# Branch and bound

$P_0$: Minimise $f(x)$ s.t. $x \in S_0$

At any point in the algorithm we have a collection $\mathcal{P}$ of sub-problems

$p \in \mathcal{P}$ is defined by the description of a subset $S_p \subseteq S_0$

$p$: minimise $f(x) : x \in S_p$

We have a lower bound $b_p \leq \min\{f(x) : x \in S_p\}$

Procedure BOUND computes $b_p$

Sometimes BOUND gives us a solution $x_p \in S_p$

BOUND might decide that $S_p = \emptyset$

Initialise $\mathcal{P} = \{P_0\}$. We assume that we start with $x^* \in S_0$ and $v^* = f(x^*)$ [If no $x^*$ then $v^* = \infty$]

Often BOUND has the following form.
There is $T_p \supseteq S_p$ and
$b_p = \min\{f(x) : x \in T_p\}$

Step 1: If $\mathcal{P} = \emptyset$, $x^*$ solves $P_0$

Step 2: Choose $P \in \mathcal{P}$, $\mathcal{P} \leftarrow \mathcal{P} \setminus \{P\}$

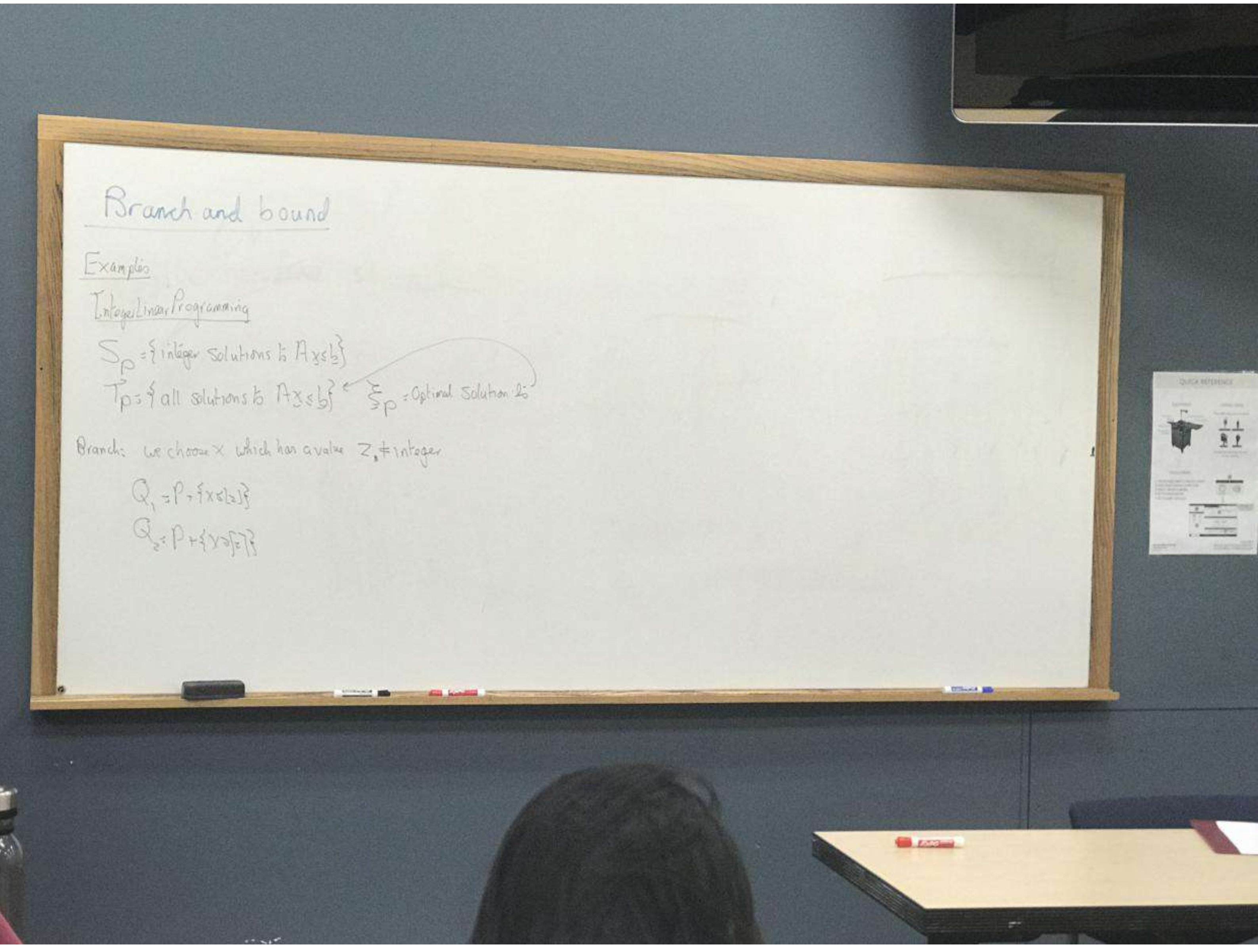
Step 3: Run BOUND(P) to compute $b_p$

Step 4: If $S_p = \emptyset$ or $b_p \geq v^*$ then $P$ is "solved".

Step 5: If BOUND generates $x_p$ and $f(x_p) < v^*$ then $x^* \leftarrow x_p$, $v^* \leftarrow f(x_p)$

Step 6: Split $P$ into sub-problems $Q_1, Q_2, \dots, Q_\ell$ where $S_p = \bigcup_{i=1}^{\ell} Q_i$ & $Q_i \neq S_p$, $\forall i$

Step 7: $\mathcal{P} = \mathcal{P} \cup \{Q_1, Q_2, \dots, Q_\ell\}$

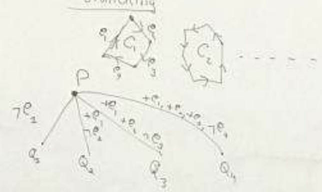

Branch and bound
Examples
