EE 459/611: Smart Grid Economics, Policy, and Engineering

Lecture 7: Unit Commitment

Luis Herrera
Dept. of Electrical Engineering
University at Buffalo

Fall 2020

Topics that Will be Covered

(U.C.)

- ***Unit Commitment Problem Formulation***
- * *Dynamic Programming* *
- * *Mixed Integer LP Formulation* *
- → *Other Solution Methods*
- *Examples*

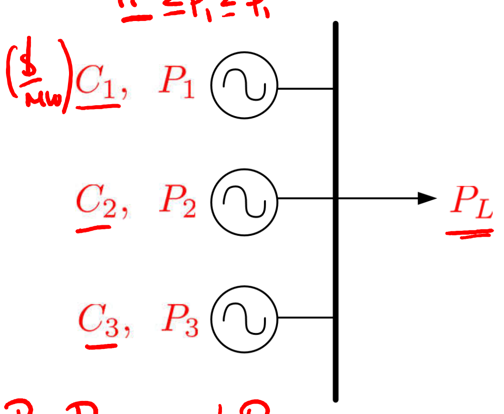
} 2 methods to solve the U.C. problem

Economic Dispatch vs Unit Commitment

- * **Economic Dispatch:*** (ignore network)
  Having N_g generators **connected** to the network, what is the optimal policy (power levels) that minimizes the total cost of operation? (while supplying the load)

$$\begin{aligned} \min \quad & C_1P_1 + C_2P_2 + C_3P_3 \\ \text{s.t.} \quad & \underline{P_i} \le P_i \le \overline{P_i} \quad i = 1, 2, 3 \\ & P_1 + P_2 + P_3 = P_L \end{aligned}$$

$\underline{P_1} \le P_1 \le \overline{P_1}$



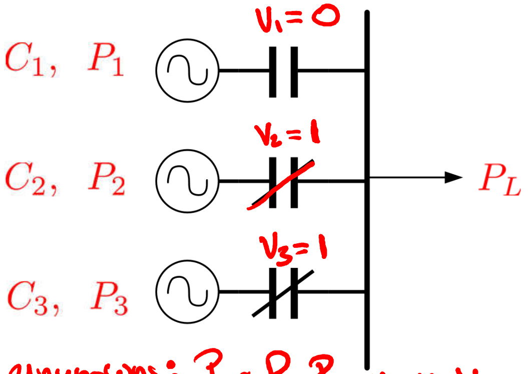
Unknowns: P_1, P_2, P_3 LP

- **Unit Commitment** → we have the option to turn on/off generator
  Having N_g generators "**available**", which set of generators should be online to minimize the cost of operation while meeting the load? We need to take V_i into account in the opt. problem

→ *
$$\boxed{\begin{array}{ll} \min \ C_1P_1 + C_2P_2 + C_3P_3 & \\ V_i\underline{P_i} \le P_i \le V_i\overline{P_i} & i = 1, 2, 3 \\ P_1 + P_2 + P_3 = P_L & \\ V_i \in \{0, 1\} & i = 1, 2, 3 \end{array}}$$

$$\begin{cases} \text{if } V_i = 1 \\ \quad \underline{P_i} \le P_i \le \overline{P_i} \\ \text{elseif } V_i = 0 \\ \quad P_i = 0 \ * \end{cases}$$

$0 \le P_i \le 0 \Rightarrow P_i = 0$

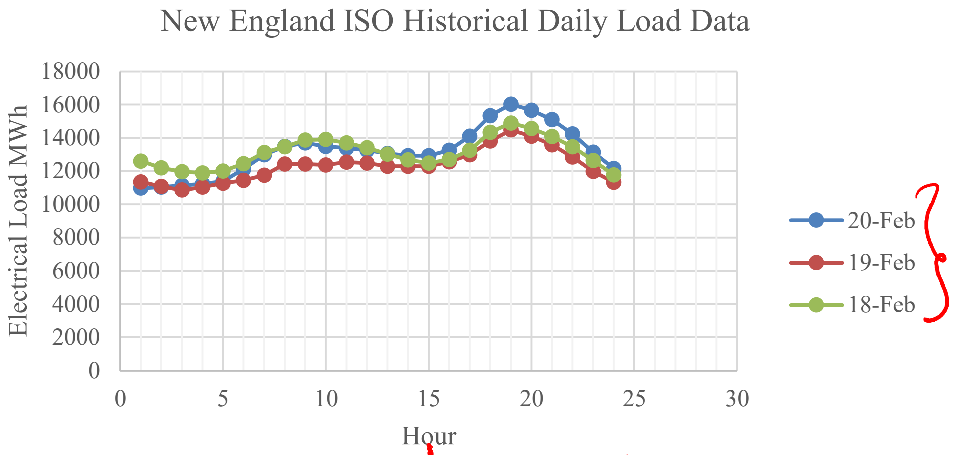


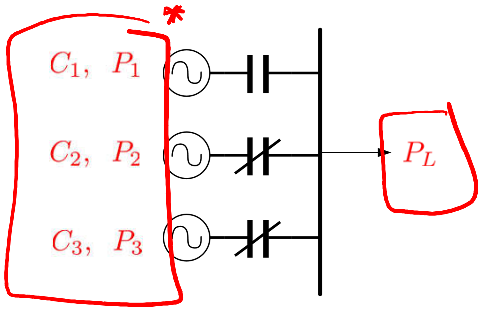
Unknowns: $P_1, P_2, P_3, V_1, V_2, V_3$

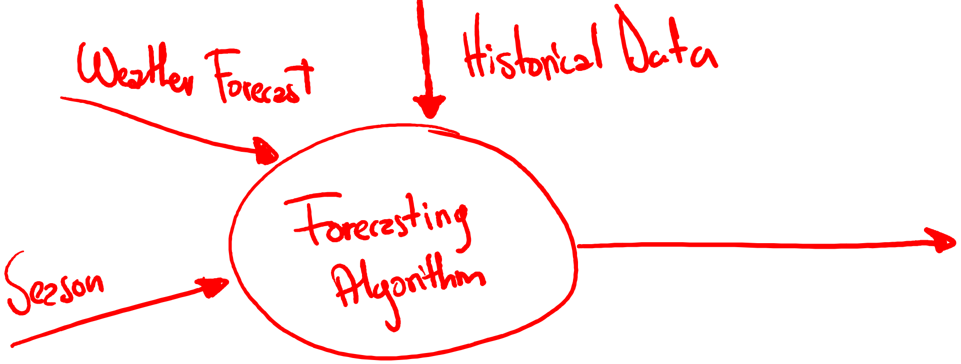
- * Besides the power level, we can also control the "status" of the generator, e.g. on (1) or off (0) → Mixed Integer Linear Program (MILP)

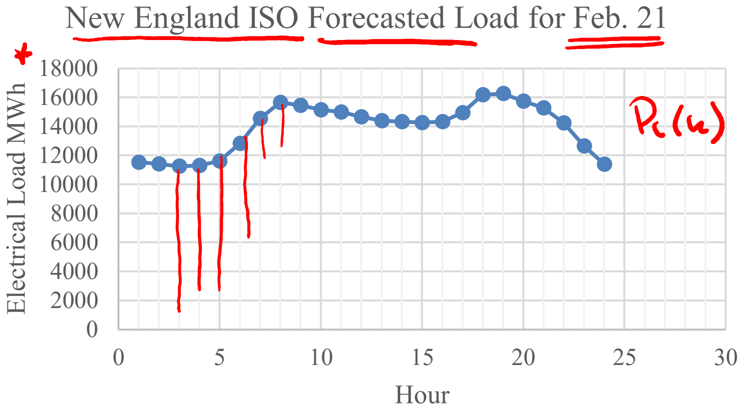
Unit Commitment - Planning Horizon

- To take into account the startup/shutdown cost, the unit commitment problem considers a **planning horizon* (e.g. next day, week, etc.)** $\underbrace{k = 1,\ 2,\ \ldots,\ 24}$ *









- ***We can consider forecasted load through historical data to use in the problem**

Unit Commitment - Planning Horizon

- Starting up an electric power thermal unit (e.g. coal plant) can have a very high costs
- To take into account the startup/shutdown cost, the unit commitment problem considers a **planning horizon* (e.g. next day, week, etc.)** $k = 1,\ 2,\ ...,\ 24$

$$\min \sum_{k=1}^{24} C_1P_1(k) + C_2P_2(k) + C_3P_3(k)$$

s.t.

