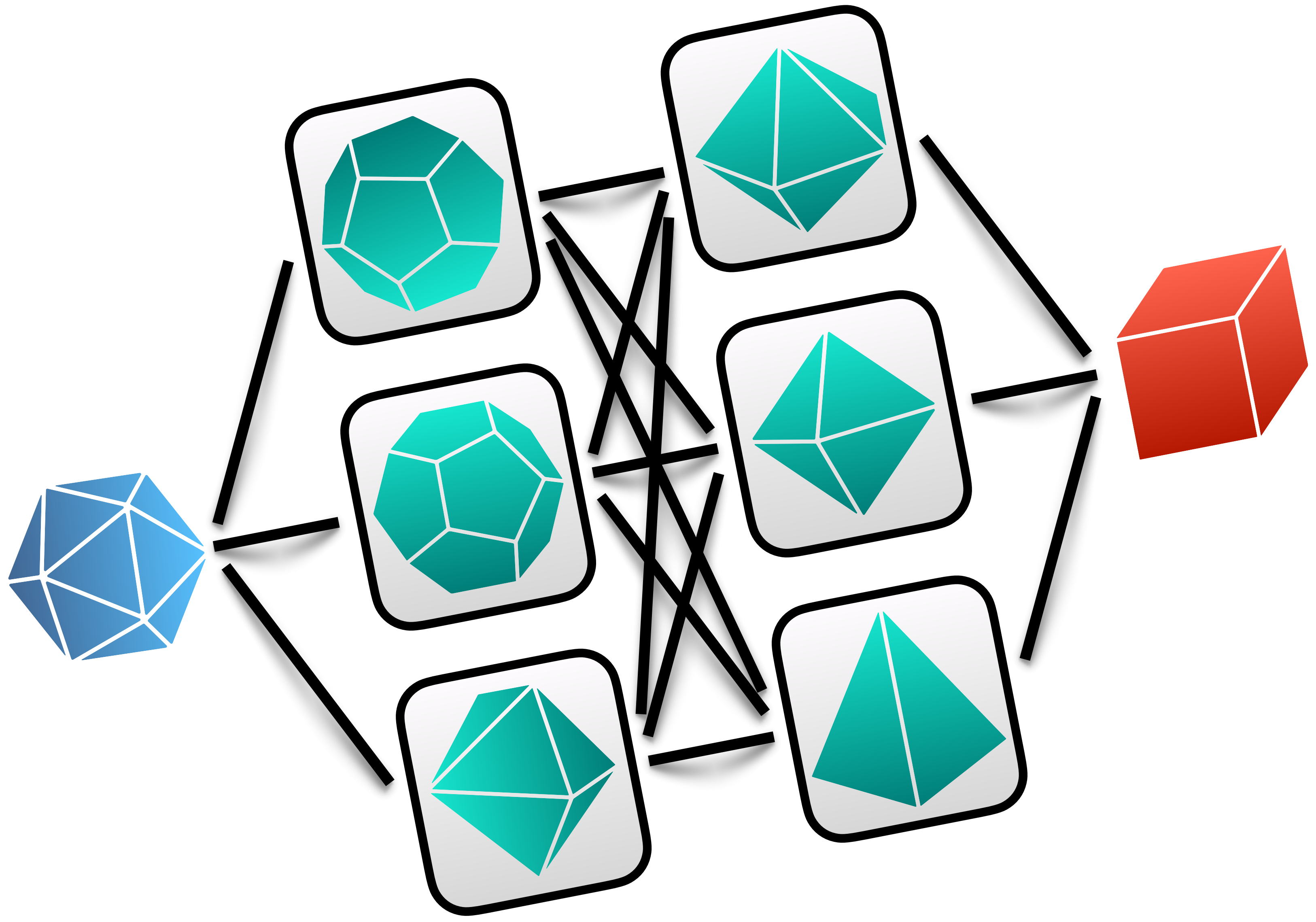
Did I forget to hit record? Please remind me!



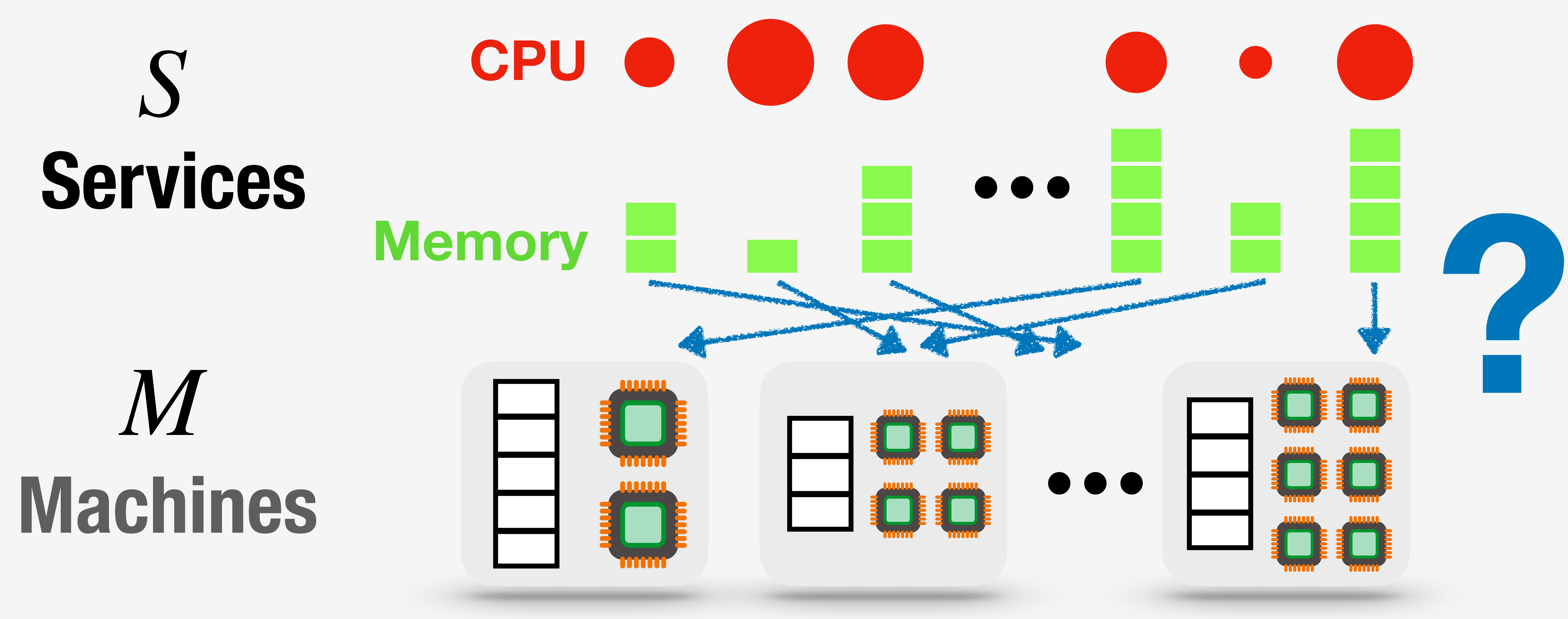
Integer Programming

MIE1666: Machine Learning for Mathematical Optimization

Readings and figures from Chapters 1, 2, 7, 13 of Integer Programming by Wolsey

Elias B. Khalil — 20/09/21

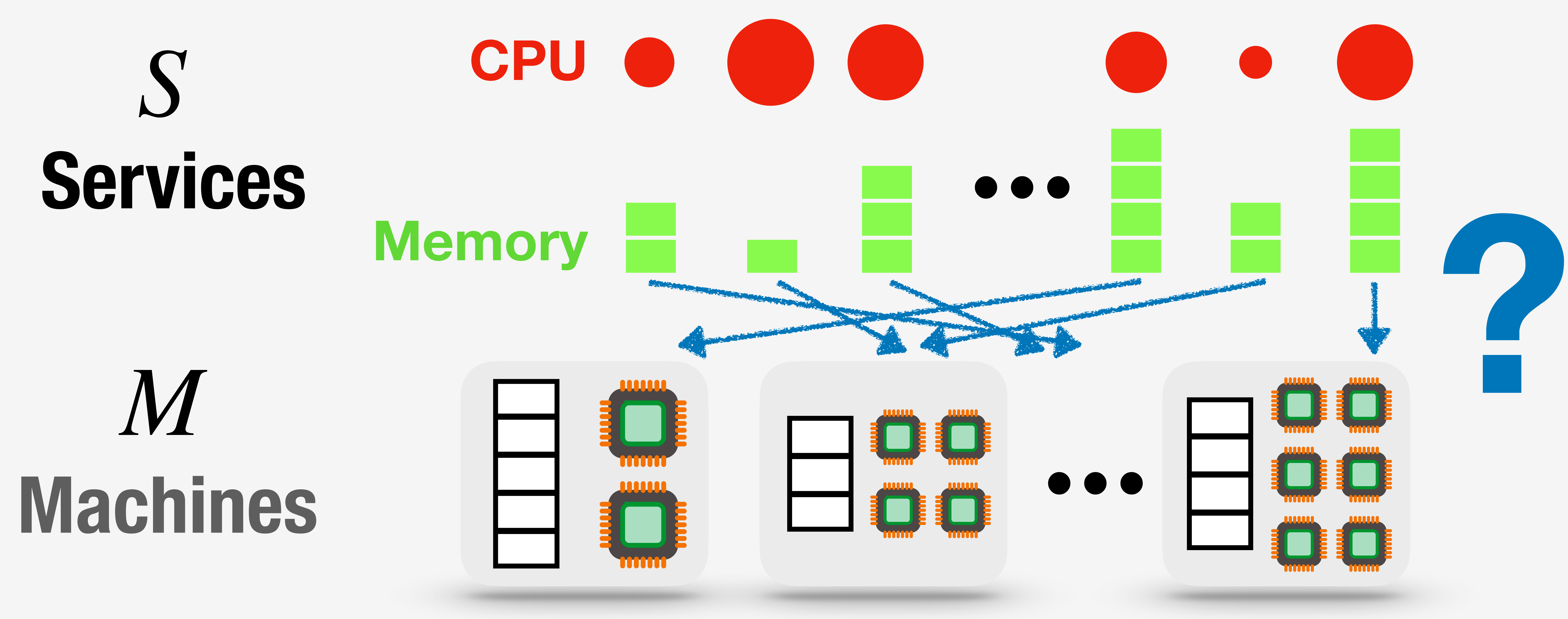
UNIVERSITY OF TORONTO



S
Services
CPU
Memory
M
Machines
?

