links

you devise of the ruler

to solve

ruler folding problem is likely

Schwartz & Sharir [1981]

exp. in # dofs,

poly in # of obs.

→ Canny & Reif [1992] → Put proper ref.

??

moving obs.

NP hard ⟷ exp. in # of obstacles

→ 1) geometric complexity : captures types and deg of surface

\# of obs.

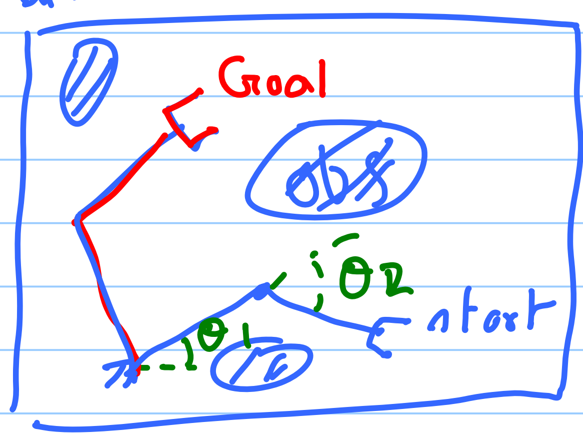
Exp. → 2) Deg. of freedom of robot

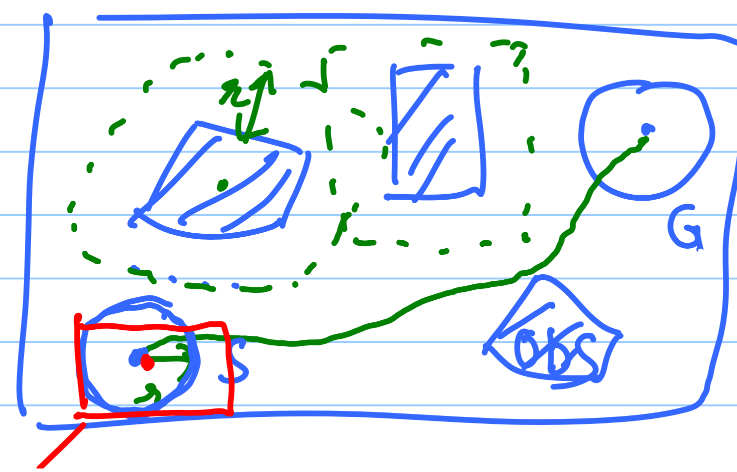
BASIC MP problem:

General Notions: Will give you an idea of the types of things / concepts we will need to solve it.

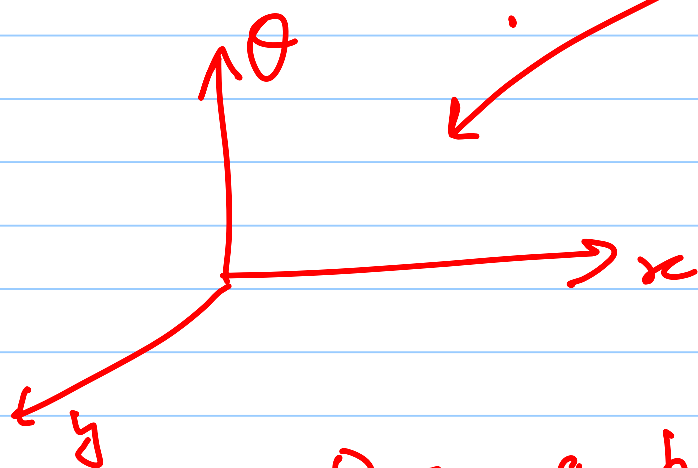
1966 ← of.

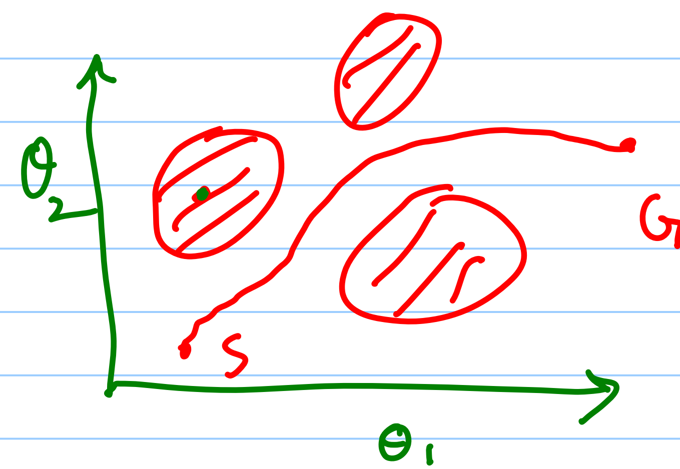
1978 ← Caltech





1978 – Lozano-Perez





Does a path exist? "connected"

"topology" "grow" ⟷ transformation to c-space

or

nature of search the space efficiently

config. space