$$\left.\begin{array}{l} v_i(k)\underline{P_i} \le P_i(k) \le \overline{P_i} \cdot v_i(k) \end{array}\right\} \begin{array}{l} i=1,2,3 \\ k=1,2,\dots 24 \end{array}$$

$$P_1(k)+P_2(k)+P_3(k) = \overset{\checkmark}{P_L}(k) \qquad k=1,\dots,24$$

* Startup cost ∀i

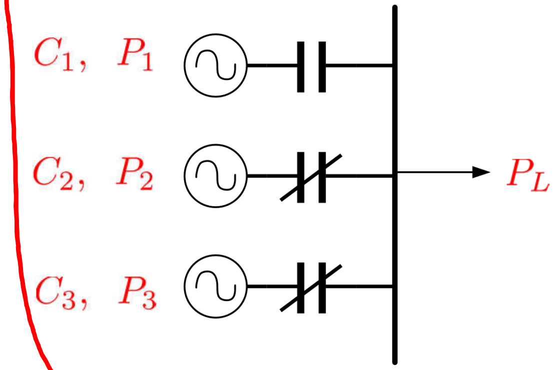
* Shutdown cost ∀k

MILP

Unknowns: $P_1(1),\dots,P_1(24),\ P_2(1),\dots,P_2(24),\ P_3(1),\dots,P_3(24)$ — real

144 variables

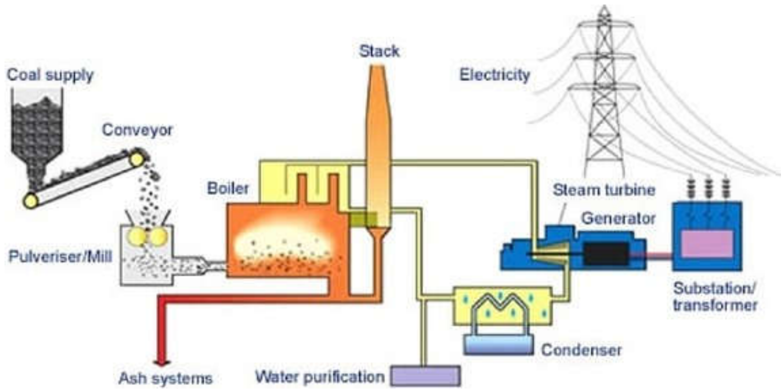
$v_1(1),\dots,v_1(24),\ v_2(1),\dots,v_2(24),\ v_3(1),\dots,v_3(24)$ — integer



- ***We can consider forecasted load through historical data to use in the problem**

Thermal Units – Start Up Costs

- Supposed a thermal unit (e.g. coal power plant) has been offline for a significant amount of time (“cold”)
- In order to bring this unit online, a significant amount of energy/fuel needs to be expended
- This energy does not produce any output power -> take into account as a **start-up cost**



- This cost can be a function of the time the unit has been offline

coal start cost

fixed cost

$$\text{s.u. cost} = C_c\left(1-e^{-t/\alpha}\right)F + C_f$$

if t=0 (just turned off)

if t=100 (offline for 100 hours)

$\lim_{t\to\infty}$ (SU cost) = C_cF+C_f

- We can simplify to be a **constant value** (reasonable simplification)

approximate startup cost by $\boxed{A=C_cF+C_f}$

$$s_j(k) = \begin{cases} 1 & \text{unit is turned on at } k \\ 0 & \text{o.w.} \end{cases}$$

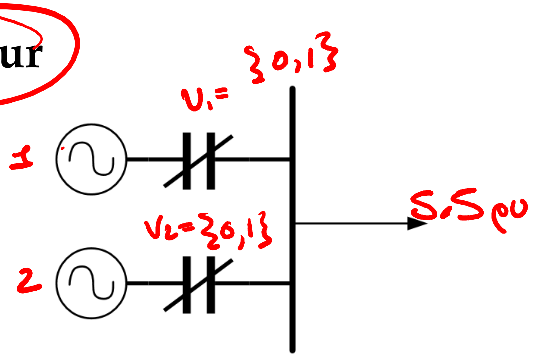
Startup Cost = $As_j(k)$

Unit Commitment - Illustrating Example

- Let's consider a system with **two generators** for a **single hour** *

Gen.	$P_{\min}$	$P_{\max}$	Cost/power
1	1.5	7	7.2 \$/pu
2	1	6	7.8 \$/pu

$P_L = 5.5$



- How many different scenarios can we have with offline/online generators?

Hour 1

For 1 hour and 2 generators: $2^2 = 4$

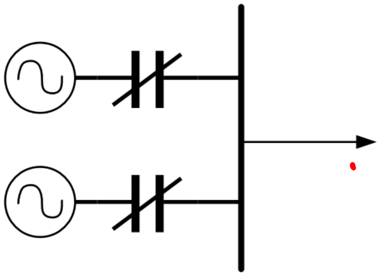
	Gen 1	Gen 2	
Both gens off	0	0	*
Gen 2 is on	0	1	* $P_2^* = 5.5$
Gen 1 is on	1	0	$P_1^* = 5.5$
Both gens on	1	1	$P_1^* = ?$ $P_2^* = ?$ (to minimize cost)

Unit Commitment - Illustrating Example

- Let's consider a system with **two generators** for a **single hour**

Gen.	$P_{\min}$	$P_{\max}$	Cost/power
1	1.5	7	7.2 cheaper
2	1	6	7.8 (\$/pu)

$P_L = 5.5$ ✓



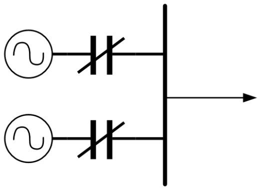
- Based on these different scenarios, which one will provide lowest cost **to meet the load**?

	Unit 1	Unit 2	$P_{\min}$	$P_{\max}$	P_1^*	P_2^*	Cost
Both gens off	0	0	0	0	0	0	∞ we can't supply load
Gen 2 is on	0	1	1	6	0	5.5	5.5x7.8 = 42.9 \$
*Gen 1 on	1	0	1.5	7	5.5	0	5.5x7.2 = 39.6 \$ *
Both gens on	1	1	2.5	13	4.5*	1*	$\underbrace{(4.5)(7.2)}_{\text{gen 1}} + \underbrace{(1)(7.8)}_{\text{gen 2}} = 40.2$ \$

$$\begin{cases} \min\ 7.2P_1 + 7.8P_2 \\ \text{s.t.} \\ 1.5 \le P_1 \le 7 \\ 1 \le P_2 \le 6 \\ P_1 + P_2 = 5.5 \end{cases}$$

Unit Commitment - Illustrating Example

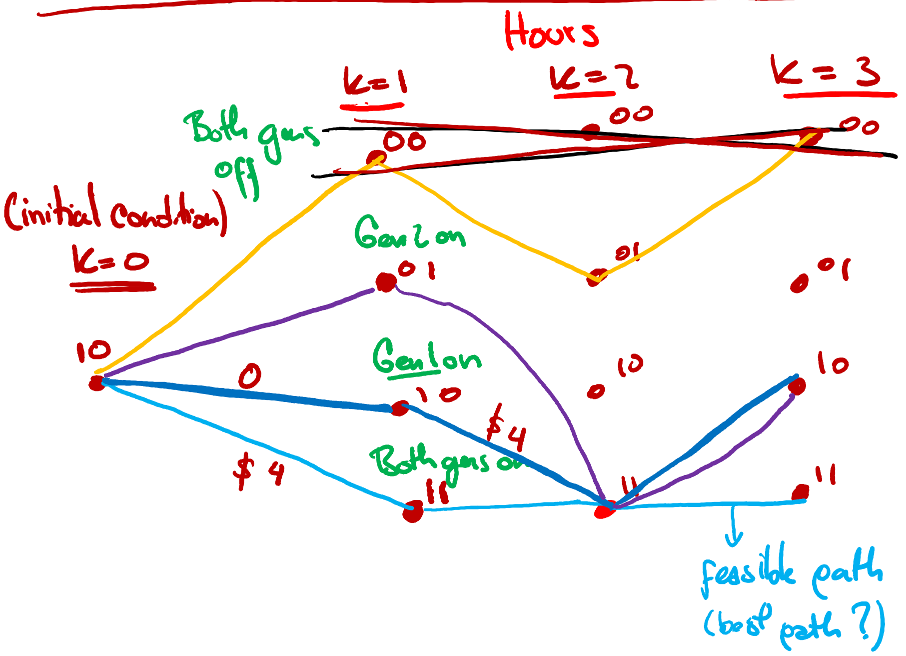
- Let's consider a system with **two generators** for **three hours**



Gen.	$P_{\min}$	$P_{\max}$	Cost/power	Start-up cost
1	1.5	7	7.2	4
2	1	6	7.8	4

Hour	1	2	3
P_L	5.5	8	4

- How many different combinations can we have in this case?