$y_m = 1$ if machine m is used

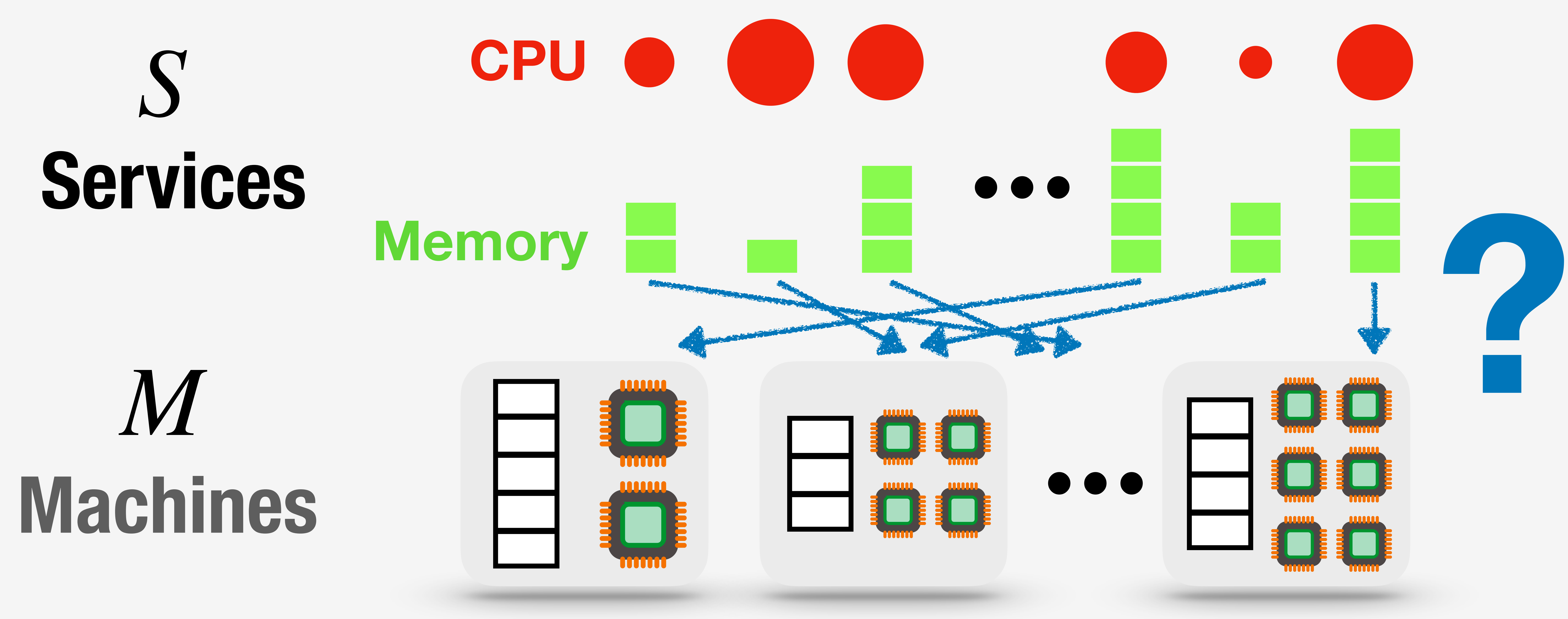
$x_{s,m} = 1$ if service s runs on m



$y_m = 1$ if machine m is used

$x_{s,m} = 1$ if service s runs on m

$x \in \{0,1\}^{S \times M}$, $y \in \{0,1\}^M$

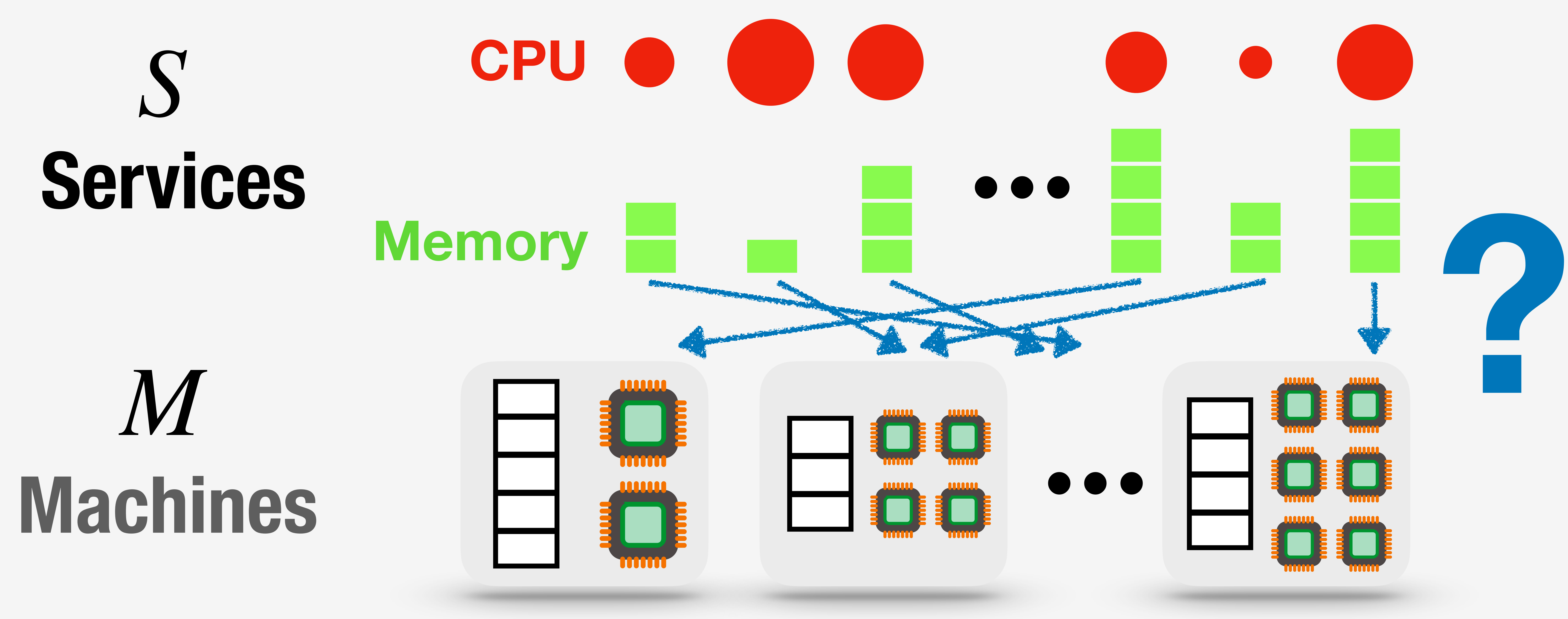


$y_m = 1$ if machine m is used

$x_{s,m} = 1$ if service s runs on m

$$x \in \{0,1\}^{S \times M},\ y \in \{0,1\}^M$$

minimize $\sum_{m=1}^{M} y_m$



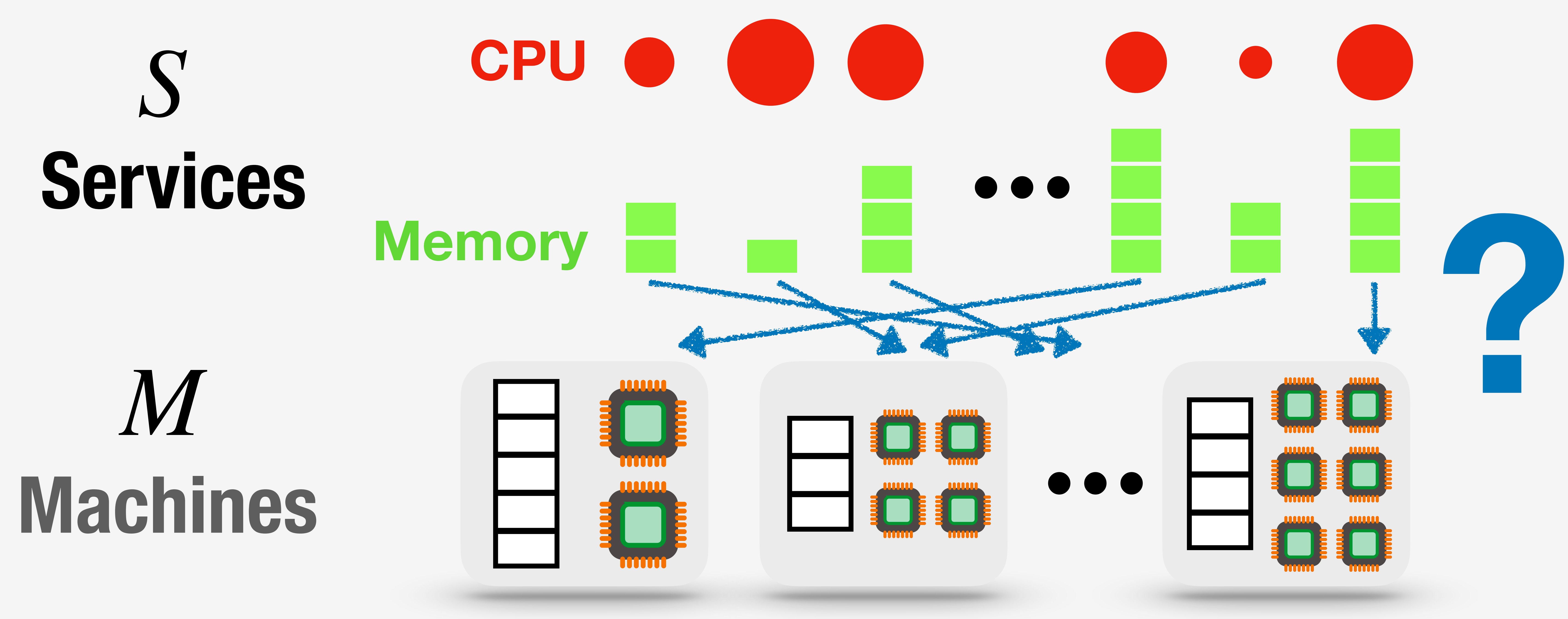
$y_m = 1$ if machine m is used

$x_{s,m} = 1$ if service s runs on m

$$x \in \{0,1\}^{S \times M},\ y \in \{0,1\}^M$$

minimize $\sum_{m=1}^{M} y_m$

Constraints:

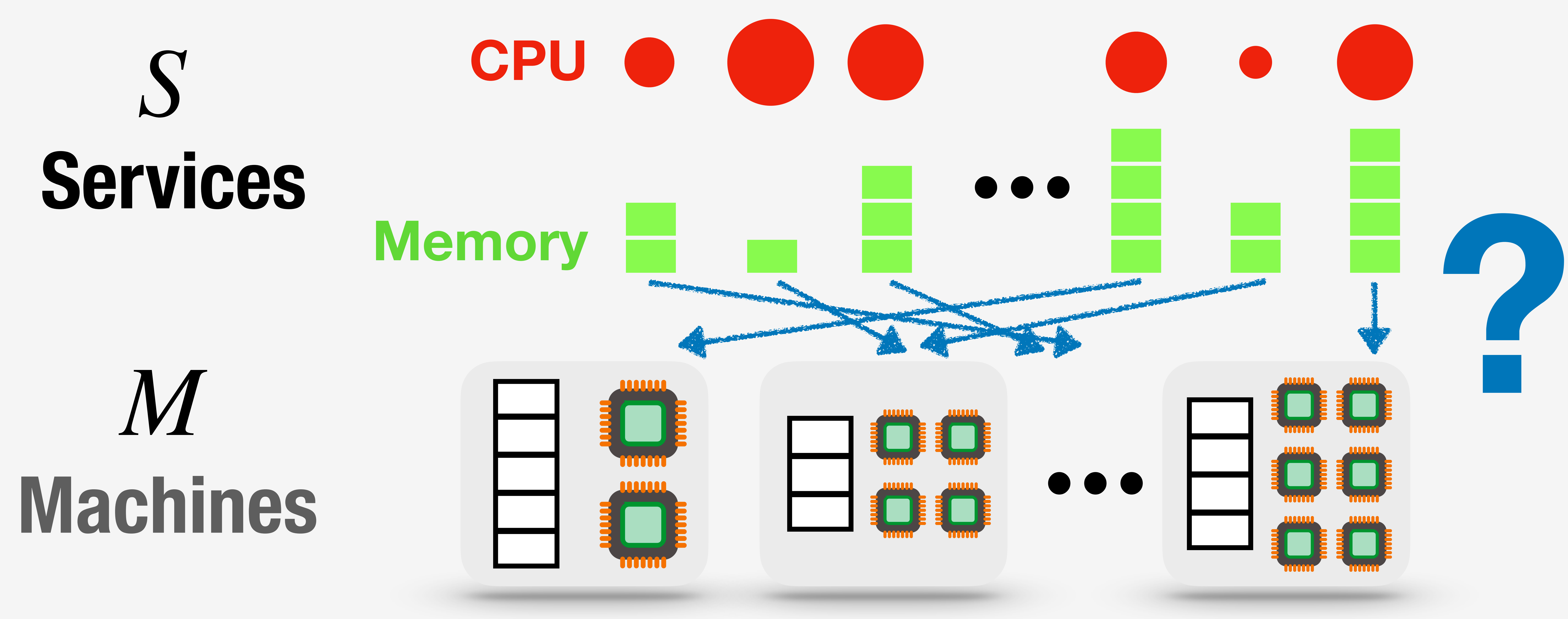


$y_m = 1$ if machine m is used

$x_{s,m} = 1$ if service s runs on m

$x \in \{0,1\}^{S \times M}$, $y \in \{0,1\}^M$

minimize $\sum_{m=1}^{M} y_m$



Constraints:

Each service on one machine only

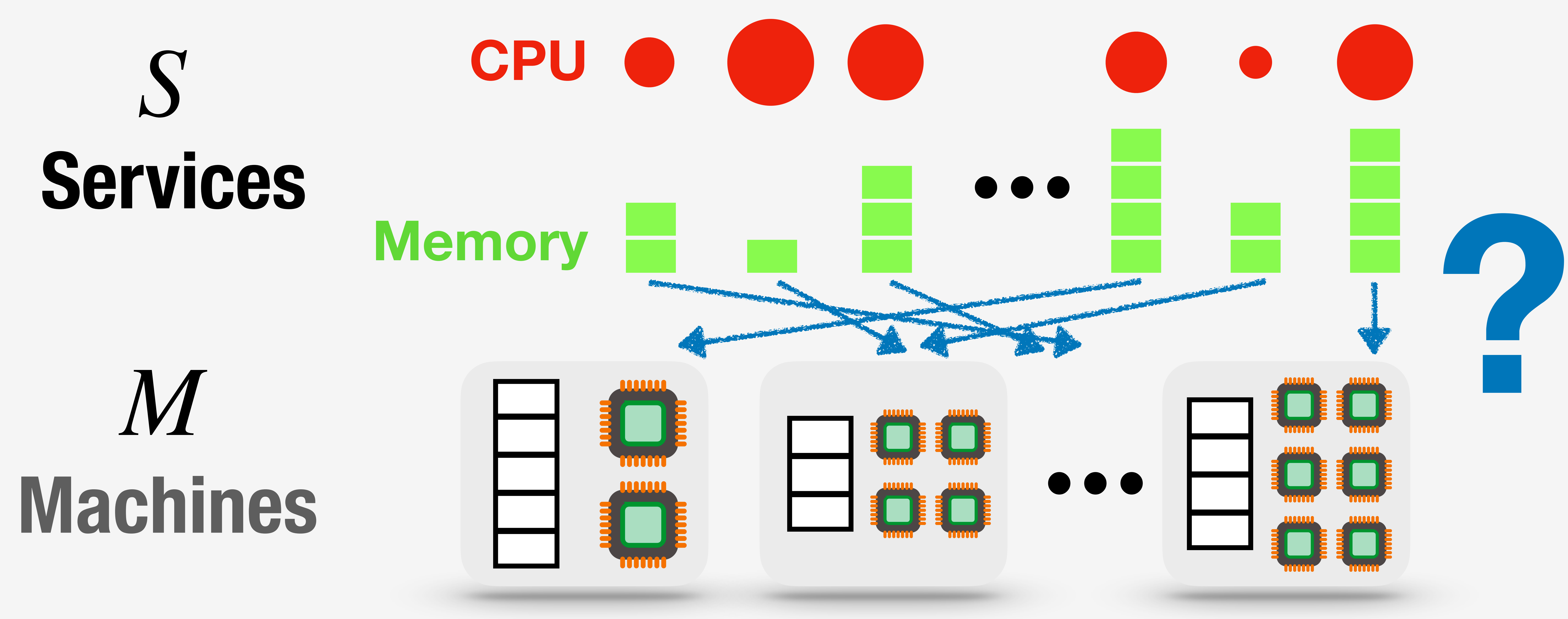
$$\sum_{m=1}^{M} x_{s,m} = 1 \quad \forall s$$

$y_m = 1$ if machine m is used

$x_{s,m} = 1$ if service s runs on m

$$x \in \{0,1\}^{S \times M},\ y \in \{0,1\}^M$$

$$\textbf{minimize} \sum_{m=1}^{M} y_m$$



Constraints:

Each service on one machine only

$$\sum_{m=1}^{M} x_{s,m} = 1 \quad \forall s$$

$$y_m \geq x_{s,m} \quad \forall s, m$$

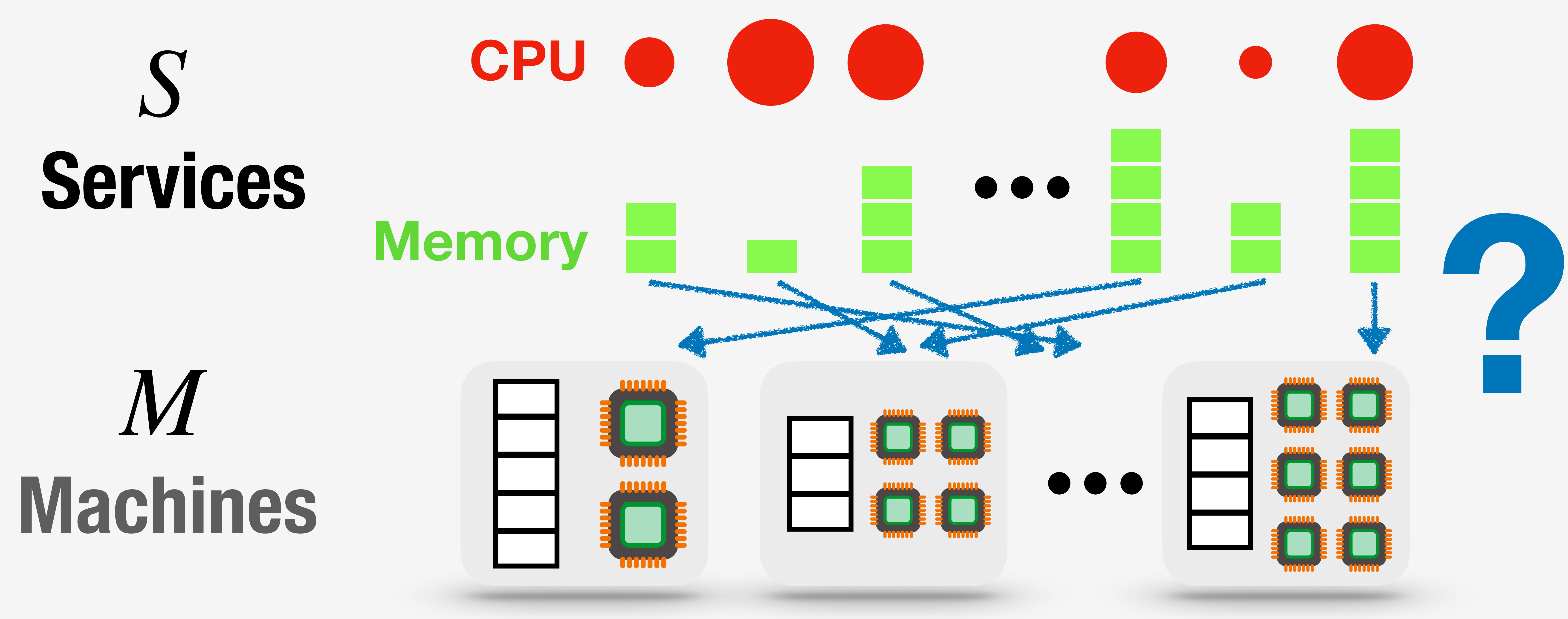
Machine is "ON" if a job is assigned to it

$y_m = 1$ if machine m is used

$x_{s,m} = 1$ if service s runs on m

$x \in \{0,1\}^{S \times M}, \; y \in \{0,1\}^M$

minimize $\sum_{m=1}^{M} y_m$



Constraints:

Each service on one machine only

$$\sum_{m=1}^{M} x_{s,m} = 1 \quad \forall s$$

Memory capacity

$$\sum_{s=1}^{S} \textbf{mem}(s) \cdot x_{s,m} \leq \textbf{cap-mem}(m) \quad \forall m$$

$$y_m \geq x_{s,m} \quad \forall s, m$$

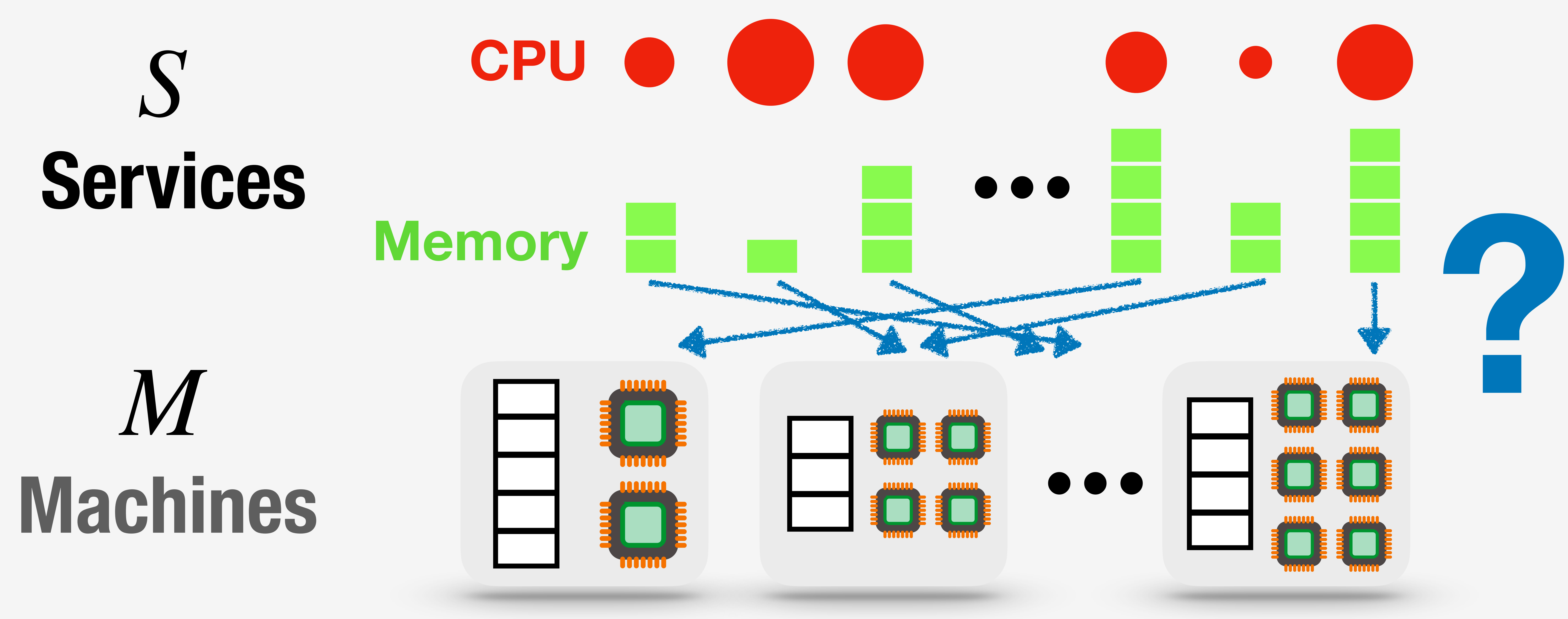
Machine is "ON" if a job is assigned to it

