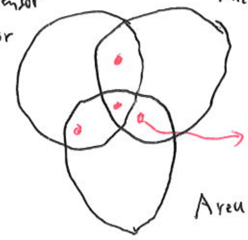
[Du and Wan, Chapter 6]

Set of Sensors $V = \{1, \dots, n\}$



Target Coverage A set of sensors $V' \subseteq V$ that can cover all the targets.

Example $\{1,2\}$, $\{1,3\}$, $\{2,3\}$, $\{1,2,3\}$

Assumption: All sensors can be used for 5 hours, can be changed to any number.

Task: We want to monitor all targets as long as possible.

Ex: Schedule

2 hours	3 hours	1 hour	1 hour
$\{1,2,3\}$	$\{1,2\}$	$\{1,3\}$	$\{1,2,3\}$

Target coverage

Exceed sensor capacity [Sensor 1 is used for 7 hours
Sensor 2 is used for 6 hours

Sensor 3 is used for 4 hours

Not a valid schedule!

Schedule

5 hours
$\{1,2,3\}$

All sensors are used for 5 hours.

Valid solution. → but, we can make it longer.

Schedule

2.5 hours	2.5 hours	2.5 hours
$\{1,2\}$	$\{1,3\}$	$\{2,3\}$

Valid solution → longer than any valid solution.

Observation 1. [The order of target coverage does not matter.

[We can swap the order and have all sensors used at the same duration.]

Our output will be so "How long will we use each sensor coverage" instead of the schedule.

Longest Target Coverage Schedule Problem

Input: Set of targets that each sensor can monitor $S_1, \dots, S_n$

Output: Suppose that all target coverages are $C_1, \dots, C_p$ — p can be as large as 2^n !

Give a time that we use each of them $\boxed{t_1, \dots, t_p}$ — assume to be mostly 0, # non-zero entries $\leq k$

Note: Because p can be as large as 2^n,

the output should be $\langle C_{i_1}, t_{i_1} \rangle, \langle C_{i_2}, t_{i_2} \rangle, \dots, \langle C_{i_k}, t_{i_k} \rangle$

for all $t_{i_j} \geq 0$.

Constraint: All batteries are not used for more than $\boxed{1 \text{ hour}}$. → It does not matter how long we can use each battery as we can scale the result up.

$$a_{i,j} = \begin{cases} 1 & \text{if } i \in C_j \\ 0 & \text{otherwise} \end{cases}$$

$$\sum_{i \in C_j} t_j \leq 1$$

$$\sum_j \underline{a_{i,j} t_j} \leq 1$$

→ The term such that $\underbrace{a_{i,j} = 0}_{i \notin C_j}$ will disappear.

Objective Function: Maximize $t_1 + \dots + t_p$

In another form, ...

Maximize $t_1 + \dots + t_p = \underbrace{\begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}^t}_{c} \cdot \underbrace{\begin{pmatrix} t_1 \\ \vdots \\ t_p \end{pmatrix}}_{x}$

Such that
$$a_{1,1} t_1 + a_{1,2} t_2 + \dots + a_{1,p} t_p \leq 1$$
$$a_{2,1} t_1 + a_{2,2} t_2 + \dots + a_{2,p} t_p \leq 1$$
$$\vdots$$
$$a_{n,1} t_1 + a_{n,2} t_2 + \dots + a_{n,p} t_p \leq 1$$

$$\Downarrow$$

$$\underbrace{\begin{pmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,p} \\ & \vdots & & \\ a_{n,1} & a_{n,2} & \dots & a_{n,p} \end{pmatrix}}_{A} \underbrace{\begin{pmatrix} t_1 \\ \vdots \\ t_p \end{pmatrix}}_{x} \leq \underbrace{\begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}}_{b}$$

→ # columns of $A = p = 2^n$
It can take LP libraries forever to solve this LP.

Maximize $c^t \cdot x$

$Ax \leq b$

LP; can be solved by LP libraries?