Total # of paths/combinations

$(2^{N_g})^M$

N_g: # of generators

M: Total hours

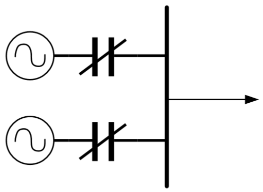
$(2^2)^3 = 64$

if we omit 00 option: $(2^{N_g}-1)^M$

N_{gen}	No. of Combinations for 24 hours
5	6.2×10^{35}
10	1.73×10^{72}
20	3.12×10^{144}
40	(too big) 10^{288}

Unit Commitment - Illustrating Example

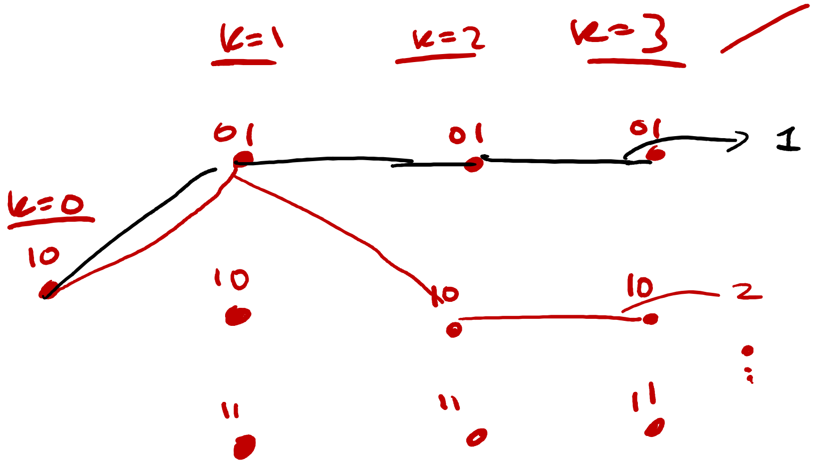
- Let's consider a system with **two generators** for **three hours**



Gen.	$P_{\min}$	$P_{\max}$	Cost/power	Start-up cost
1	1.5	7	7.2	4
2	1	6	7.8	4

Hour	1	2	3
P_L	5.5	8	4

- Draw the overall view of the different paths:



$(2^2-1)^3$

- One way to solve is to compute the cost of all paths (64)
- Pick the path with lowest cost
  ↓
  optimal path
- Not the best way to solve problem

- One way to solve this problem is using ***Dynamic Programming***

Topics that Will be Covered

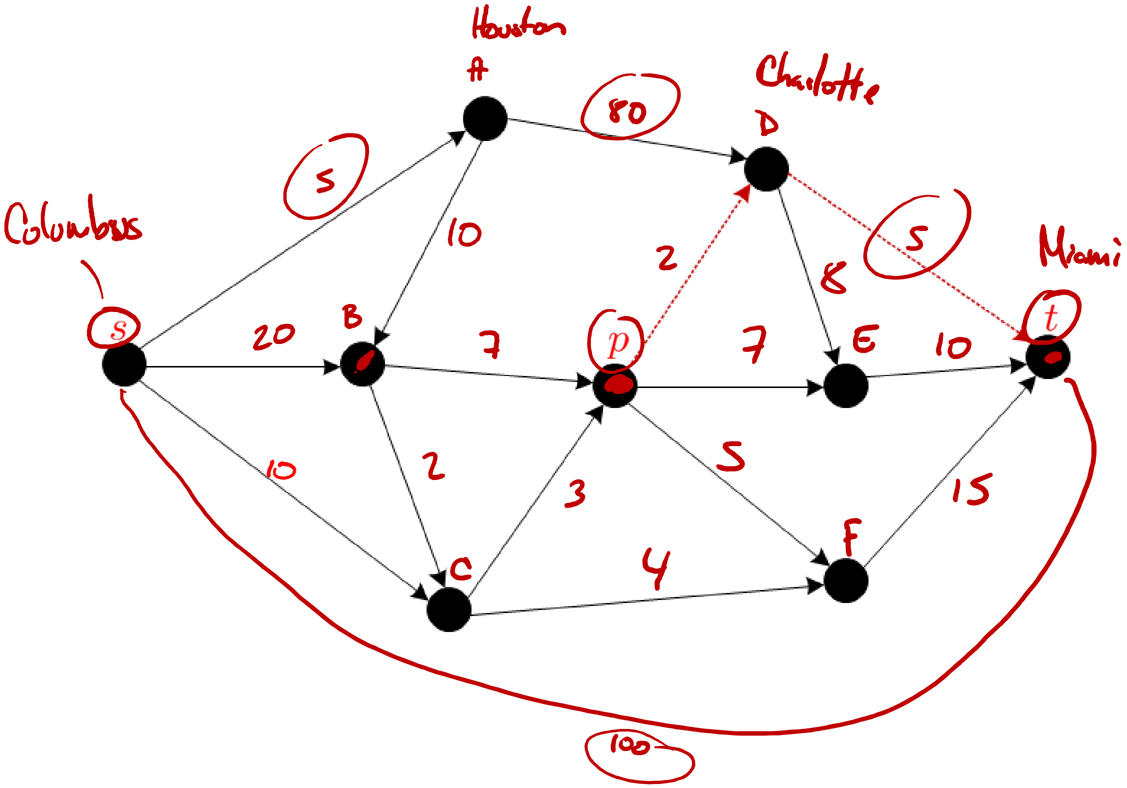
- *Unit Commitment Problem Formulation*
- ***Dynamic Programming*** → *Shortest path problems* → * Unit commitment *
- *Mixed Integer LP Formulation*

(CVX cannot solve MILP) , (CPLEX) , Matlab
× ? ✓

- *Other Solution Methods*
- *Examples*

Dynamic Programming – Optimality Principle

- Consider the problem of finding the shortest path from s to *t*,
  - We are given that if we started at *p*, the optimal path to *t* is shown in dashed lines



- *Naïve Approach:*
  - *Find all possible paths from s-t*
  - *Evaluate the total length*/cost *of each path*
  - *Find the minimum length*/cost

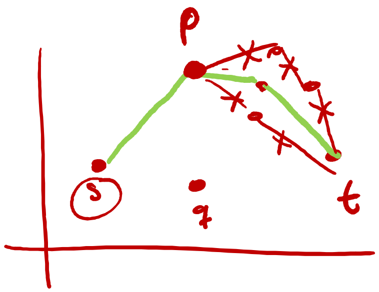
Main Challenges:

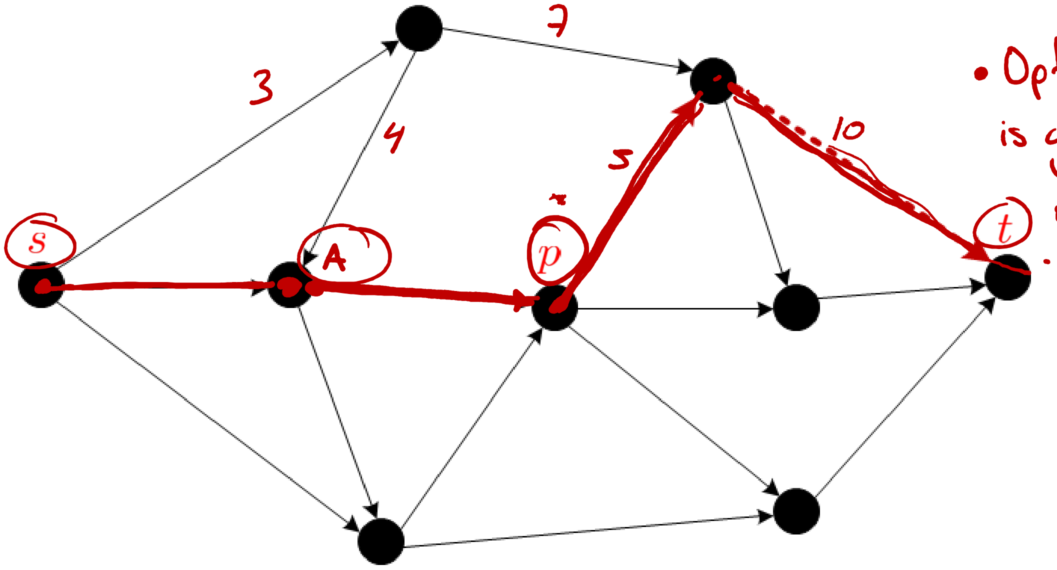
- The total number of paths from *s* to *t* increases exponentially
- Can be infeasible for a very large system

Dynamic Programming – Optimality Principle

- Consider the problem of finding the shortest path from s to *t*,
    - We are given that if we started at *p*, the optimal path to *t* is shown in dashed lines
    - If the optimal path from s to t takes us through p, then the optimal path would go through the optimal path from p to t (already known)

Optimality Principle.