$y_m = 1$ if machine m is used

$x_{s,m} = 1$ if service s runs on m

$$x \in \{0,1\}^{S \times M},\ y \in \{0,1\}^M$$

$$\textbf{minimize} \sum_{m=1}^{M} y_m$$



Constraints:

Each service on one machine only

$$\sum_{m=1}^{M} x_{s,m} = 1 \quad \forall s$$

Memory capacity

$$\sum_{s=1}^{S} \textbf{mem}(s) \cdot x_{s,m} \leq \textbf{cap-mem}(m) \quad \forall m$$

$$y_m \geq x_{s,m} \quad \forall s, m$$

Machine is "ON" if a job is assigned to it

$$\sum_{s=1}^{S} \textbf{cpu}(s) \cdot x_{s,m} \leq \textbf{cap-cpu}(m) \quad \forall m$$

Processor capacity

Linear Program

$$\max\ c^\intercal x$$

$$Ax \leq b$$

$$x \geq 0$$

Data (known) $A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^{m \times 1}, c \in \mathbb{R}^{n \times 1}$

Variables (unknown) $x \in \mathbb{R}^{n \times 1}$

(Pure) Integer (Linear) Program

$$\max c^\mathsf{T} x$$

$$Ax \leq b$$

$$x \geq 0 \text{ and integer}$$

$$A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^{m \times 1}, c \in \mathbb{R}^{n \times 1}$$

$$x \in \mathbb{R}^{n \times 1}$$

Mixed Integer (Linear) Program

$$\max c^{\intercal} x + h^{\intercal} y$$
$$Ax + Gy \leq b$$
$$x \geq 0 \text{ and integer, } y \geq 0$$

$$A \in \mathbb{R}^{m \times n}, G \in \mathbb{R}^{m \times p}, b \in \mathbb{R}^{m \times 1}, c \in \mathbb{R}^{n \times 1}, h \in \mathbb{R}^{p \times 1}$$

$$x \in \mathbb{R}^{n \times 1}, y \in \mathbb{R}^{p \times 1}$$

0-1 (Binary) Program

$$\begin{aligned} \max\ & c^\intercal x \\ & Ax \leq b \\ & x \in \{0,1\}^n \end{aligned}$$

$$A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^{m \times 1}, c \in \mathbb{R}^{n \times 1}$$

$$x \in \mathbb{R}^{n \times 1}$$

Combinatorial Optimization Problem

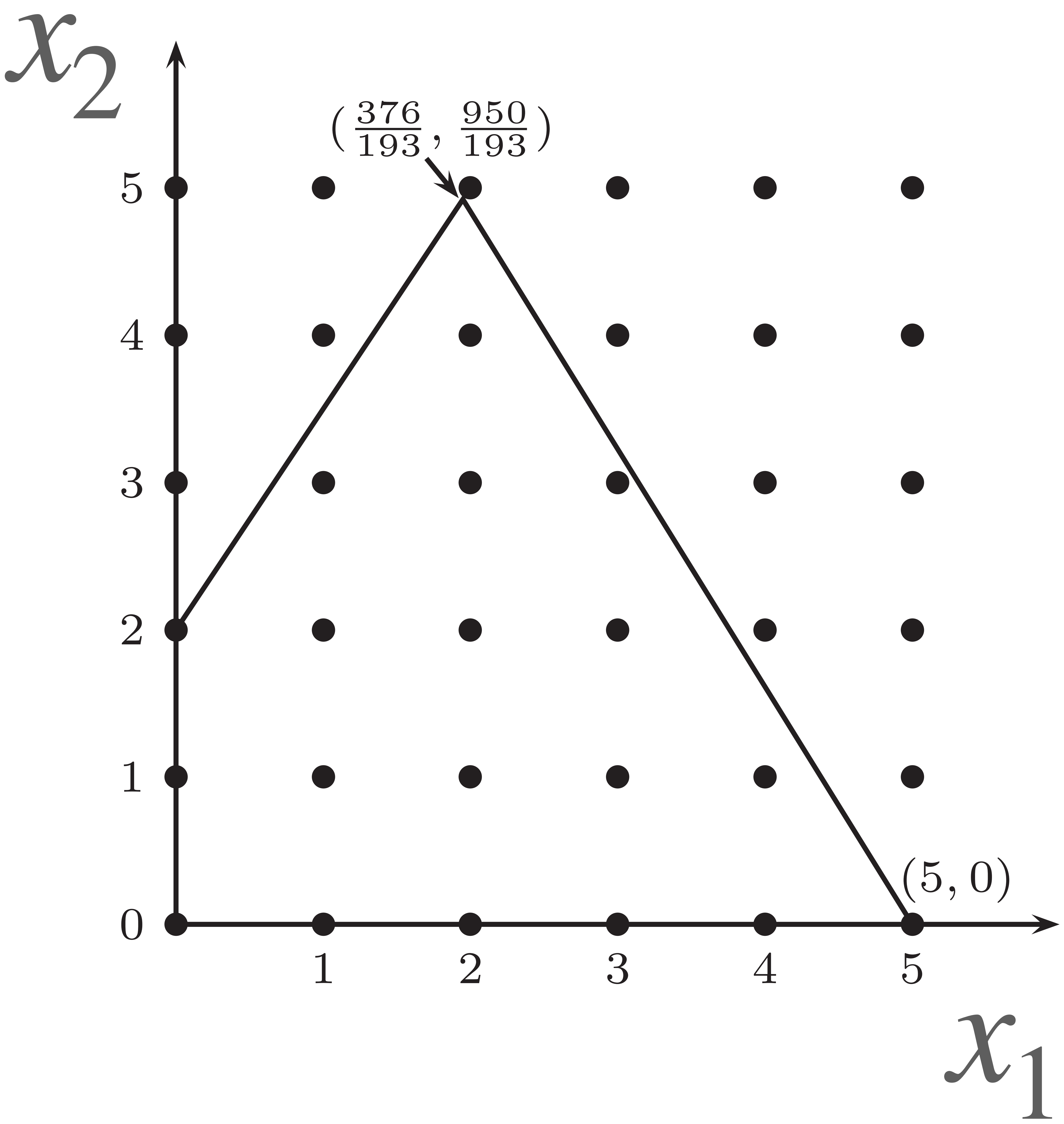
$$\min_{S \subseteq N} \left\{ \sum_{j \in S} c_j : S \in \mathcal{F} \right\}$$

$N = \{1, \ldots, n\}$ is a finite set,

$c_j \in \mathbb{R}$ is a *weight* for each $j \in N$,

$\mathcal{F}$ is a set of feasible subsets of N.

$$
\begin{array}{llllc}
\max & 1.00x_1 & +0.64x_2 & & \\
& 50x_1 & +31x_2 & \leq & 250 \\
& 3x_1 & -2x_2 & \geq & -4 \\
& x_1, & x_2 & \geq & 0 \text{ and integer.}
\end{array}
$$



From Integer Programming by Wolsey

Formulating optimization problems

1. Decision variables: what do you control?
2. Constraints: what conditions do these variables need to satisfy?
3. Objective function: what are you maximizing in the variables?
4. Check: do 1-3 accurately represent your problem? If not, repeat.

Assignment Problem

There are n people available to carry out n jobs. Each person is assigned to carry out exactly one job. Some individuals are better suited to particular jobs than others, so there is an estimated cost c_{ij} if person i is assigned to job j. The problem is to find a minimum cost assignment.

Definition of the variables.

$x_{ij} = 1$ if person i does job j, and $x_{ij} = 0$ otherwise.

Definition of the constraints.

Each person i does one job:

$$\sum_{j=1}^{n} x_{ij} = 1 \quad \text{for } i = 1, \ldots, n.$$

Each job j is done by one person:

$$\sum_{i=1}^{n} x_{ij} = 1 \quad \text{for } j = 1, \ldots, n.$$

The variables are 0–1:

$$x_{ij} \in \{0, 1\} \quad \text{for } i = 1, \ldots, n, j = 1, \ldots, n.$$

Definition of the objective function.

The cost of the assignment is minimized:

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij}.$$

0-1 Knapsack Problem

There is a budget b available for investment in projects during the coming year and n projects are under consideration, where a_j is the outlay for project j and c_j is its expected return. The goal is to choose a set of projects so that the budget is not exceeded and the expected return is maximized.

Definition of the variables.

$x_j = 1$ if project j is selected, and $x_j = 0$ otherwise.

Definition of the constraints.

The budget cannot be exceeded:

$$\sum_{j=1}^{n} a_j x_j \leq b.$$

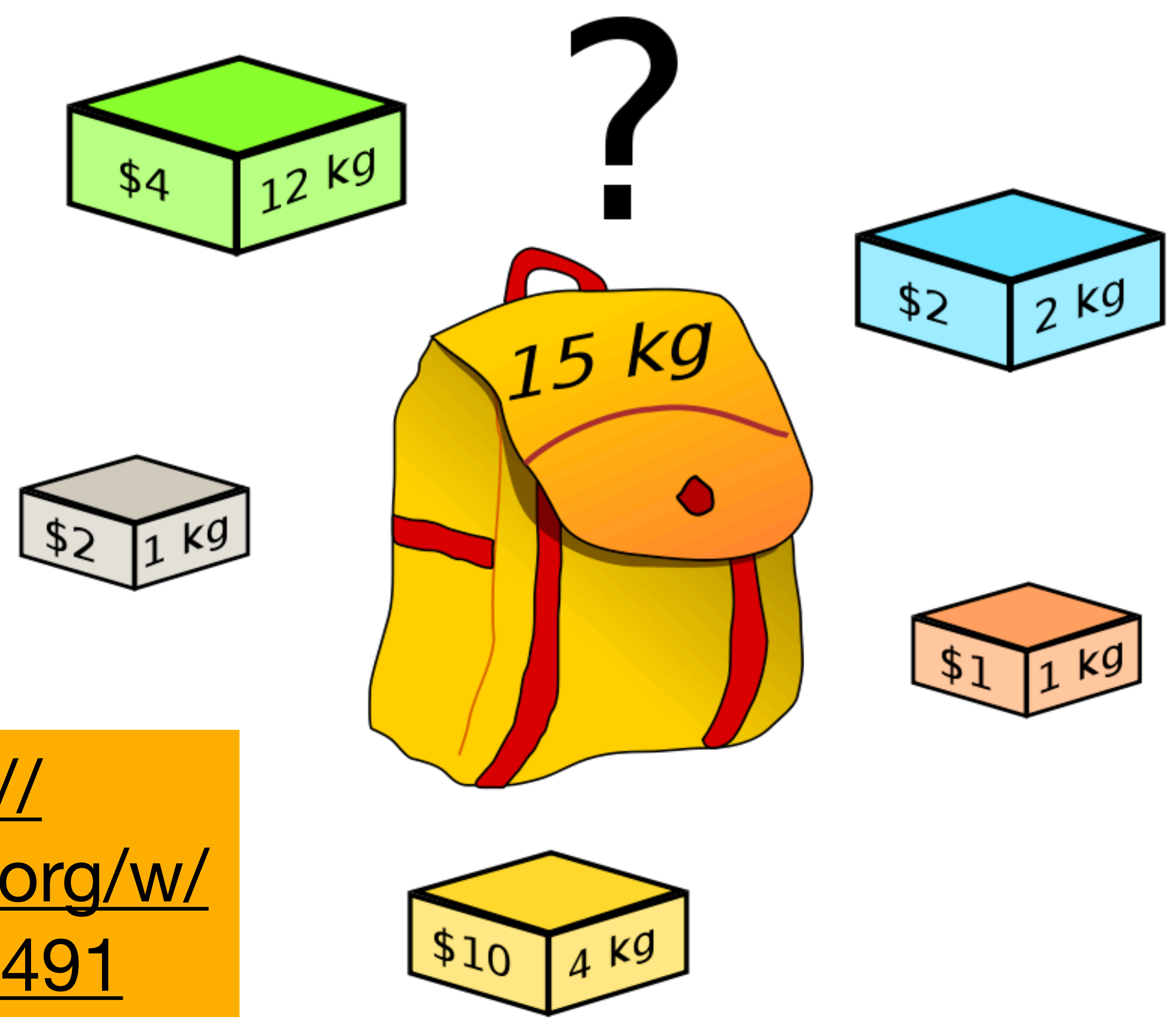
The variables are 0–1:

$$x_j \in \{0, 1\} \quad \text{for } j = 1, \ldots, n.$$

Definition of the objective function.

The expected return is maximized:

$$\max \sum_{j=1}^{n} c_j x_j.$$



CC BY-SA 2.5, https://commons.wikimedia.org/w/index.php?curid=985491

Set Covering Problem

Given a certain number of regions, the problem is to decide where to install a set of emergency service centers. For each possible center, the cost of installing a service center and which regions it can service are known. For instance, if the centers are fire stations, a station can service those regions for which a fire engine is guaranteed to arrive on the scene of a fire within eight minutes. The goal is to choose a minimum cost set of service centers so that each region is covered.

$$\min_{T \subseteq N} \left\{ \sum_{j \in T} c_j : \cup_{j \in T} S_j = M \right\}$$

$M = \{1, \ldots, M\}$ is the set of regions,

$N = \{1, \ldots, n\}$ is the set of potential centers,

$c_j \in \mathbb{R}^+$ is the per-region installation cost.

Set Covering Problem

Given a certain number of regions, the problem is to decide where to install a set of emergency service centers. For each possible center, the cost of installing a service center and which regions it can service are known. For instance, if the centers are fire stations, a station can service those regions for which a fire engine is guaranteed to arrive on the scene of a fire within eight minutes. The goal is to choose a minimum cost set of service centers so that each region is covered.

Definition of the variables.

$x_j = 1$ if center j is selected, and $x_j = 0$ otherwise.

Definition of the constraints.

At least one center must service region i:

$$\sum_{j=1}^{n} a_{ij} x_j \geq 1 \quad \text{for } i = 1, \ldots, m.$$

The variables are 0–1:

$$x_j \in \{0, 1\} \quad \text{for } j = 1, \ldots, n.$$

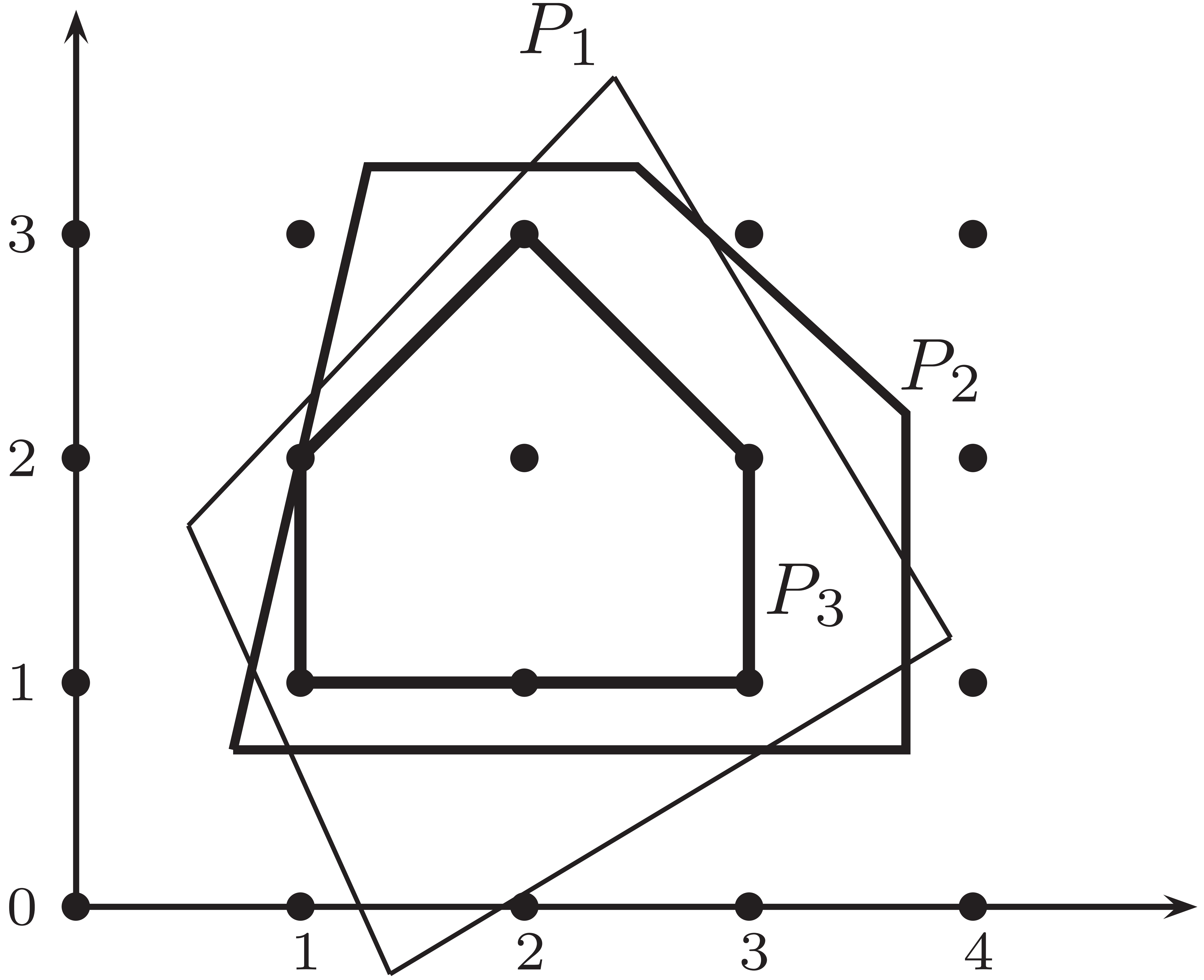
Definition of the objective function.