$\underline{\text{Ex}}$ Target coverages are $\underbrace{\{1,2\}}_{C_1}, \underbrace{\{1,3\}}_{C_2}, \underbrace{\{2,3\}}_{C_3}, \underbrace{\{1,2,3\}}_{C_4}$.

$$A = \begin{array}{c} \left(\begin{array}{cccc} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} \\ a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4} \\ a_{3,1} & a_{3,2} & a_{3,3} & a_{3,4} \end{array}\right) \\ \begin{array}{cccc} \overline{C_1} & \overline{C_2} & \overline{C_3} & \overline{C_4} \end{array} \end{array} \begin{array}{c} |1 \\ |2 \\ |3 \end{array} = \begin{pmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix}$$

$$b = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \qquad c = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

Maximize $\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}^t \cdot \begin{pmatrix} t_1 \\ t_2 \\ t_3 \\ t_4 \end{pmatrix}$

Primal linear Programming

such that $\begin{pmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} t_1 \\ t_2 \\ t_3 \\ t_4 \end{pmatrix} \leq \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$

$$\Updownarrow$$

~~Maximize~~
Minimize $\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}^t \cdot \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix}$

Dual linear programming

$$\begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} \geq \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

Minimize $y_1 + y_2 + y_3$

such that
$$\left.\begin{aligned} y_1 + y_2 &\geq 1 \\ y_1 + y_3 &\geq 1 \\ y_2 + y_3 &\geq 1 \\ y_1 + y_2 + y_3 &\geq 1 \end{aligned}\right\}$$

The sum of y value in all coverages are larger than 1.

Minimize $y_1 + y_2 + \dots + y_n$

$$a_{1,1} y_1 + a_{2,1} y_2 + \dots + a_{n,1} y_n \geq 1$$
$$a_{1,2} y_1 + a_{2,2} y_2 + \dots + a_{n,2} y_n \geq 1$$
$$\vdots$$
$$a_{p,2} y_1 + a_{p,2} y_2 + \dots + a_{n,p} y_n \geq 1$$

Garg – Koremann Framework

1: $x \leftarrow \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \begin{pmatrix} t_1 \\ \vdots \\ t_p \end{pmatrix}$ $\quad y \leftarrow \begin{pmatrix} \delta \\ \delta \\ \vdots \\ \delta \end{pmatrix} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}$ $\quad \delta$ is a small value.

2: Find i^* such that $\boxed{a_{1,i}\, y_1 + a_{2,i}\, y_2 + \dots + a_{n,i}\, y_n}$ is minimized.

the constraint in the dual linear program
with the most serious problem

3: If $a_{1,i^*}\, y_1 + a_{2,i^*}\, y_2 + \dots + a_{n,i^*}\, y_n \geq 1$, then terminates.

the most serious problem is not a problem.
No problem. we can terminate.

4: For all j such that $a_{j,i^*} = 1$, $y_j \leftarrow y_j \cdot (1+\theta)$

increasing y_j for $a_{j,i^*} = 1$ will increase

$a_{1,i^*}\, y_1 + a_{2,i^*} y_2 + \dots + a_{n,i^*}\, y_n \Rightarrow$ Fix the most serious problem.

5: $x_{i^*} \leftarrow x_{i^*} + \tau$

3 parameters $\rightarrow \delta, \theta, \tau$

6: Go to Step 2

Example $\quad \theta = 0.2 \quad \tau = 0.4 \quad \tau = 0.4$

Initialize: $\quad x \leftarrow \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad y \leftarrow \begin{pmatrix} 0.4 \\ 0.4 \\ 0.4 \end{pmatrix}$

Step 1:

$i=1 \quad y_1 + y_2 = 0.8$ $\rightarrow$ most serious problem

$i=2 \quad y_1 + y_3 = 0.8$ $\rightarrow i^* = 1$

$i=3 \quad y_2 + y_3 = 0.8$

$i=4 \quad y_1 + y_2 + y_3 = 1.2$

$y_1 \leftarrow y_1 \cdot (1+0.2) = 0.4 \cdot 1.2 = 0.48$

$y_2 \leftarrow y_2\,(1+0.2) = 0.4 \cdot 1.2 = 0.48$

$x_1 \leftarrow x_1 + 0.4 = 0.4.$

Step 2: $\quad x \leftarrow \begin{pmatrix} 0.4 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad y \leftarrow \begin{pmatrix} 0.48 \\ 0.48 \\ 0.4 \end{pmatrix}$