- Now if we start at A
- Optimal path from A to t is going to take us through optimal path from p to t

- ***Optimality Principle:*** An optimal policy (path) contains optimal sub-policies (sub-paths)

Dynamic Programming – Optimality Principle

- ***Bellman's Optimality Principle:***
  An optimal policy (path) contains optimal sub-policies (sub-paths)

- **States:** set of nodes $\mathcal{X} = \{s,\ A,\ B,\ \ldots,\ t\}$ *

- **What is a control policy?** A feasible control sequence (path) $\pi = \{\mu_0,\ \ldots,\ \mu_{N-1}\}$

Example $^*\pi_1 = \{\overset{\mu_0}{sA}, \overset{\mu_1}{AD}, \overset{\mu_2}{Dt}\}$

$^*\pi_2 = \{\underset{\mu_0}{sC}, \underset{\mu_1}{Cp}, \underset{\mu_2}{pE}, \underset{\mu_3}{Et}\}$

- **Total cost associated with a policy:**

$$\overset{\text{cost}}{J_\pi(x_0)} = \sum_{k=0}^{N-1} \underbrace{l_k(x_k, \mu_k(x_k))}_{\text{running cost}} + \underbrace{g_N(x_N)}_{\text{terminal cost}}$$

x_0 = initial state $\quad x_0 = s$

- **Optimal cost:** $J^*(x_0) = \min_\pi J_\pi(x_0)$

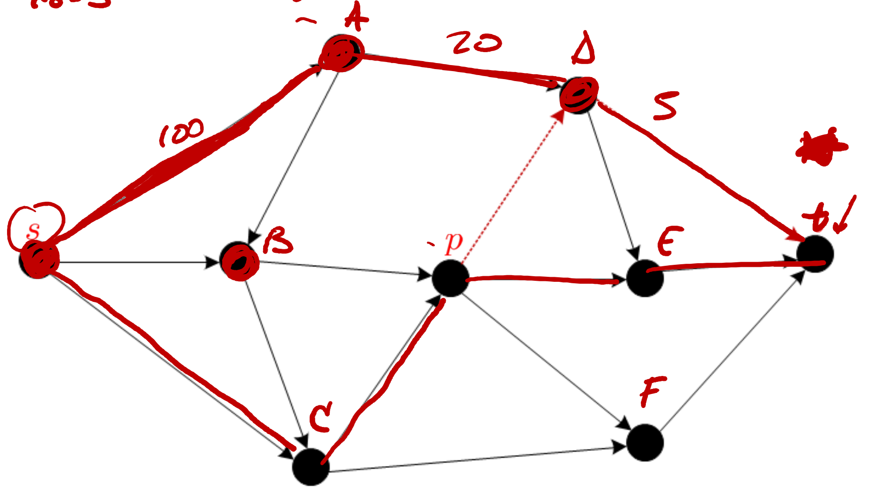
→ picks policy (π_i) with least cost
path

- **Optimal policy:**

$J_{\pi^*}(x_0) = J^*(x_0)$

$\pi^* = \{\mu_0^*,\ \ldots,\ \mu_{N-1}^*\}$ } gives us optimal cost

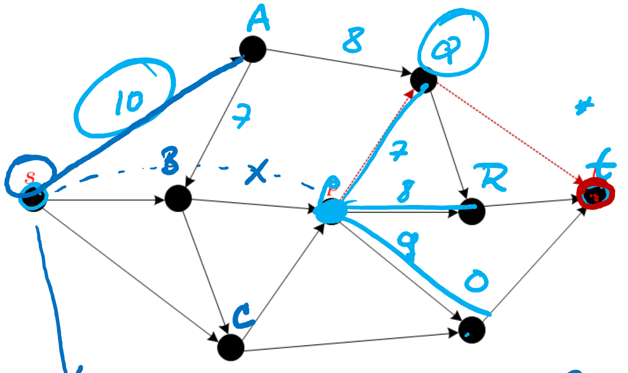
↑ path



Reference: D. Bersekas. Dynamic Programming and Optimal Control. Vol. 1 Athena Scientific

Dynamic Programming Algorithm Definitions

$$J_\pi(x_0) = \sum_{k=0}^{N-1} l_k(x_k, \mu_k(x_k)) + g_N(x_N)$$



possible control sequences at s?

$\mathcal{U} = \{sA, sB, sC\}$

- **Final cost:** (we need to define)

$$J_N(x_N) = g_N(x_N)$$

$$g_N = \begin{cases} 0 & \text{if } x_N = t \\ \infty & \text{o.w.} \end{cases}, \quad g_N(t) = 0$$

- **Running cost:** (need to define)

$$\underbrace{l_k(x_k, \mu(x_k))}$$

$k = 0, \quad x_0 = s$

x_{k+1}

end at A

$$\longrightarrow l_0(x_0, \mu(x_0)) = l_0(s, sA) = \underline{10}$$

start x_k

- **Value function iteration**

start

$x_k \to x_{k+1}$

$$J_k(x_k) = \min_{\mu} \{ \underbrace{l_k(x_k, \overbrace{\mu_k(x_k)})} + \underbrace{J_{k+1}(x_{k+1})} \} \qquad \forall k = 0, \ldots, N-1$$

step k

edge/control sequence

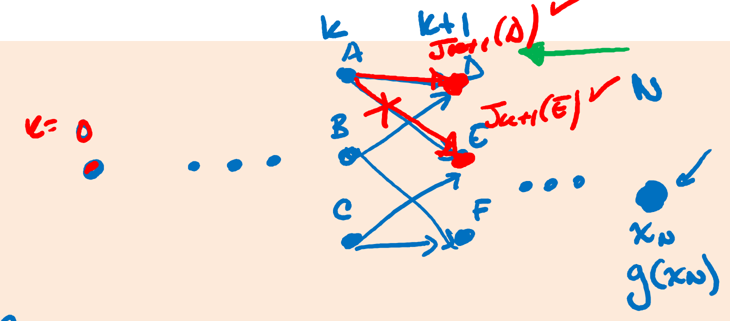
*

- **Optimal control at time *k*:**

$$\mu_k^*(x_k) = \arg\min_{\mu} \{ l_k(x_k, \mu_k(x_k)) + J_{k+1}(x_{k+1}) \} \Rightarrow \text{e.g. } \mu_k^*(P) = PQ$$

Reference: D. Bersekas. Dynamic Programming and Optimal Control. Vol. 1 Athena Scientific

Backwards Dynamic Programming – Algorithm



1. Start at the last step *

$$\underline{J_N(x_N)} = \underbrace{g_N(x_N)}_{\text{final cost}}$$

2. Go backwards in time k using

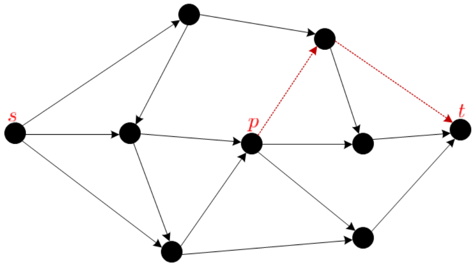
$$J_k(x_k) = \min_{\boxed{\mu}} \{ \; l_k(x_k, \underset{\text{pick best control sequence}}{\mu_k(x_k)}) + \overset{\text{known}}{\underline{J_{k+1}(x_{k+1})}} \; \} \qquad \forall k = 0, \; ..., \; N-1$$

$x_k = A$ $\quad \mu_k(A) = \begin{Bmatrix} A\,D \\ A\,E \end{Bmatrix}$

3. Then, the $J_0(x_0)$ generated in the last step, is equal to the optimal cost

$$\boxed{J_0(x_0) = J^*(x_0)}$$

⇒ We can obtain the optimal policy π^* ✓ /path



Reference: D. Bersekas. Dynamic Programming and Optimal Control. Vol. 1 Athena Scientific

Backwards Dynamic Programming – Shortest path Example

- Consider the problem of traveling from city *A* to city *P*, use Dynamic Programming to find the shortest path!