The total cost is minimized:

$$\min \sum_{j=1}^{n} c_j x_j.$$

Equivalent formulations of the same set

$$X = \{(1,1), (2,1), (3,1), (1,2), (2,2), (3,2), (2,3)\}$$



0-1 (Binary) Program

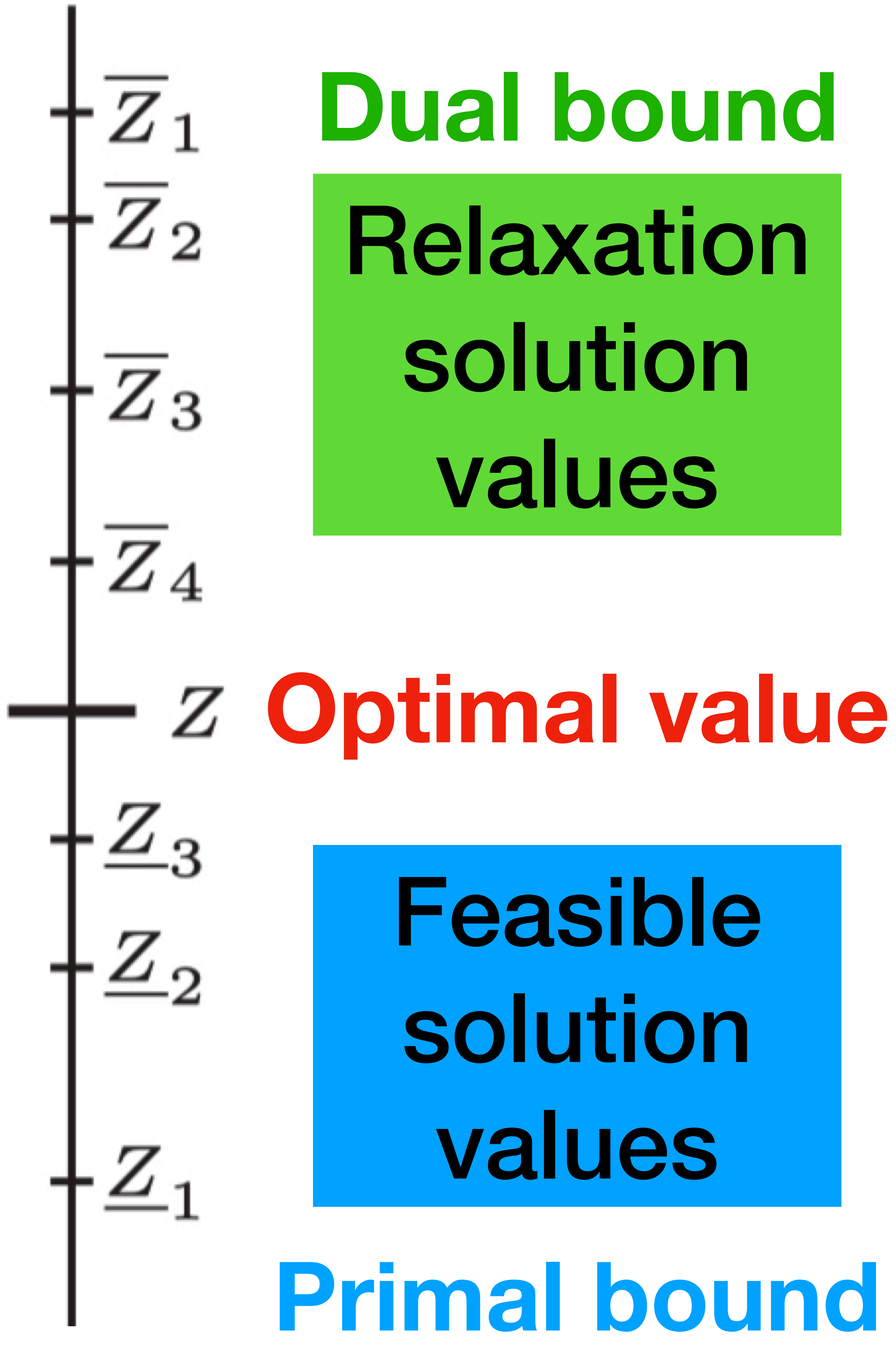
$$Z = \max \; c^{\intercal} x$$

$$Ax \leq b$$

$$x \in \{0, 1\}^n$$

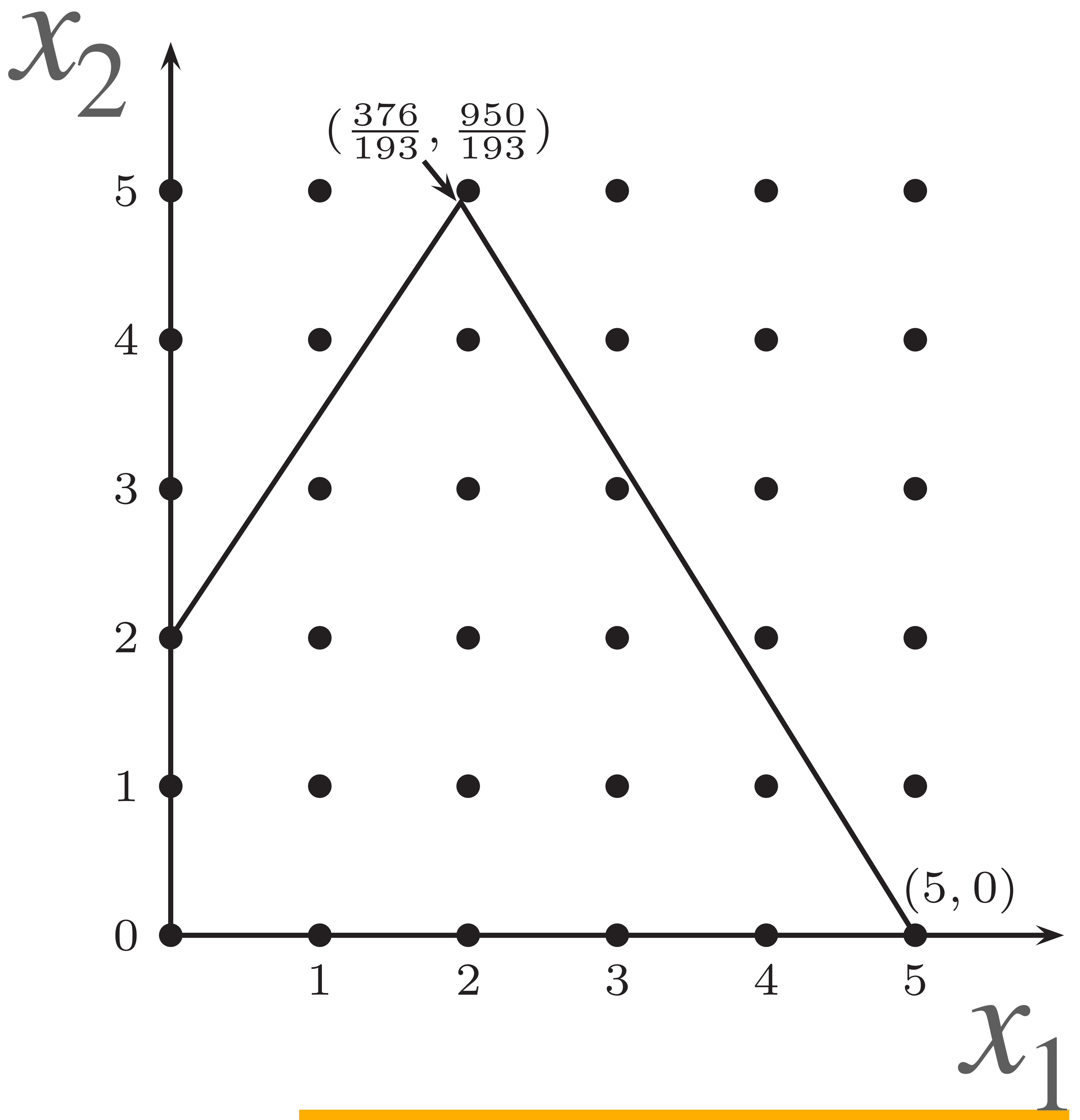
$$A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^{m \times 1}, c \in \mathbb{R}^{n \times 1}$$

$$x \in \mathbb{R}^{n \times 1}$$



Dual bound from Relaxations

$$\begin{array}{lrrcl}
\max & 1.00x_1 & +0.64x_2 & & \\
 & 50x_1 & +31x_2 & \le & 250 \\
 & 3x_1 & -2x_2 & \ge & -4 \\
 & x_1, & x_2 & \ge & 0 \text{ and integer.}
\end{array}$$



From Integer Programming by Wolsey

Set Covering Problem

Given a certain number of regions, the problem is to decide where to install a set of emergency service centers. For each possible center, the cost of installing a service center and which regions it can service are known. For instance, if the centers are fire stations, a station can service those regions for which a fire engine is guaranteed to arrive on the scene of a fire within eight minutes. The goal is to choose a minimum cost set of service centers so that each region is covered.

$$\min_{T \subseteq N} \left\{ \sum_{j \in T} c_j : \cup_{j \in T} S_j = M \right\}$$

Can you find a feasible solution easily?

$M = \{1, \ldots, M\}$ is the set of regions,

$N = \{1, \ldots, n\}$ is the set of potential centers,

$c_j \in \mathbb{R}^+$ is the per-region installation cost.

0-1 Knapsack Problem

There is a budget b available for investment in projects during the coming year and n projects are under consideration, where a_j is the outlay for project j and c_j is its expected return. The goal is to choose a set of projects so that the budget is not exceeded and the expected return is maximized.

Definition of the variables.

$x_j = 1$ if project j is selected, and $x_j = 0$ otherwise.

Definition of the constraints.

The budget cannot be exceeded:

$$\sum_{j=1}^{n} a_j x_j \le b.$$

The variables are 0–1:

$$x_j \in \{0, 1\} \quad \text{for } j = 1, \dots, n.$$

Definition of the objective function.

The expected return is maximized:

$$\max \sum_{j=1}^{n} c_j x_j.$$

Can you find a feasible solution greedily?

0-1 Knapsack Problem

Greedy algorithm

$$\min \left\{ \sum_{j=1}^{n} c_j x_j \; : \; \sum_{j=1}^{n} a_j x_j \geq b, \; x \in \{0,1\}^n \right\}$$

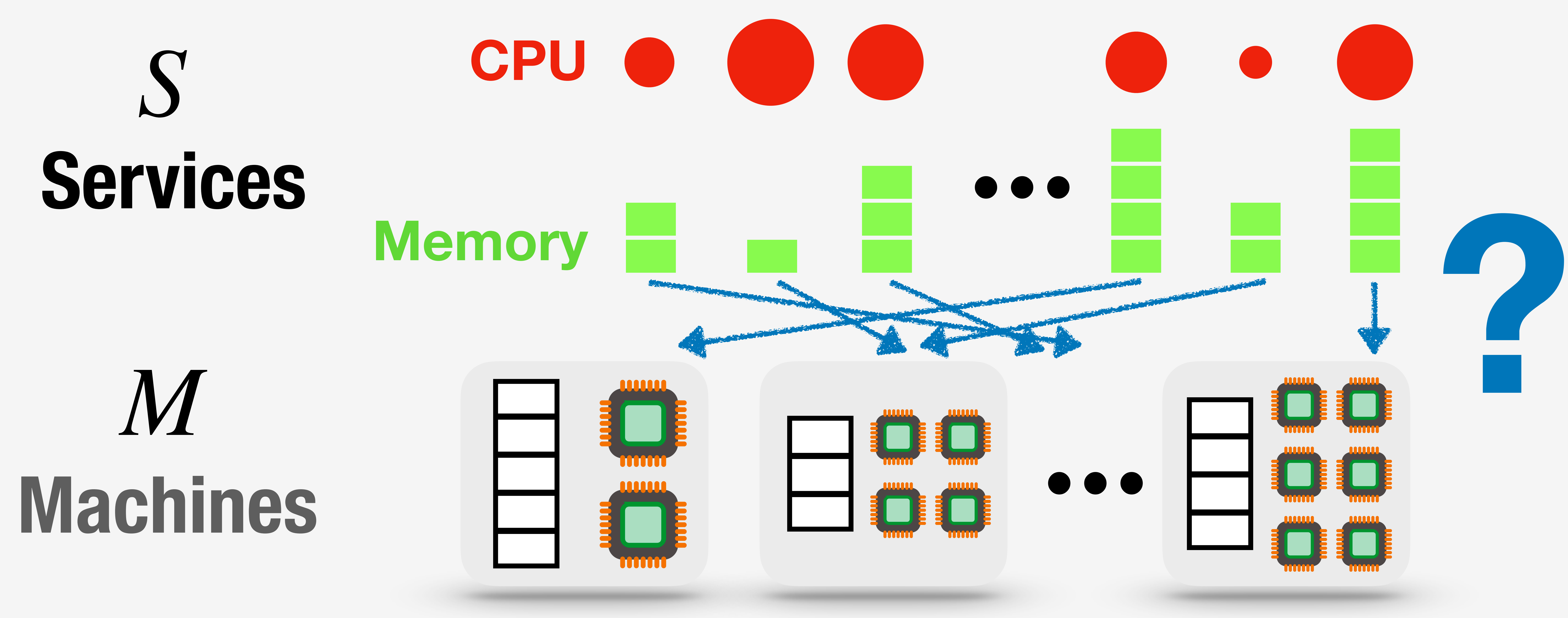
1. Set $S^0 = \emptyset$ (start with the empty set). Set $t = 1$.
2. Set $j_t = \arg\min \frac{c(S^{t-1} \cup \{j_t\}) - c(S^{t-1})}{v(S^{t-1} \cup \{j_t\}) - v(S^{t-1})}$ (choose the element whose additional cost per unit of resource is minimum).
3. If the previous solution S^{t-1} is feasible, i.e. $v(S^{t-1}) \geq k$, and $c\left(S^{t-1} \cup \{j_t\}\right) \geq c\left(S^{t-1}\right)$, stop with $S^G = S^{t-1}$.
4. Otherwise set $S^t = S^{t-1} \cup \{j_t\}$.
5. If $t = n$, stop with $S^G = N$.
6. Otherwise set $t \leftarrow t + 1$, and return to 2.

