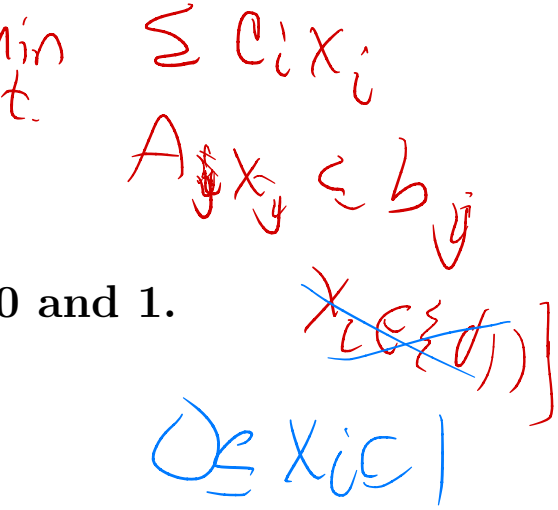
Branch and Bound

- One way of dealing with an NP-complete problem.
- Not polynomial.
- Complete enumeration in the worst case, but uses a clever idea to cut off much of the search space

Branch and Bound for minimizing (0-1) IPs

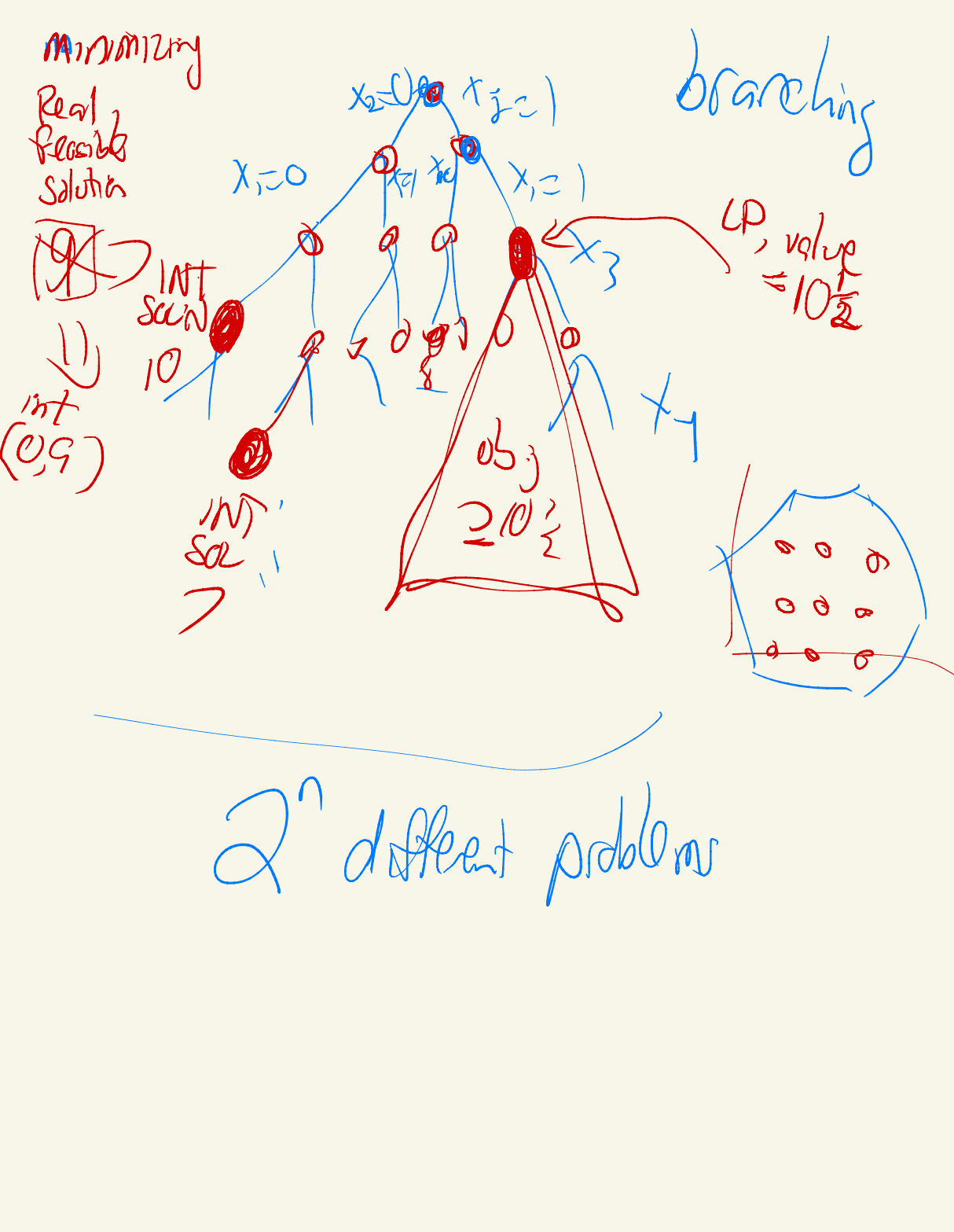
- Solve an LP relaxation
- Choose a variable x_i with value strictly between 0 and 1.
- Branch on whether $x_i = 0$ or $x_i = 1$.
- Bound
  - Generate feasible solutions to the IP
  - At each node generate a lower bound (using the LP) on the solutions in that branch.
  - If the lower bound on a branch is better than a feasible solution, kill that branch.
- When there is no more branching to be done, stop.



Dealing w/ NP-complete problems

1. Exact solution
2. Fast algorithm
3. Works for all inputs

For NP-c. problem, Give up on 1 of these features


Minimizing
Real feasible solution
INT sol'n 10
int (0,9)
branching
$x_2=0$
$x_2=1$
$x_1=0$
$x_1=1$
x_3
x_4
LP, value $=10\frac{1}{2}$
Obj $\geq 10\frac{1}{2}$
INT sol'n
2^n different problems

Branch and Bound

Combinatorial Branch and Bound for a minimization problem

- **Enumerate possible values for a decision.**
- **Branch on all the possiblilities**
- **Bound**
  - **Generate feasible solutions to original problem.**
  - **At each node generate a lower bound (using a relaxation) on the solutions in that branch.**
  - **If the lower bound on a branch is better than a feasible solution, kill that branch.**
- **When there is no more branching to be done, stop.**

Branch and Bound for $1|r_j|L_{\max}$

- **Branch on "which job runs in position i .**
- **Use $1|r_j, pmtn|L_{\max}$ as a lower bound**

- get feasible solns

Example

j	r_j	p_j	d_j
1	**0**	**4**	**8**
2	**1**	**2**	**12**
3	**3**	**6**	**11**
4	**5**	**5**	**10**

preemptive EDD
among the jobs that have been released but not finished, run the one w/ earliest deadline