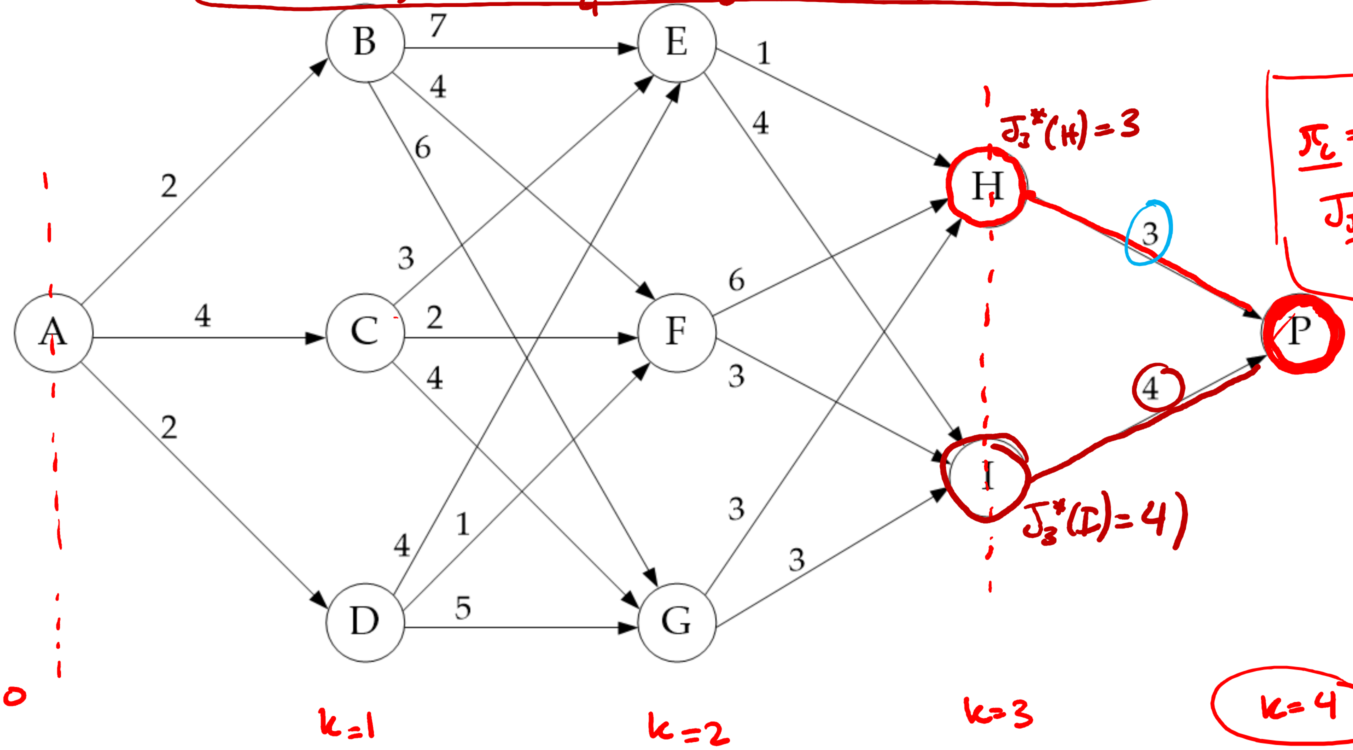
@k=4 (Last step) Define terminal/final cost

$$\boxed{J_4(x_4 = P) = g_4(P) = 0}$$

Move backwards

@k=3 $x_3 = H \rightarrow$ $J_3^*(H) = \min_{\mu} \{ l_3(H,\mu) + J_4(x_4) \} = \min_{\mu} \{ \underbrace{l_3(H, HP)}_{3} + \underbrace{J_4(P)}_{0} \} = \boxed{3 \quad \mu_3^*(H) = HP}$

$x_3 = I \rightarrow$ $\boxed{J_3^*(I) = \min_{\mu} \{ \underbrace{l_3(I, IP)}_{4} + \underbrace{J_4(P)} \} = 4 \quad \mu_3^*(I) = IP}$

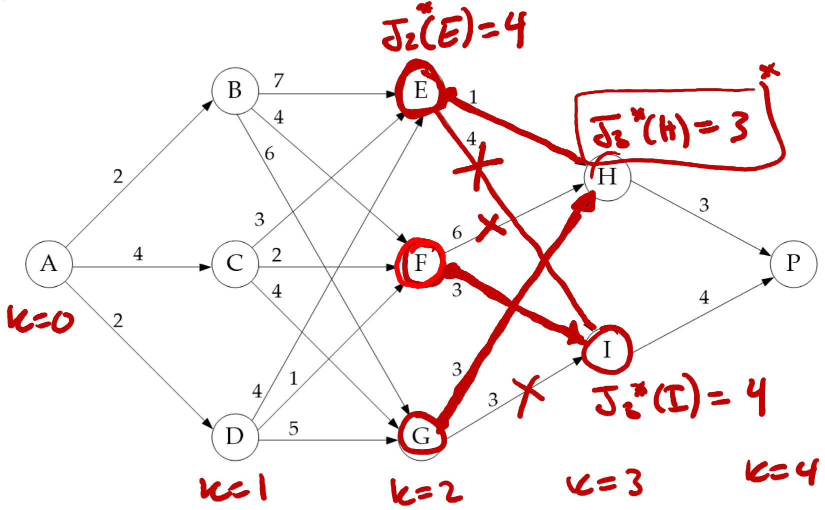


$\pi_c = \{AC, CE, EH, HP\}$

$J_{\pi_c} = 4+3+1+3$
$= 11$

Backwards Dynamic Programming – Shortest path Example

- Consider the problem of traveling from city *A* to city *P*, use Dynamic Programming to find the shortest path!



$k=2$

$x_2=E$

$$J_2^*(E)=\min_{\mu}\{l_2(E,\mu)+J_3(x_3)\}$$

$$=\min_{\mu}\{\underbrace{l_2(E,EH)}_{1}+\underbrace{J_3(H)}_{3},\ \underbrace{l_2(E,EI)}_{4}+\underbrace{J_3(I)}_{4}\}=\min_{\mu}\{\overset{\checkmark}{\underset{\downarrow\ \mu=EH}{4}},8\}=\underset{*}{\boxed{4}}\qquad \boxed{\mu_2^*(E)=EH}$$

$\underline{J_2^*(F)}=\underline{7}$ $\qquad\mu_2^*(F)=FI$

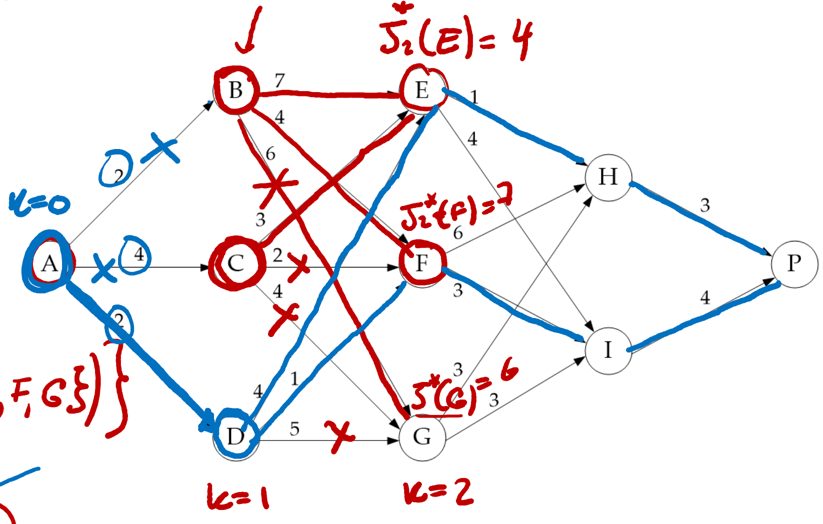
$$J_2^*(F)=\min_{\mu}\{\underbrace{l_2(F,FH)}_{6}+\underbrace{\underline{J_3(H)}}_{3},\ \underbrace{l_2(F,FI)}_{3}+\underbrace{\underline{J_3(I)}}_{4}\}=7\qquad \mu_2^*(F)=FI$$