$y_m = 1$ if machine m is used

$x_{s,m} = 1$ if service s runs on m

$x \in \{0,1\}^{S \times M}, \; y \in \{0,1\}^M$

minimize $\sum_{m=1}^{M} y_m$



Constraints:

Each service on one machine only

$$\sum_{m=1}^{M} x_{s,m} = 1 \quad \forall s$$

$$y_m \geq x_{s,m} \quad \forall s, m$$

Machine is "ON" if a job is assigned to it

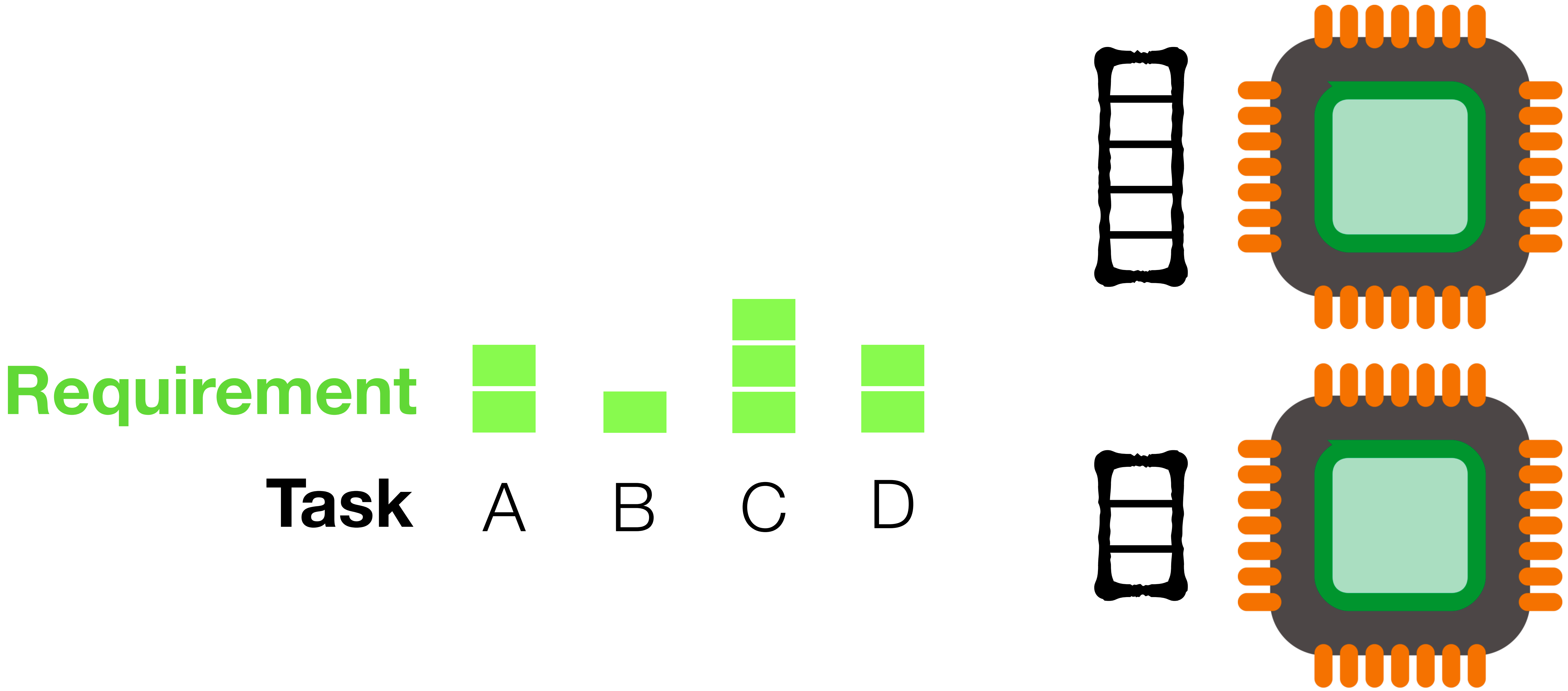
Memory capacity

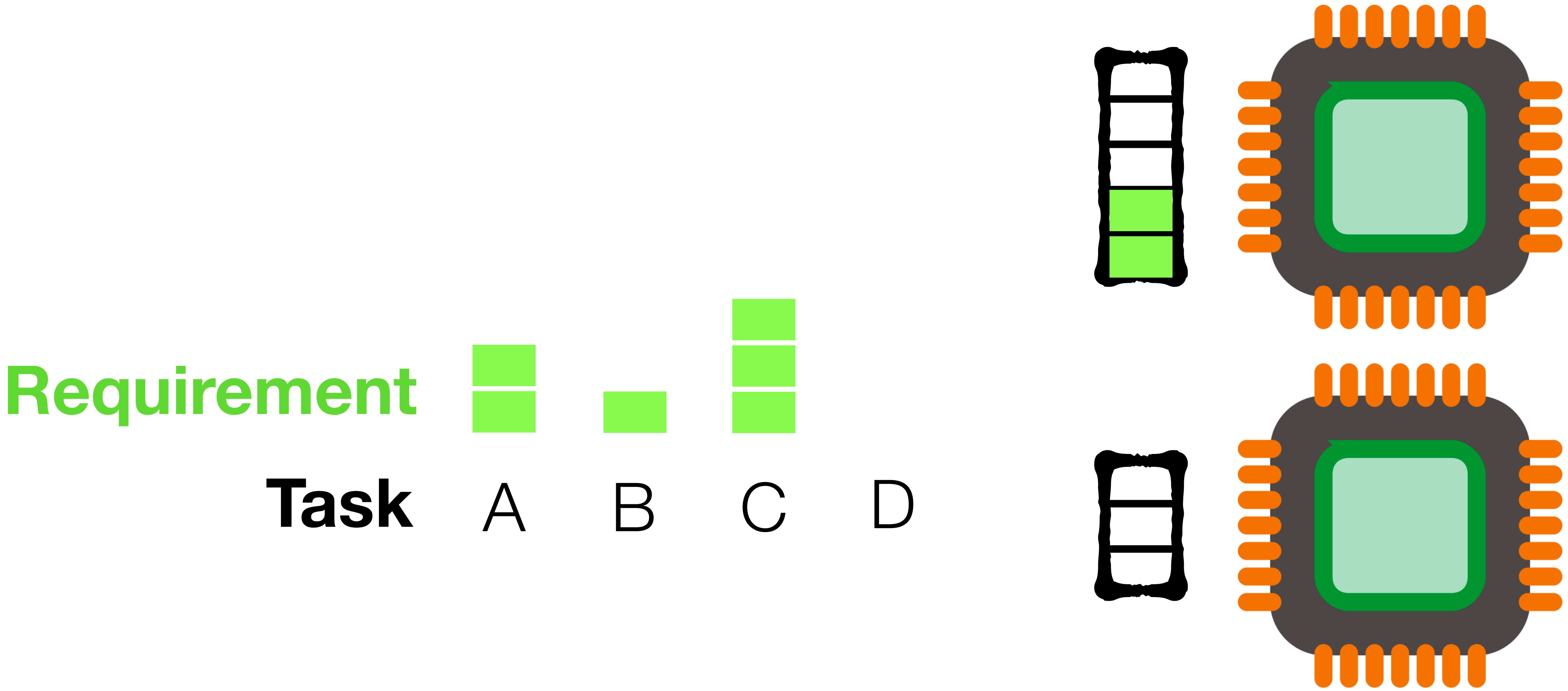
$$\sum_{s=1}^{S} \textbf{mem}(s) \cdot x_{s,m} \leq \textbf{cap-mem}(m) \quad \forall m$$

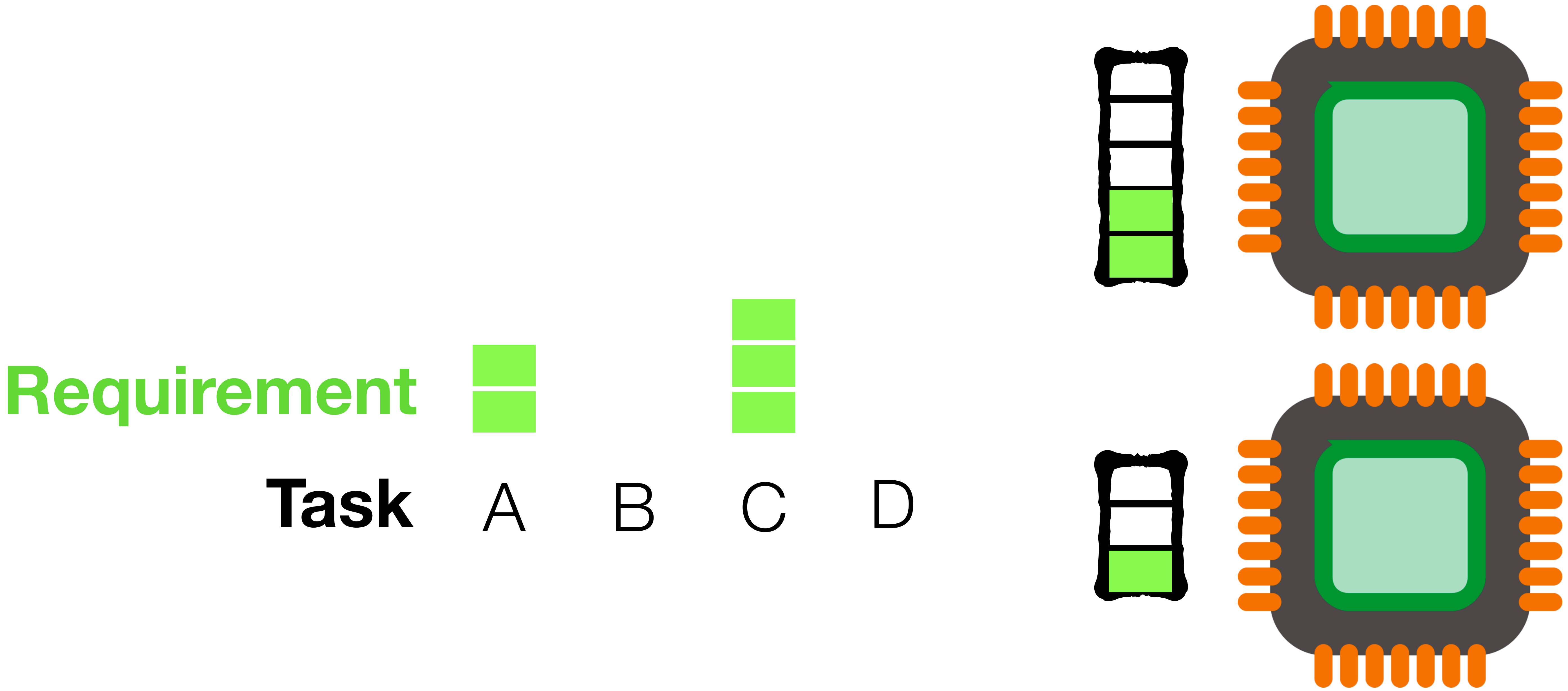
$$\sum_{s=1}^{S} \textbf{cpu}(s) \cdot x_{s,m} \leq \textbf{cap-cpu}(m) \quad \forall m$$

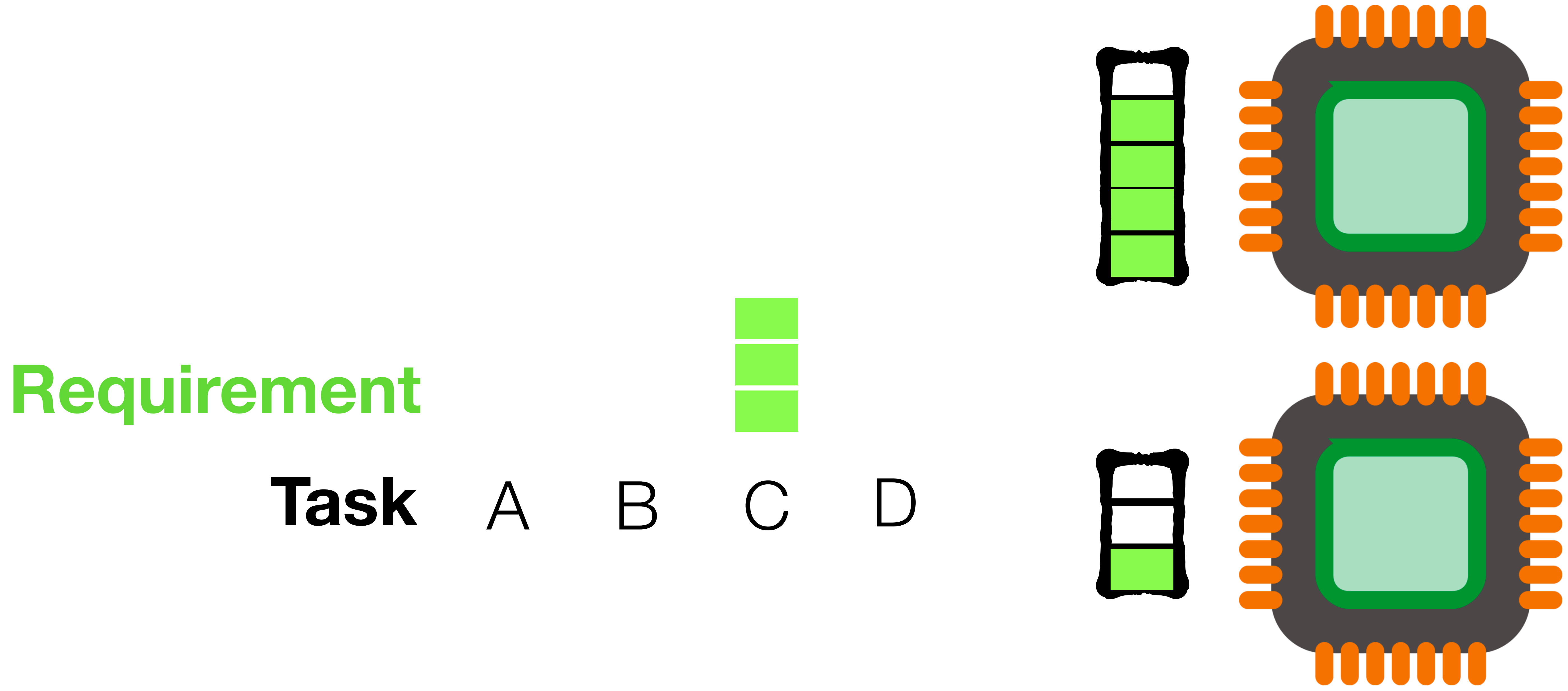
Processor capacity

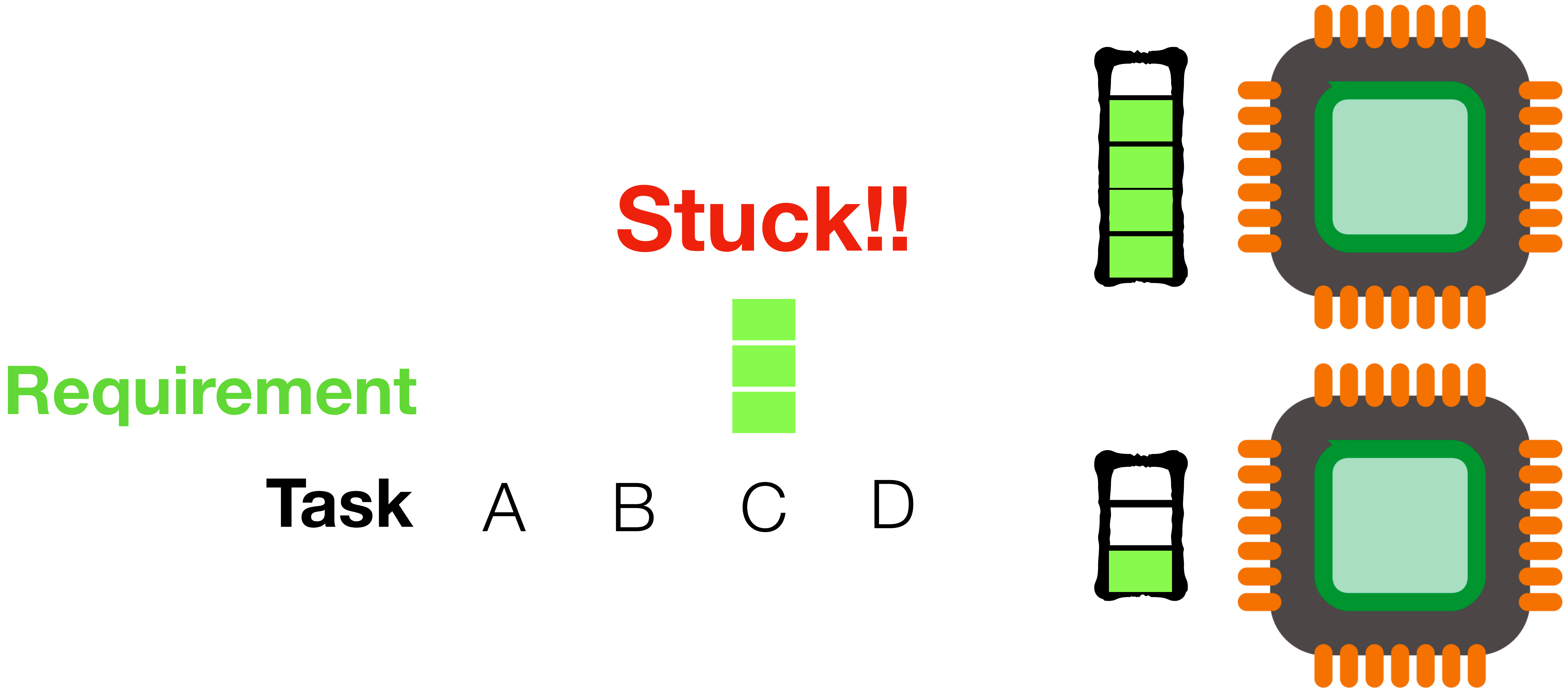
Can you find a feasible solution easily?