Integer Linear Programming
$S_p = \{$integer solutions to $A\underline{x} \le \underline{b}\}$
$T_p = \{$all solutions to $A\underline{x} \le \underline{b}\}$
$\xi_p$ = Optimal solution to
Branch: we choose x which has a value $z_1 \neq$ integer
$Q_1 = P + \{x \le \lfloor z \rfloor\}$
$Q_2 = P + \{x \ge \lceil z \rceil\}$

# Branch and bound

Examples

## Traveling Salesperson TSP

Branching

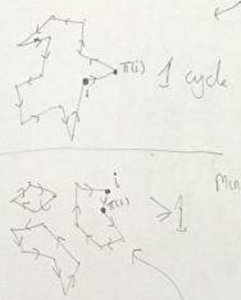



$S_p = \{tours\}$ - tour is a single cycle that covers all the vertices

$T_p = \{$Collections of vertex disjoint cycles that covers all vertices$\}$

$T_p \supseteq S_p$ minimising $f(\underline{x})$, $\underline{x} \in T_p$ → Assignment Problem

minimise $\sum c_{i,\pi(i)}$, $\pi \in \{$cyclic permutations$\}$




Assignment

Minimise $\sum c_{i,\pi(i)}$, $\pi \in \{$all permutations$\}$

A permutation $i \mapsto \pi(i)$ defines

# Branch and bound

Examples

## Implicit Enumeration

$P$ Maximise $\sum_{j=1}^{n} c_j x_j$ st $\sum_{j=1}^{n} a_{ij} x_j \geq b_i$, $i \in [m]$, $x_j \in \{0,1\}$, $j \in [n]$

A sub-problem is associated with 2 sets $I, O \subseteq [n]$

$P_{I,O} = P + \begin{matrix} x_j = 1, & j \in I \\ x_j = 0, & j \in O \end{matrix}$

$b_{I,O} = \sum_{j \notin O} \max\{0, c_j\}$ ← upper bound on solution value

Feasibility Check: $\sum_{j \notin O} \max\{0, a_{ij}\} \geq b_i$