$J_2^*(G)=\underline{6}$ $\qquad\mu_2^*(G)=GH$

$$J_2^*(G)=\min_{\mu}\{l_2(G,\{GH,GI\})+J_2(\{H,I\})\}=\min\{\underbrace{3+3}_{GH},\ \underbrace{3+4}_{GI}\}=6\qquad \mu_2^*(G)=GH$$

Backwards Dynamic Programming – Shortest path Example

- Consider the problem of traveling from city *A* to city *P*, use Dynamic Programming to find the shortest path!



k=1

$(x_1 = B) \Rightarrow J_1^*(B) = \min_\mu \{ l_1(B, \{BE, BF, BG\}) + J_2(\{E, F, G\}) \}$

$= \min_\mu \{ \underbrace{7 + \overset{J_2^*(E)}{4}}_{11}, \underbrace{4+7}_{11}, \underbrace{6+6}_{12} \} = \boxed{\underline{11}}$

B→E　B→F　B→G

$\mu_1(B) = \{BE, BF\}$

$x_1 = C \Rightarrow J_1^*(C) = \min_\mu \{ l_1(C, \{CE, CF, CG\}) + J_2(\{E, F, G\}) \} = \min \{ \underbrace{\boxed{3+4}}_{7}, \underbrace{2+7}_{9}, \underbrace{4+6}_{10} \}$

$\boxed{J_1^*(C) = 7}$　$\mu_1^*(C) = CE$

$x_1 = D$　$J_1^*(D) = \min_\mu \{ l_1(D, \{DE, DF, DG\}) + J_2(E, F, G) \} = \min_\mu \{ \underbrace{4+4}_{8}, \underbrace{1+7}_{8}, \underbrace{5+6}_{11} \} = 8$

$\boxed{J_1^*(D) = 8}$　$\mu_1^*(D) = \{DE, DF\}$

k=0

$x_0 = A$　$J_0^*(A) = \min_\mu \{ l_0(A, \{AB, AC, AD\}) + J_1(\{B, C, D\}) \} = \min \{ \underbrace{\overset{13}{2+11}}_{A\to B}, \underbrace{\overset{11}{4+7}}_{C}, \underbrace{\overset{D}{2+8}}_{10} \} = 10$

$J_0^*(A) = 10$　$\mu_0(A) = AD$

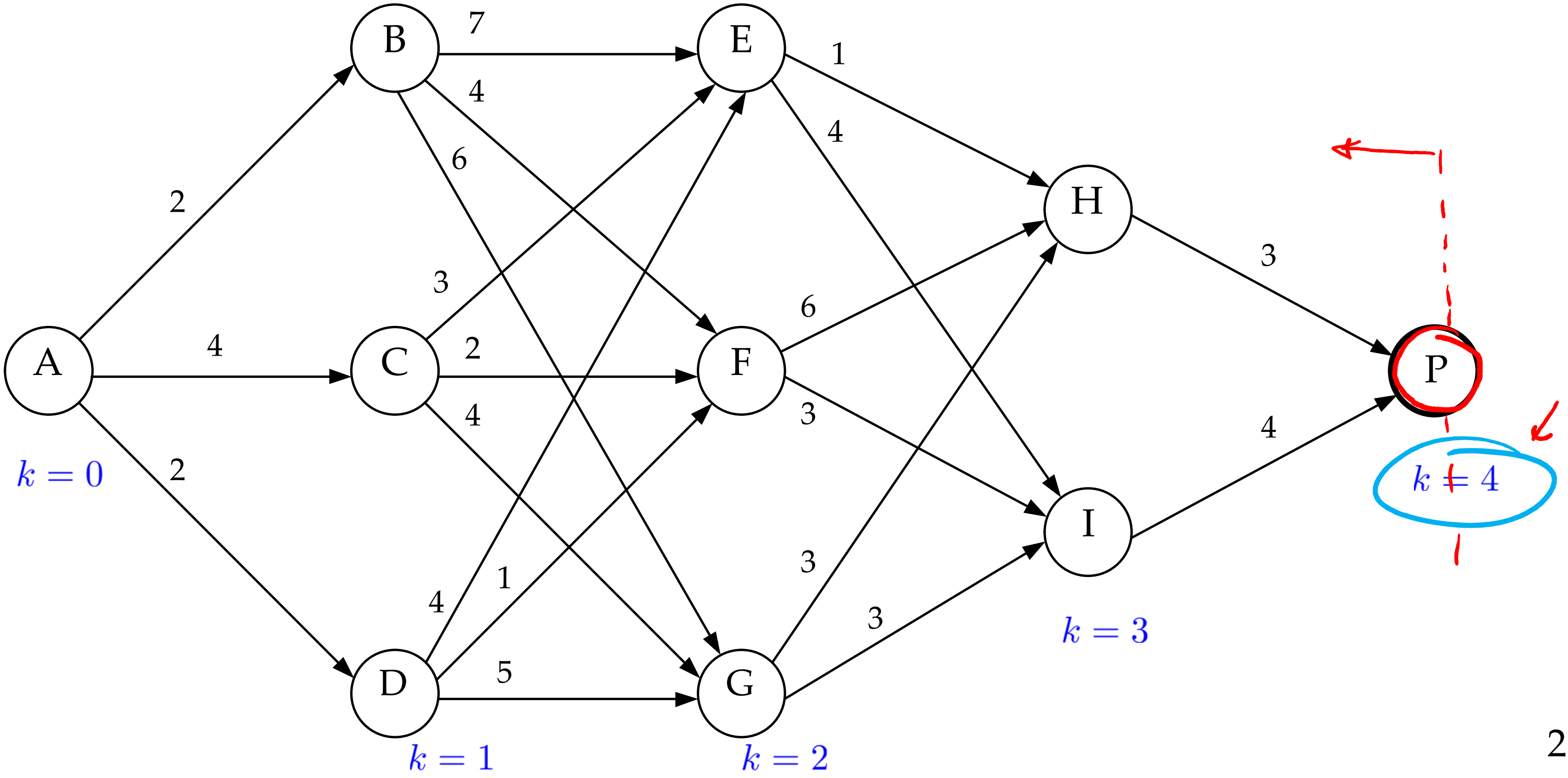
Backwards Dynamic Programming – Shortest path Example

- Dynamic Programming summary for shortest path problem!

Step 1: compute final cost

$$J_N(x_4) = g_4(x_4) = \begin{cases} 0 & \text{if } x_4 = P \\ \infty & \text{o.w.} \end{cases}$$