Requirement
Task
A
B
C
D


Requirement
Task
A
B
C
D


Requirement
Task A B C D


Requirement
Task
A
B
C
D


Stuck!!
Requirement
Task
A
B
C
D

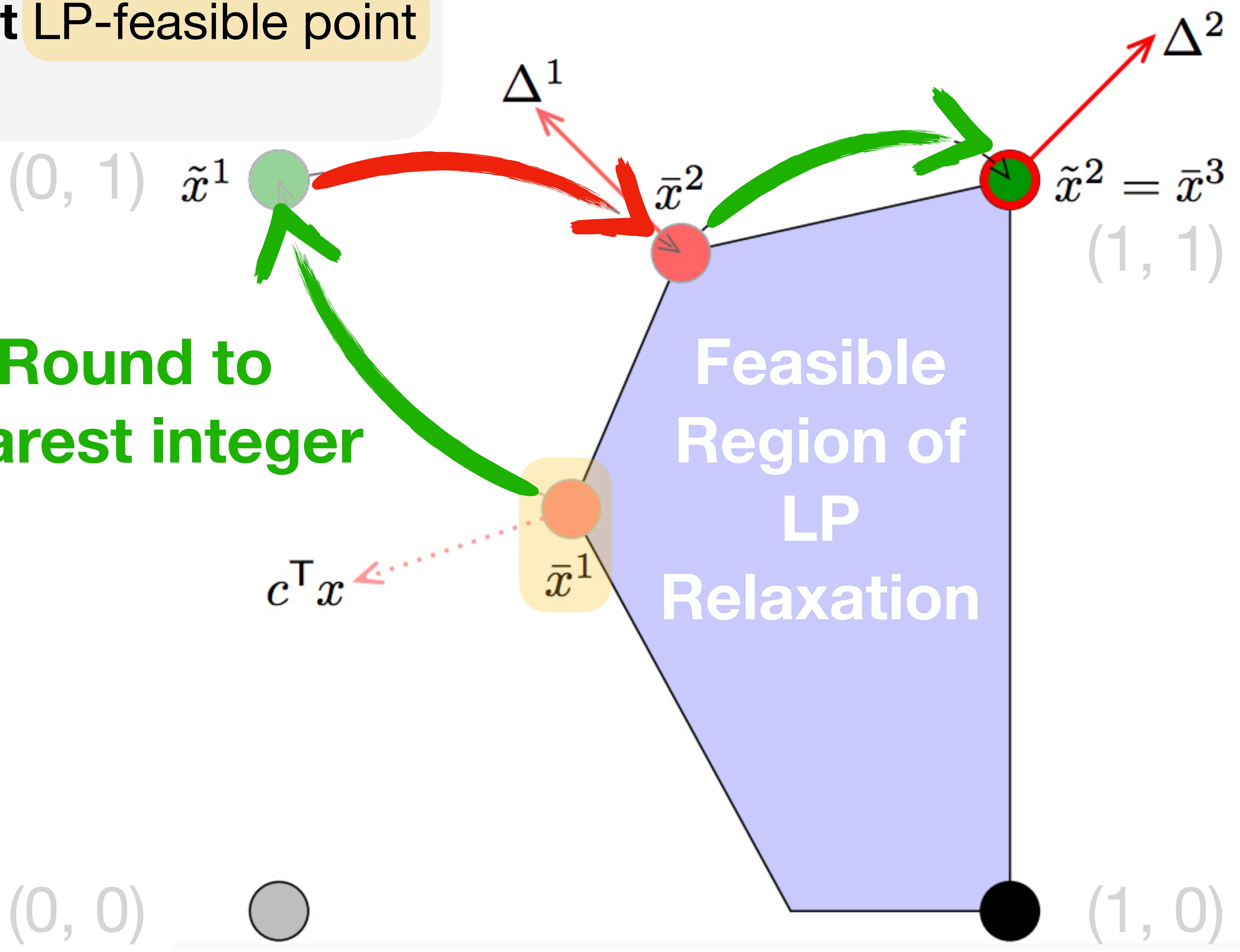
Feasibility Pump

0 Start with LP-feasible (fractional) solution

1 **Round** to nearest integer, **return** if LP-feasible

2 **Project** integer point to **nearest** LP-feasible point

3 Go back to step 1

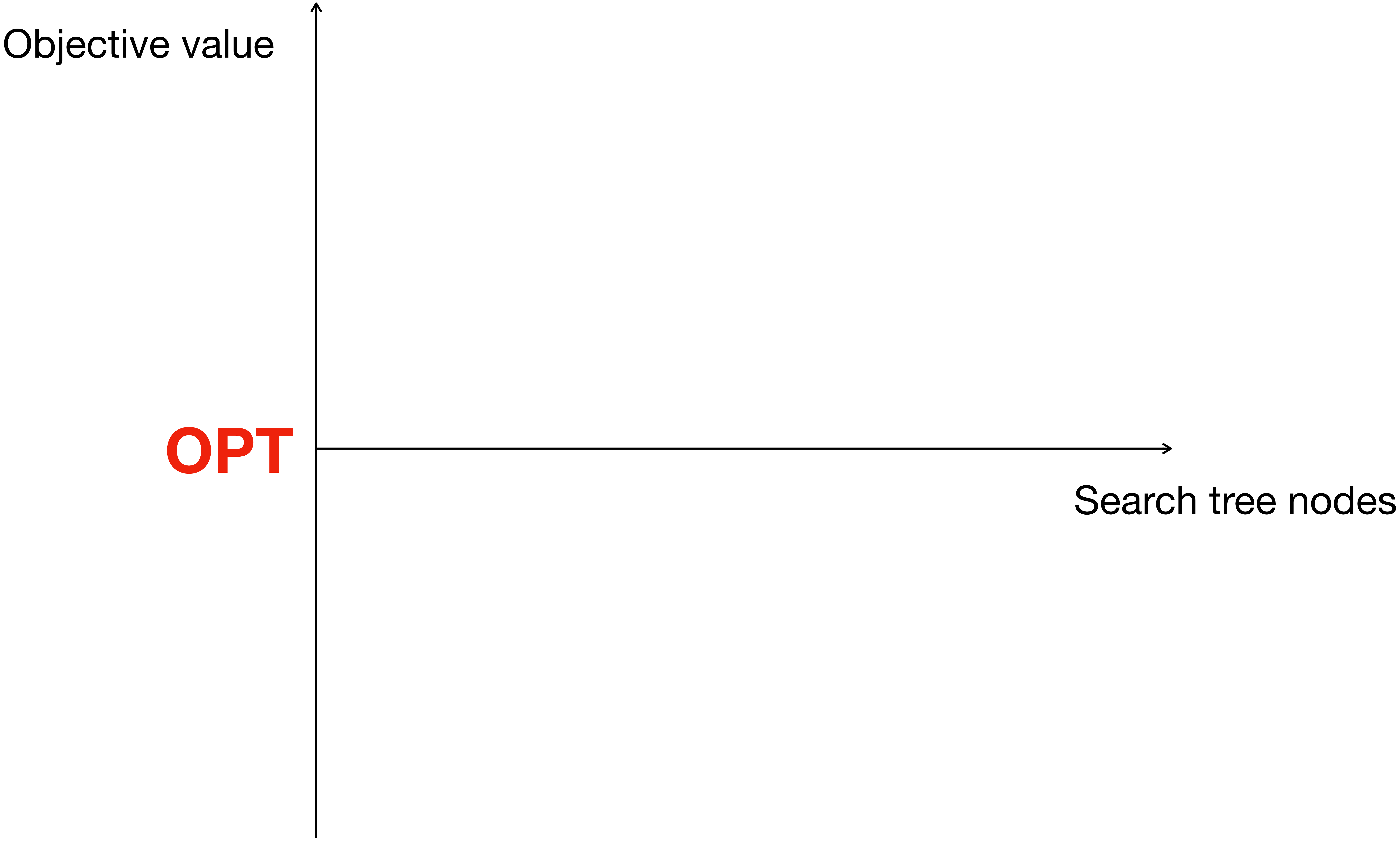


Fischetti, Matteo, Fred Glover, and Andrea Lodi. "The feasibility pump." *Mathematical Programming* 104.1 (2005): 91-104.

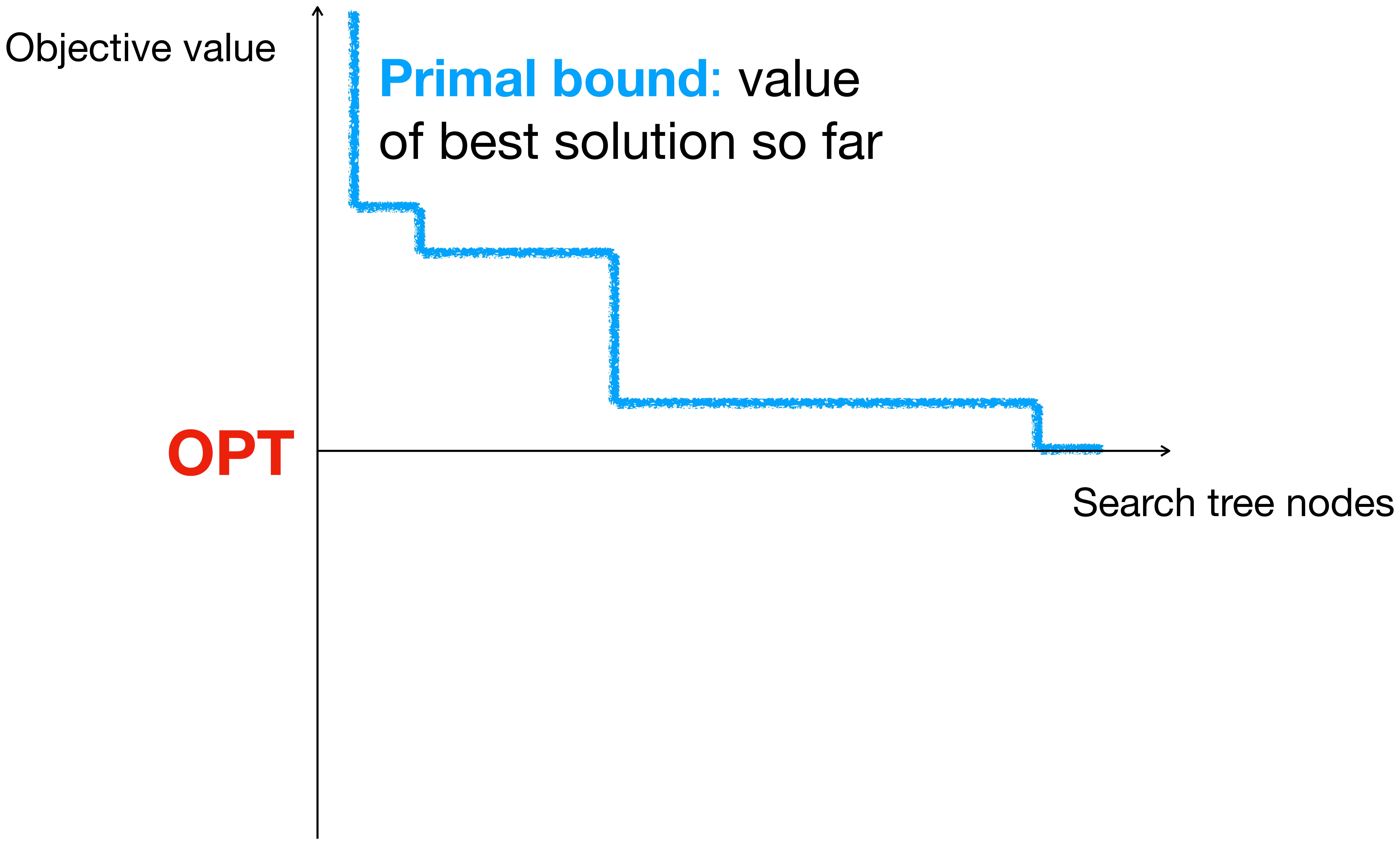
Figure in part from Berthold (2014)

$$\min_x c^T x \textbf{ s.t. } Ax \leq b, x \in \{0,1\}^n$$

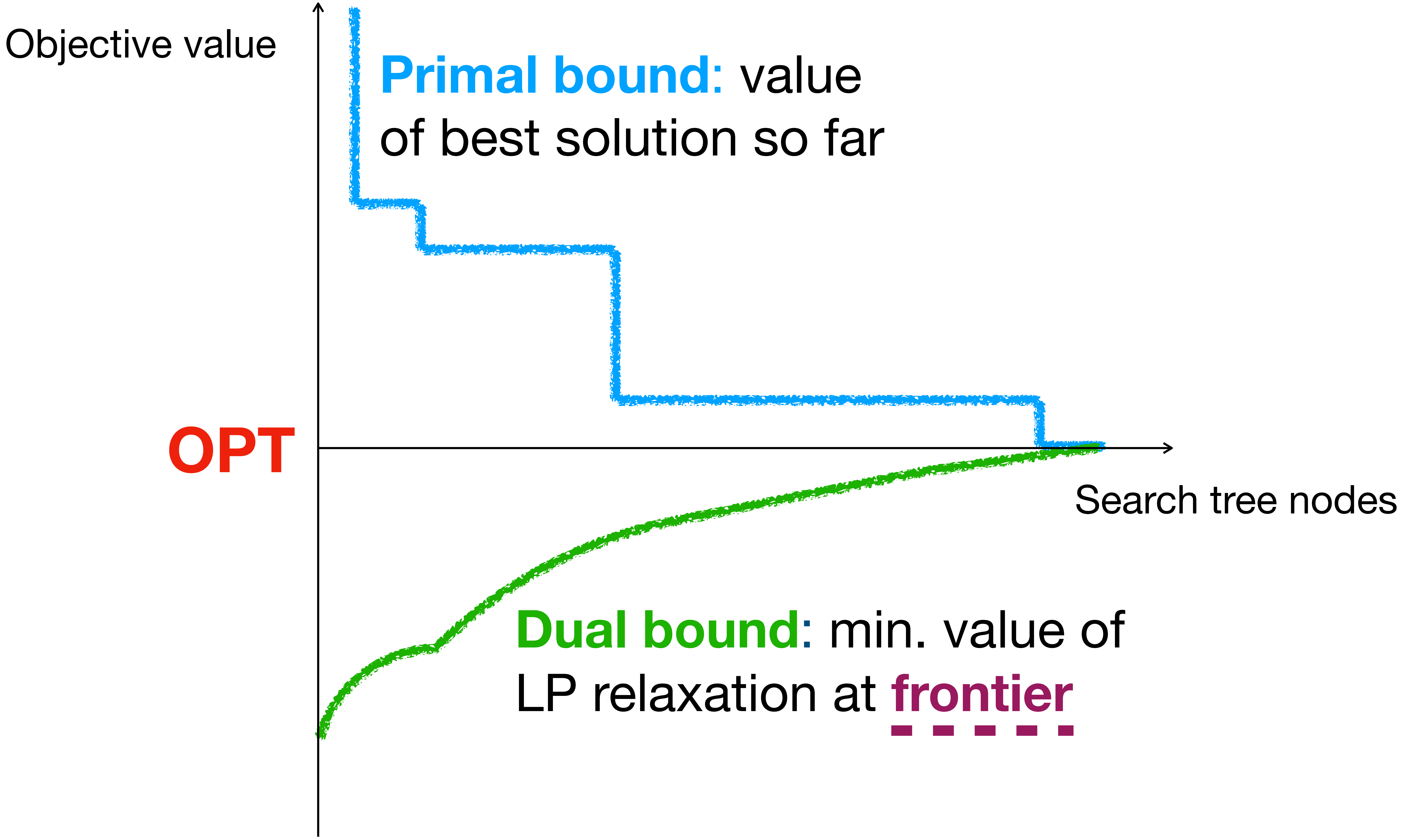
$$\min_x c^T x \textbf{ s.t. } Ax \le b, x \in \{0,1\}^n$$



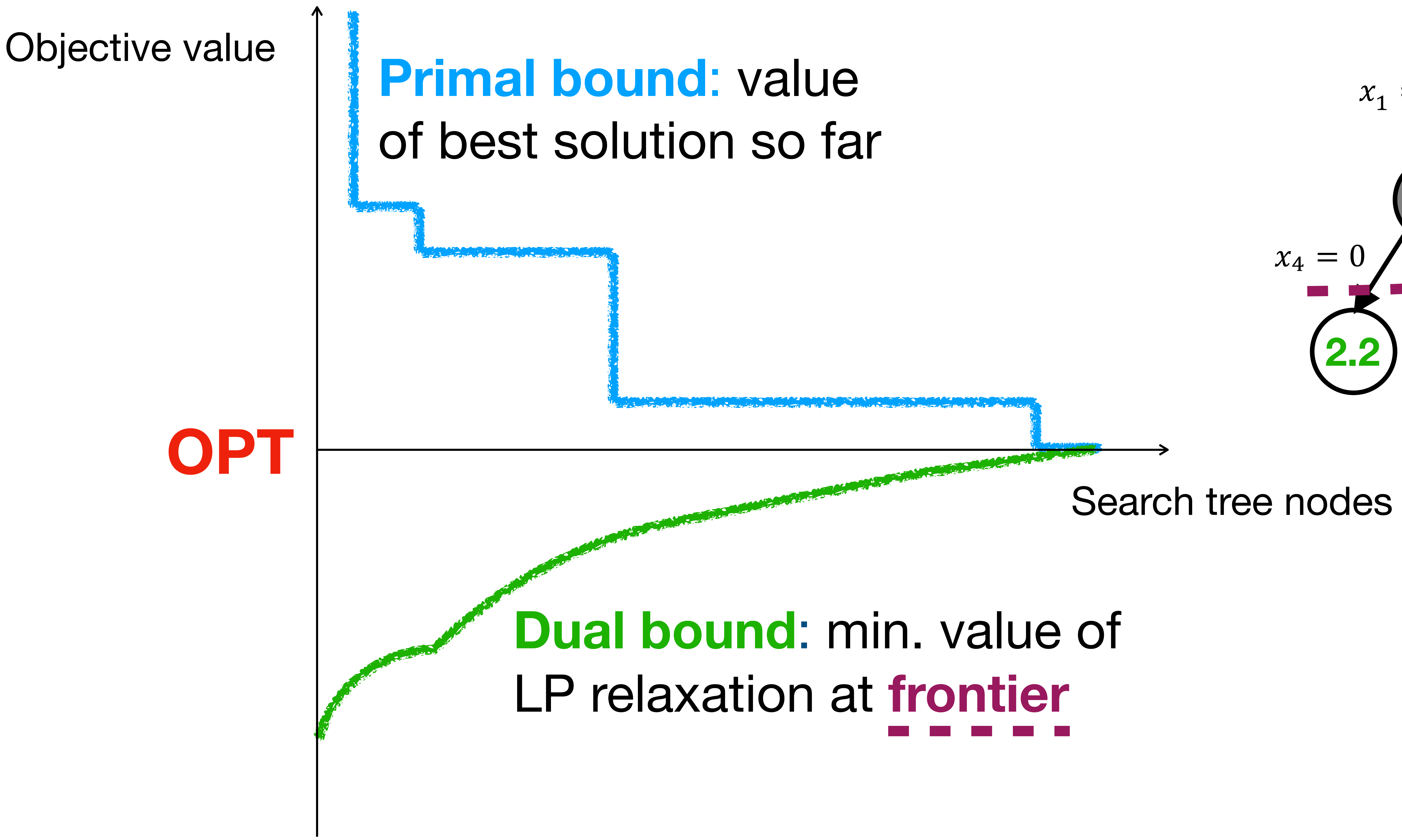
$$\min_x c^T x \textbf{ s.t. } Ax \leq b, x \in \{0,1\}^n$$