N=4



Backwards Dynamic Programming – Shortest path Example

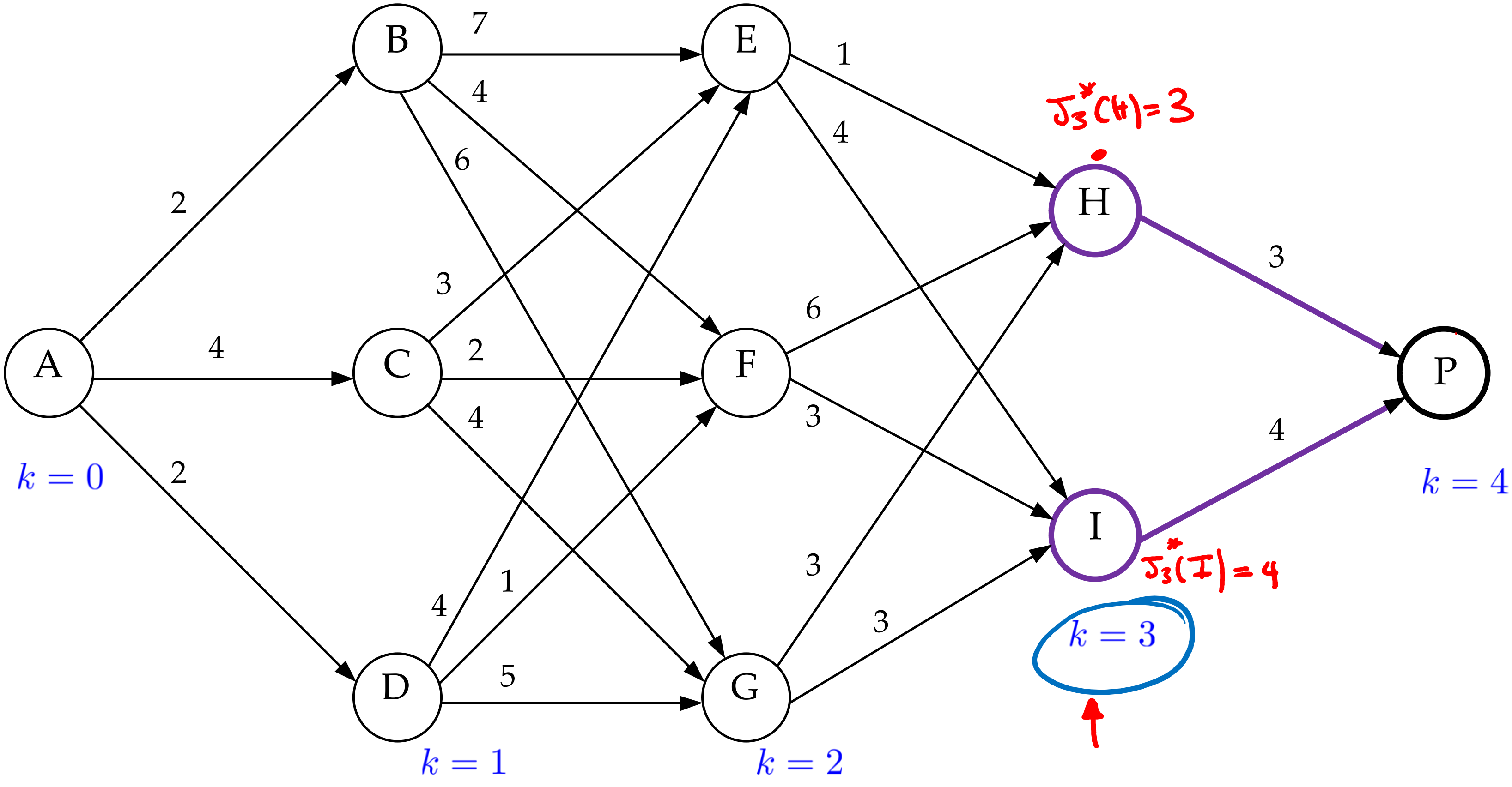
- Dynamic Programming summary for shortest path problem!

Step 2: Go backwards one time, $k = 3$

$$J_3(H) = \min_{\mu} \{\underbrace{l_3(H, HP)}_{\text{running cost}} + \overbrace{J_4(P)}^{\text{optimal cost @ P}}\} = 3 \quad \mu_3^*(H) = HP$$

$$J_3(H) = \min_{\mu} \{ \underset{l_3}{3} + \overset{J_4(P)}{0} \}$$

$$J_3(I) = \min_{\mu} \{l_3(I, IP) + J_4(P)\} = 4 \quad \mu_3^*(I) = IP$$



Backwards Dynamic Programming – Shortest path Example

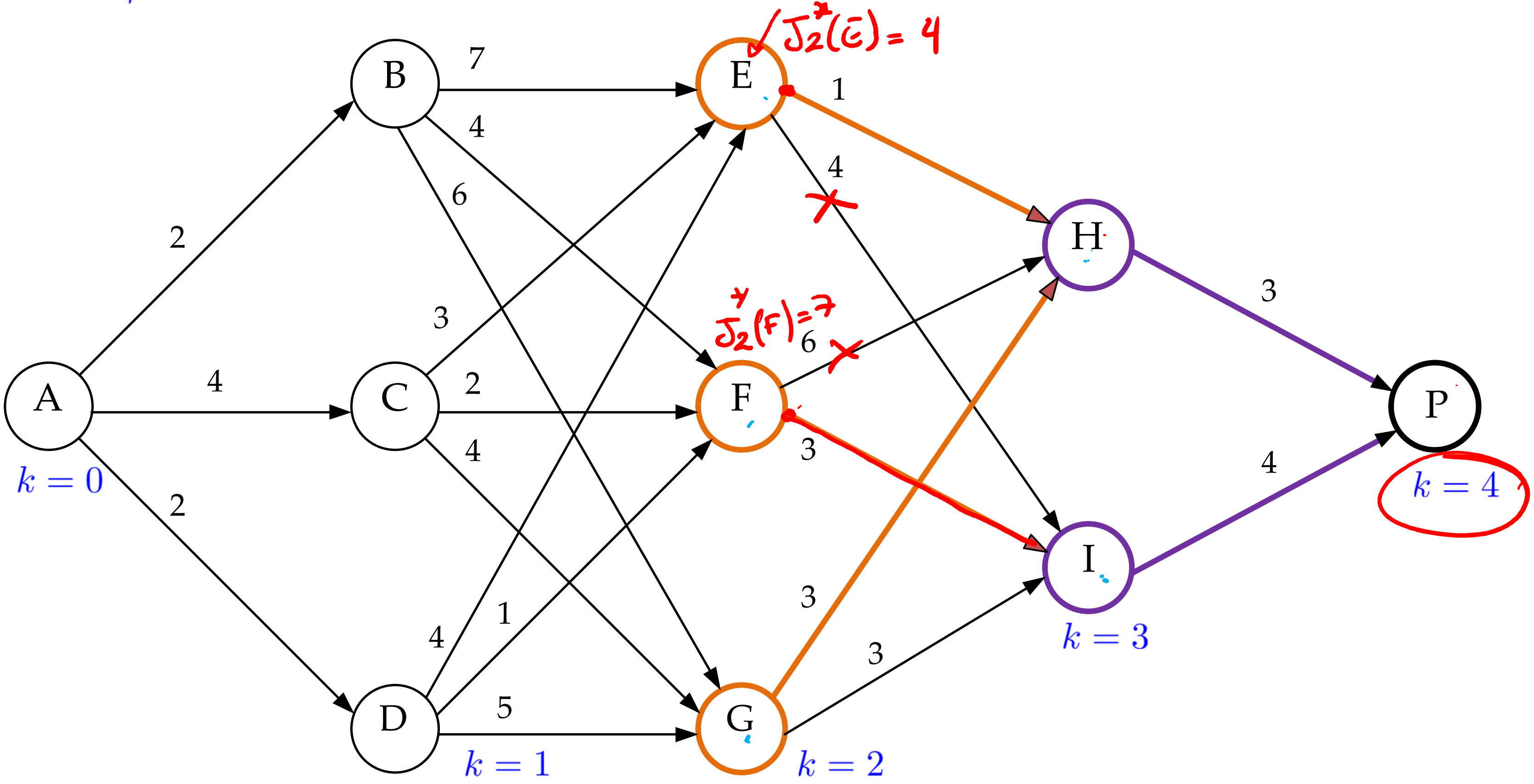
- Dynamic Programming summary for shortest path problem!