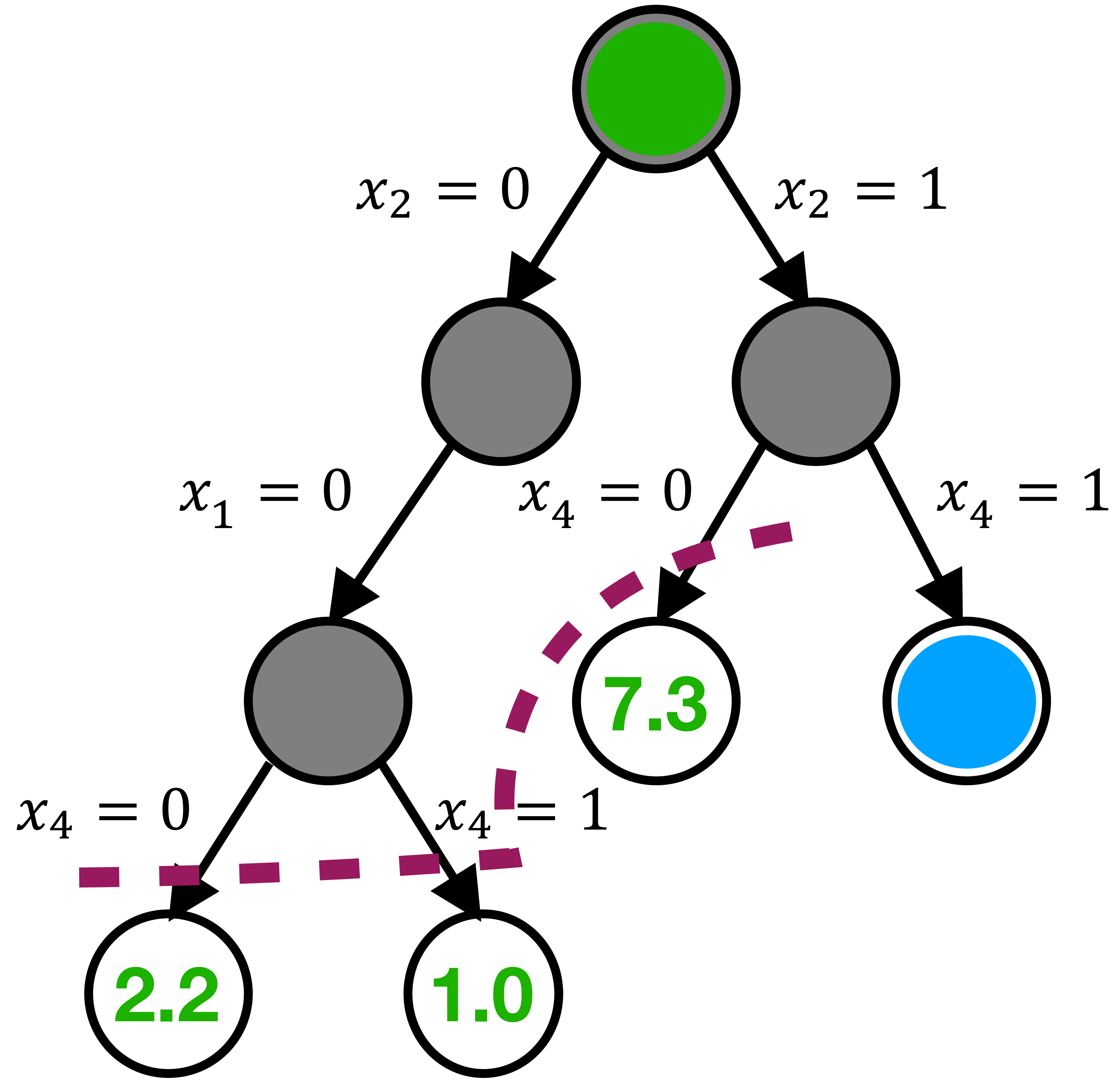


$$\min_x c^T x \textbf{ s.t. } Ax \le b, x \in \{0,1\}^n$$

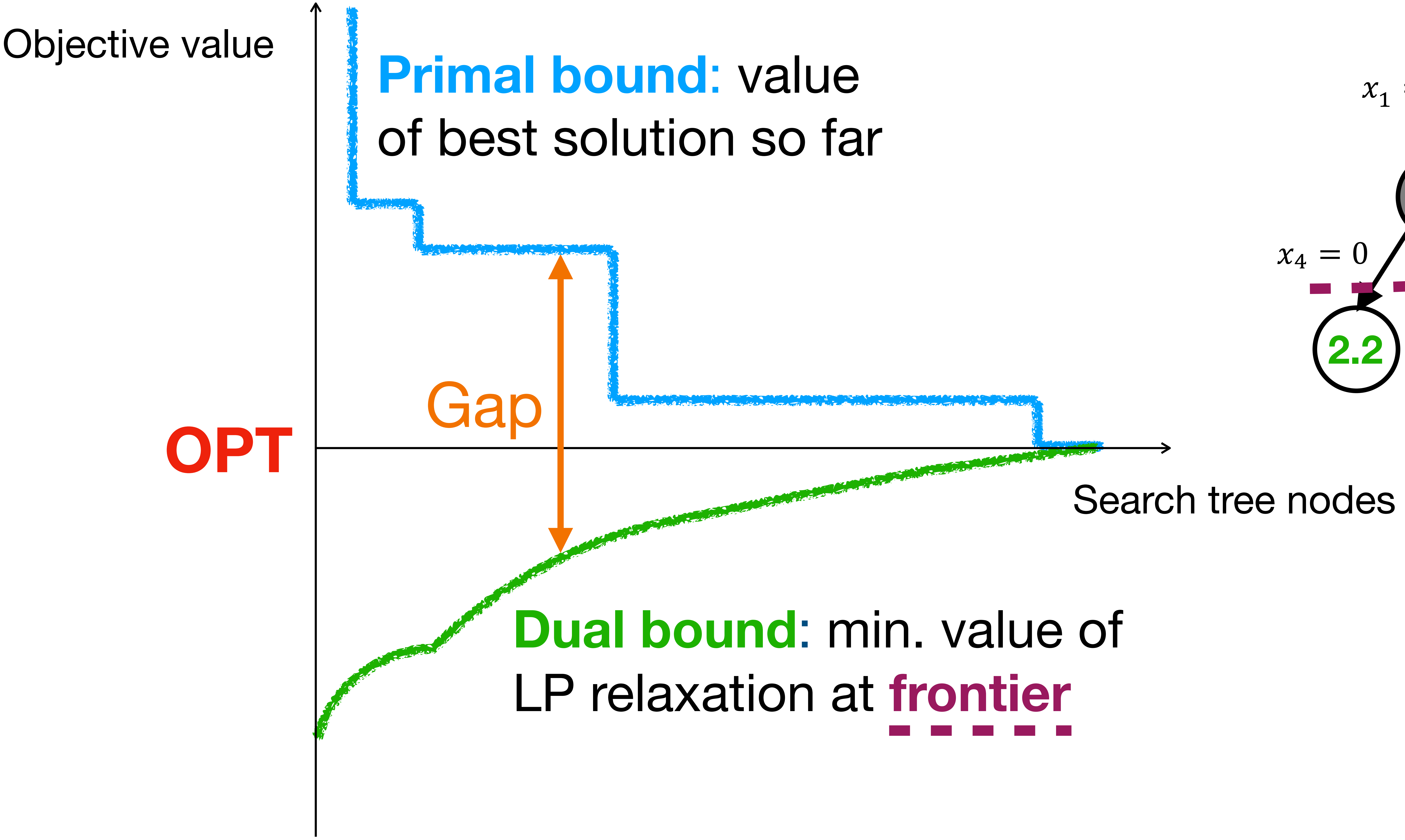


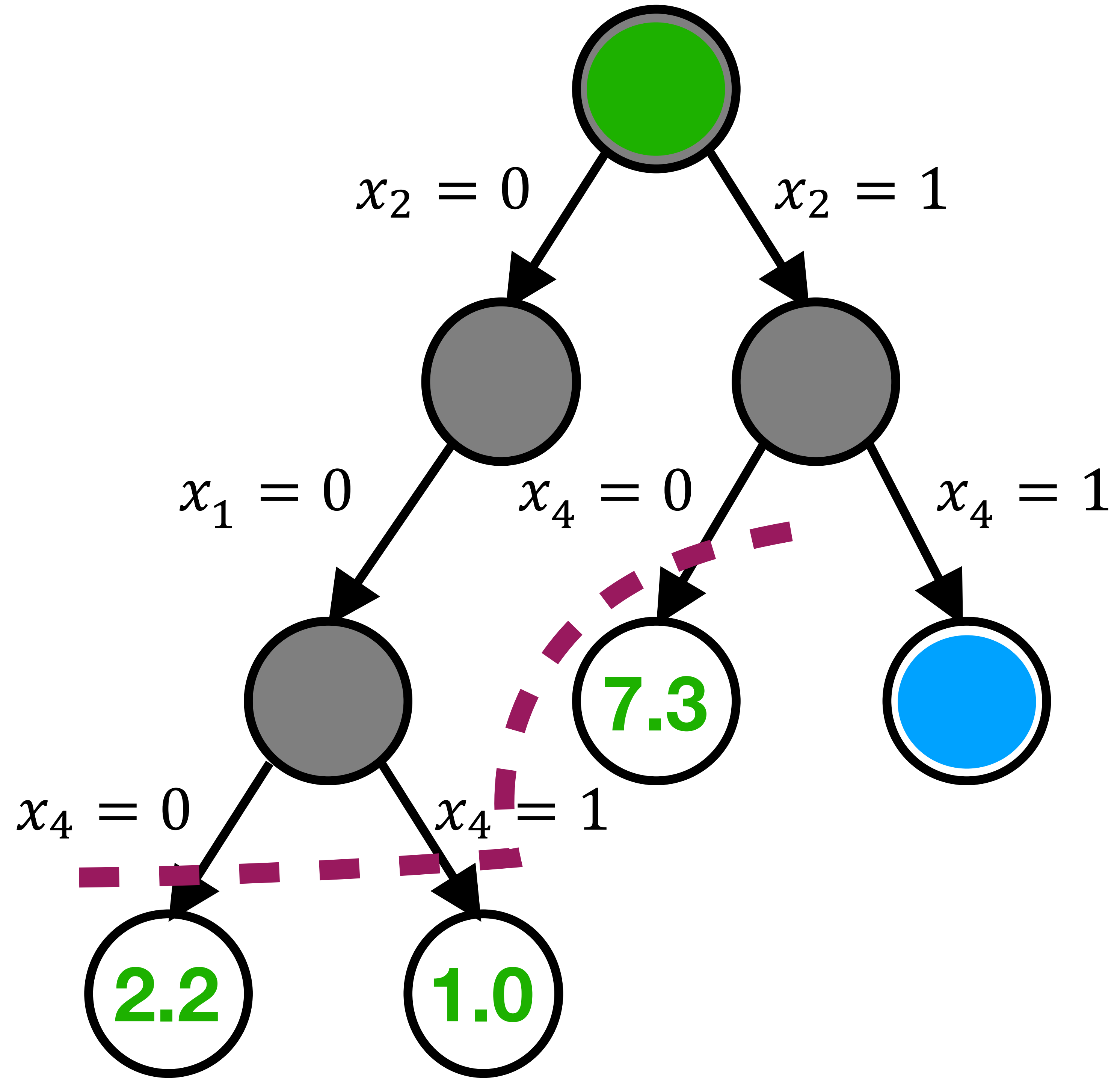
$$\min_x c^T x \textbf{ s.t. } Ax \leq b, x \in \{0,1\}^n$$



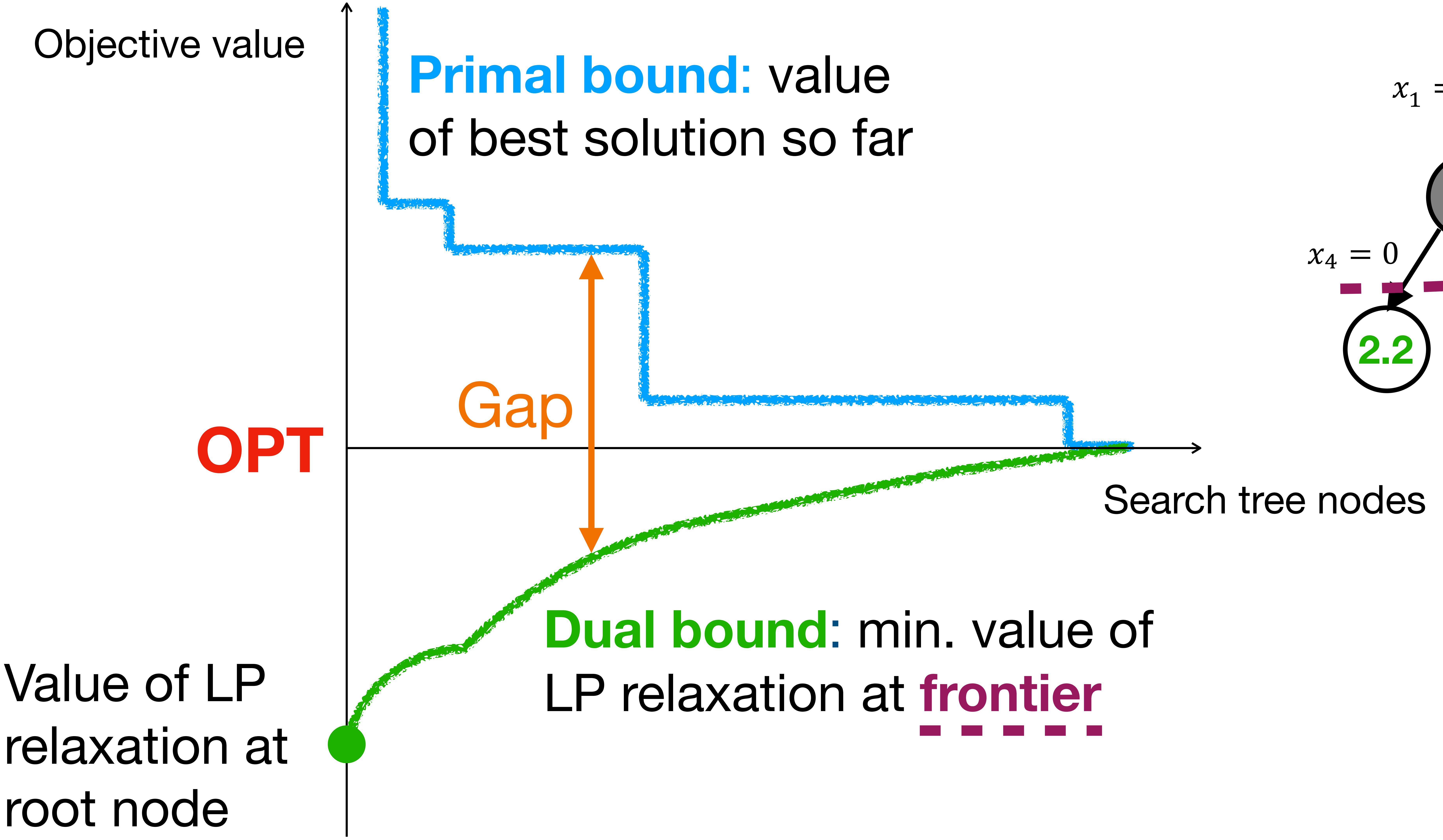


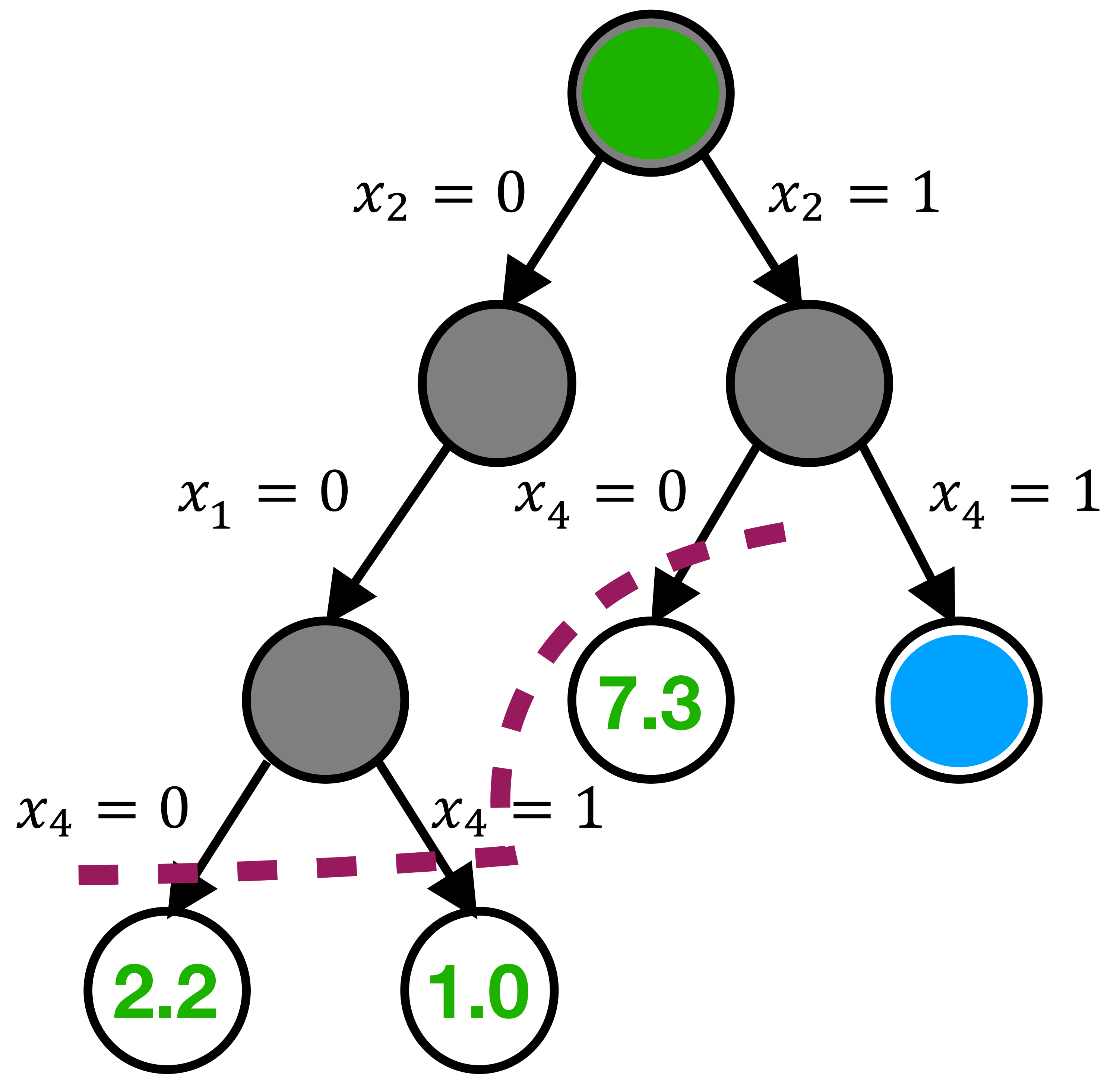
$$\min_x c^T x \textbf{ s.t. } Ax \le b, x \in \{0,1\}^n$$





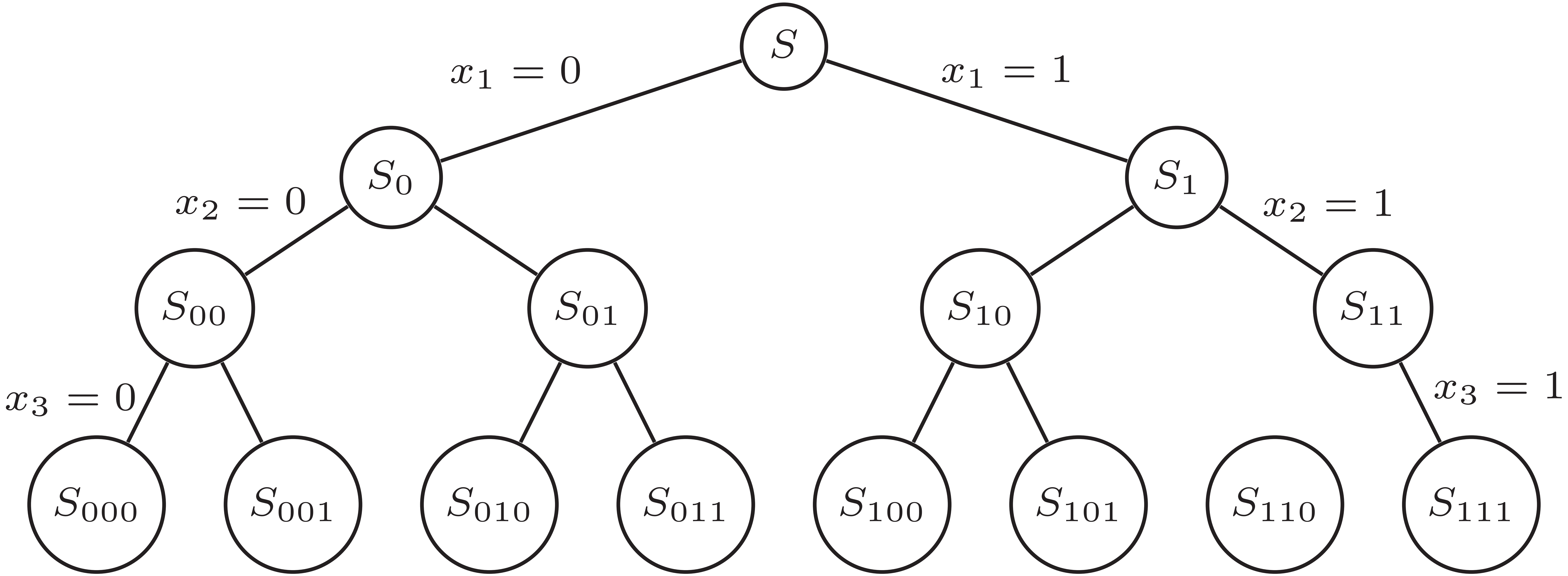
$$\min_x c^T x \text{ s.t. } Ax \leq b, x \in \{0,1\}^n$$





Branch and Bound

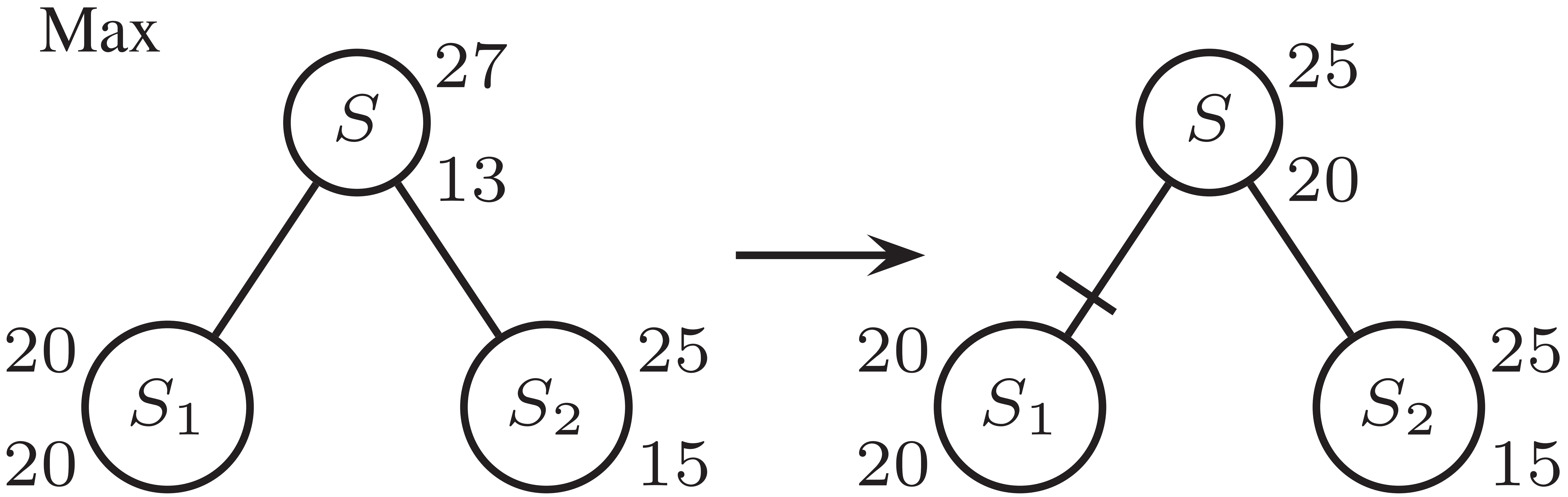
Divide and Conquer + Implicit Enumeration



Branch and Bound

Pruning by Optimality

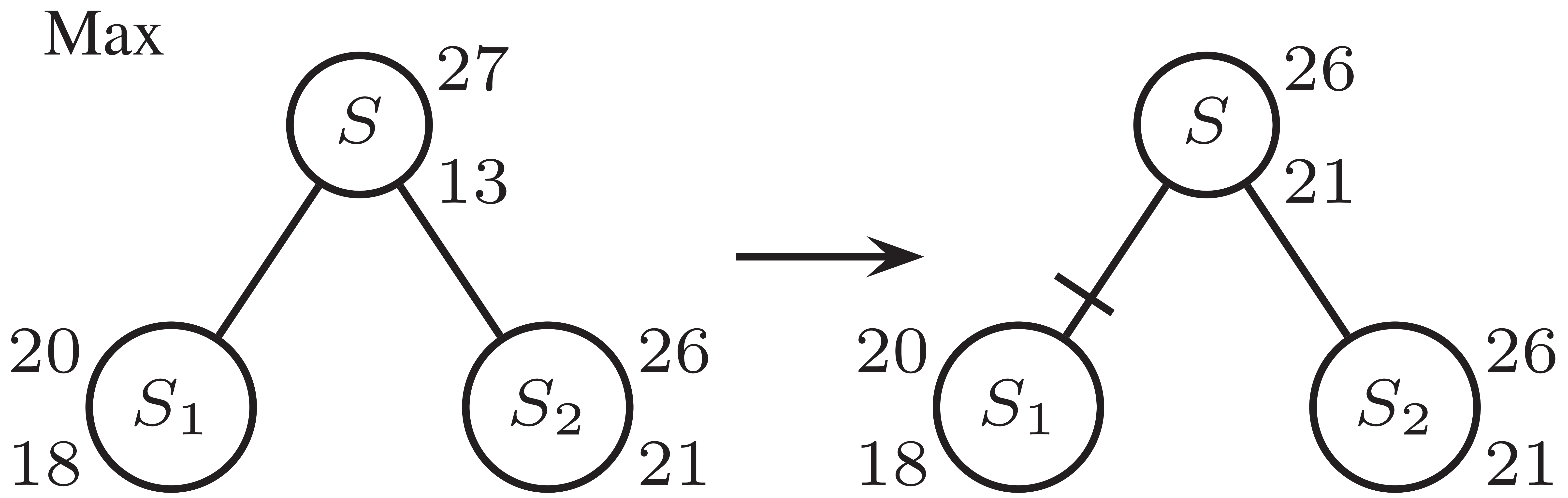
Figure 7.3 Pruned by optimality.



Branch and Bound

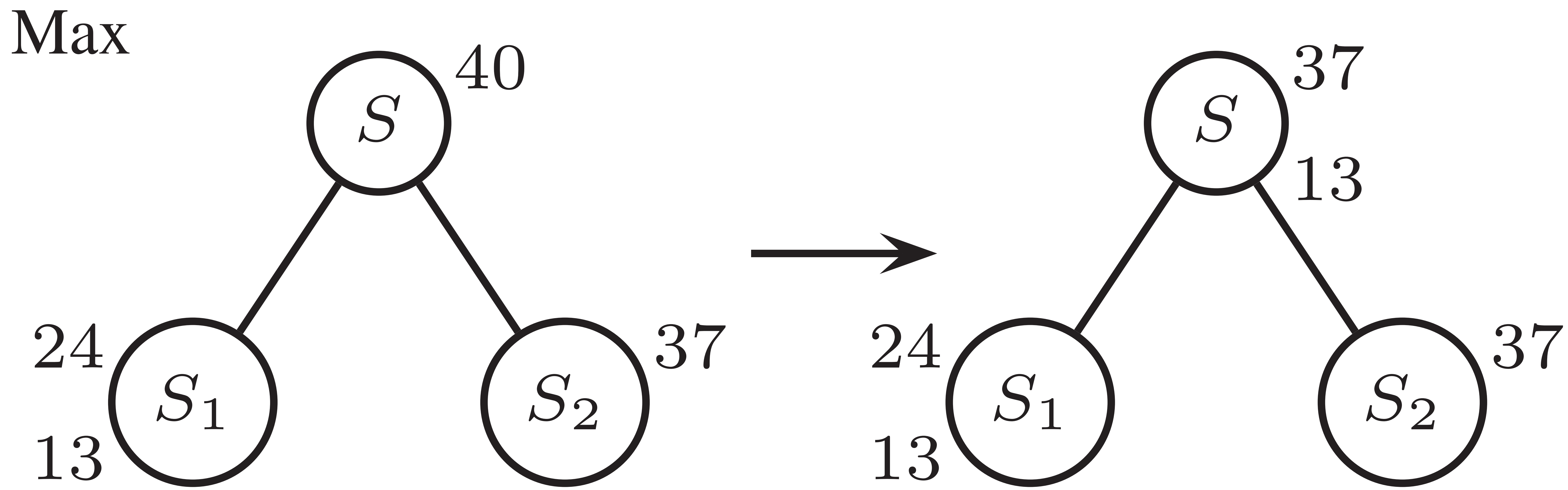
Pruning by Bound

Figure 7.4 Pruned by bound.



Branch and Bound

Pruning is not possible!



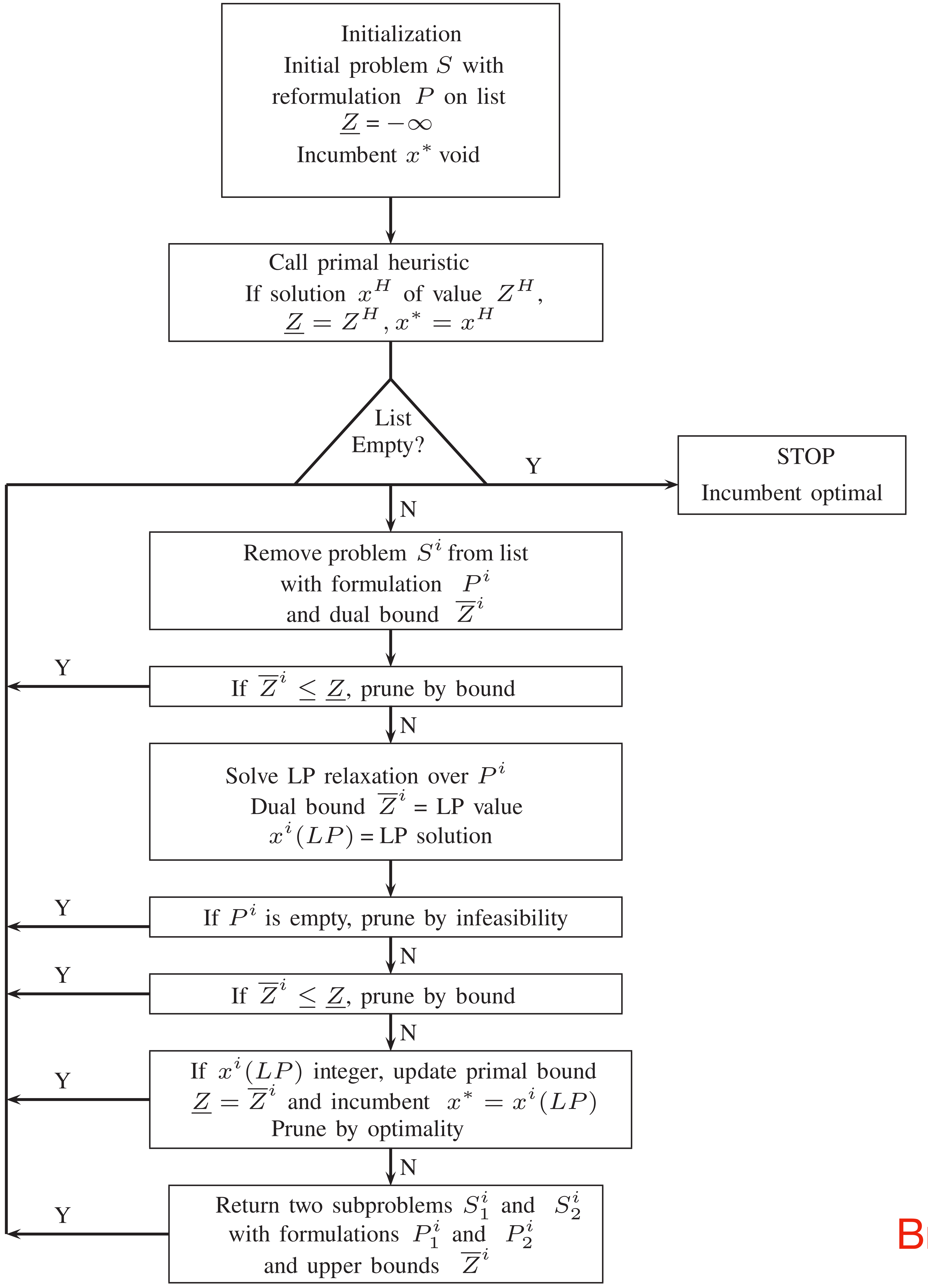


Figure 7.10 Branch-and-bound flow chart.

Primal heuristic

Termination criterion

Node selection

Pruning rule

Solving LP Relaxation

Pruning rule

Pruning rule

Pruning rule

Branching variable selection