Step 2: Go backwards one time, $k = 2$

→ $J_2(E) = \min_{\mu} \{l_2(E, \mu_2(E)) + J_3^*(\underline{H},\ I)\} = \underline{\underline{4}}$ $\mu_2^*(E) = \underline{EH}$

$J_2(F) = \min_{\mu} \{l_2(F, \mu_2(F)) + J_3^*(H,\ I)\} = 7$ $\mu_2^*(F) = FI$

$J_2(G) = \min_{\mu} \{l_2(G, \mu_2(G)) + J_3^*(H,\ I)\} = 6$ $\mu_2^*(F) = GH$



Backwards Dynamic Programming – Shortest path Example

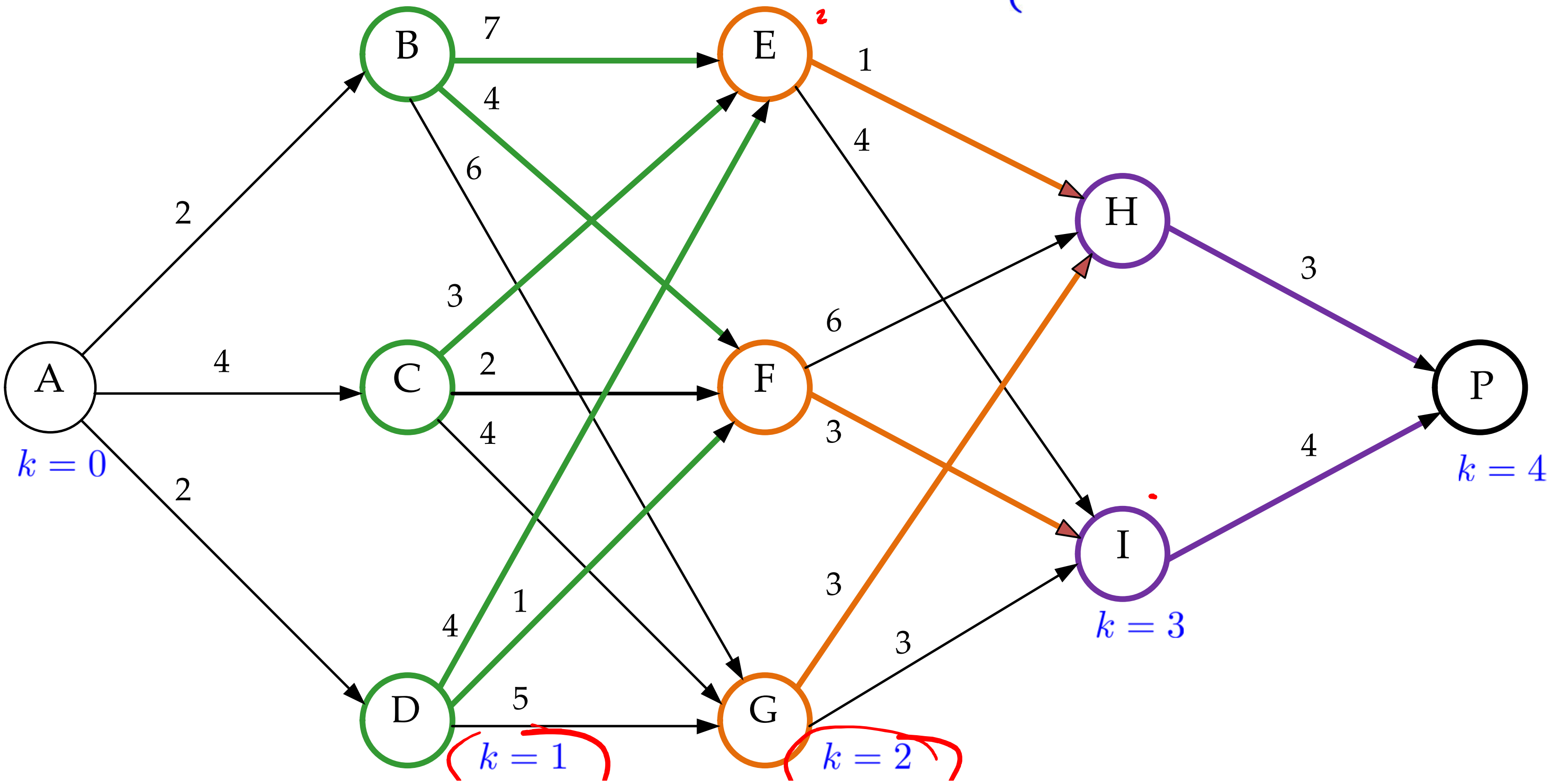
- Dynamic Programming summary for shortest path problem!

Step 2: Go backwards one time, $k = 1$

$$\underline{J_1(B)} = \min_{\mu} \{l_1(B, \mu_1(B)) + \underline{J_2}(E,\ F,\ G)\} = 11 \qquad \mu_1^*(B) = \begin{cases} BE \\ BF \end{cases}$$

$$J_1(C) = \min_{\mu} \{l_1(C, \mu_1(C)) + \underline{J_2}(E,\ F,\ G)\} = 7 \qquad \mu_1^*(C) = CE$$

$$J_1(D) = \min_{\mu} \{l_1(D, \mu_1(D)) + \underline{J_2}(E,\ F,\ G)\} = 8 \qquad \mu_1^*(D) = \begin{cases} DE \\ DF \end{cases}$$

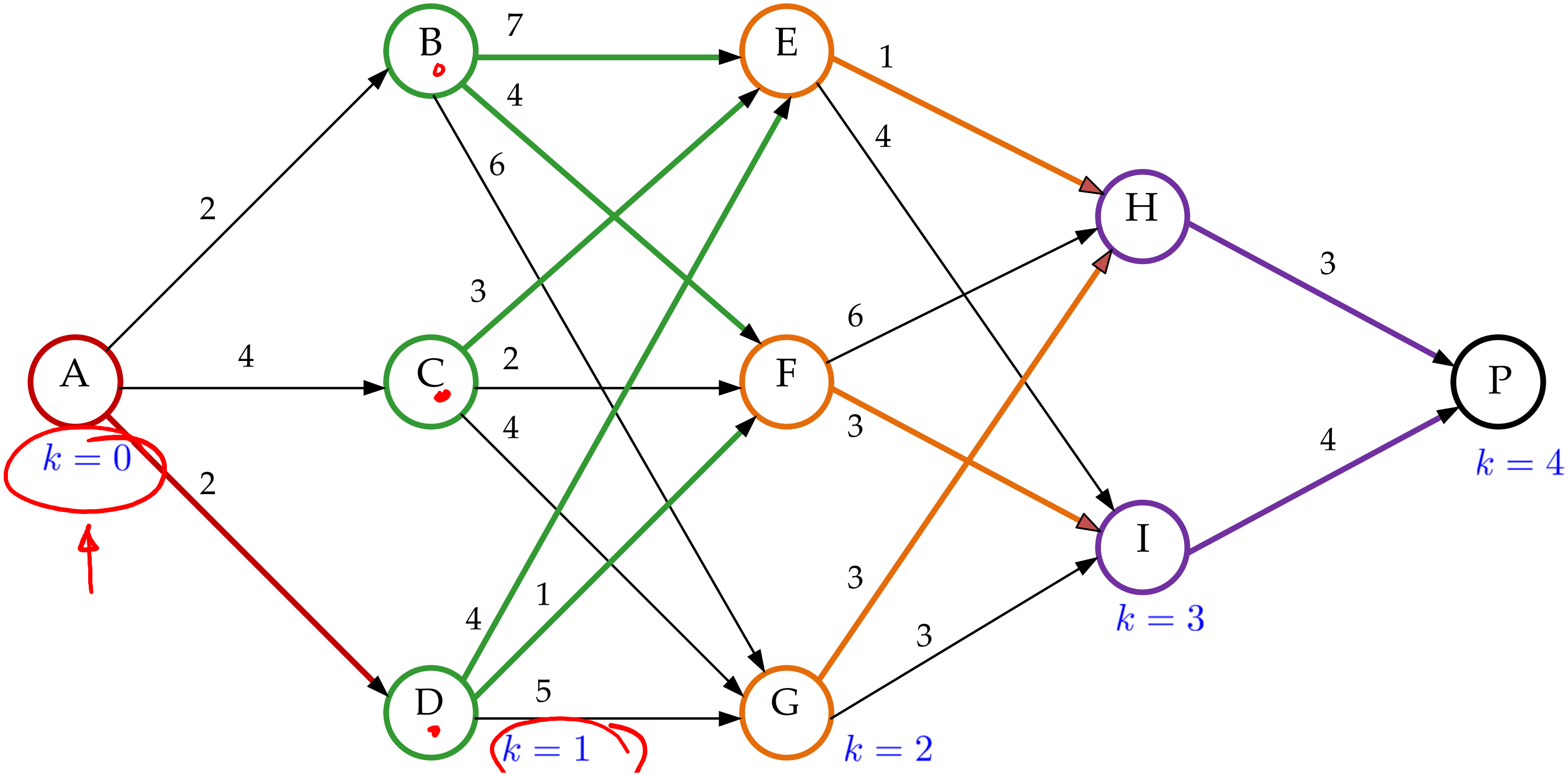


Backwards Dynamic Programming – Shortest path Example

- Dynamic Programming summary for shortest path problem!

Step 3: Go backwards one time, $k = 0$

$$J_0(A) = \min_{\mu} \{l_0(A, \mu_0(A)) + J_1(B,\ C,\ D)\} = 10 \qquad \mu_0^*(A) = AD$$



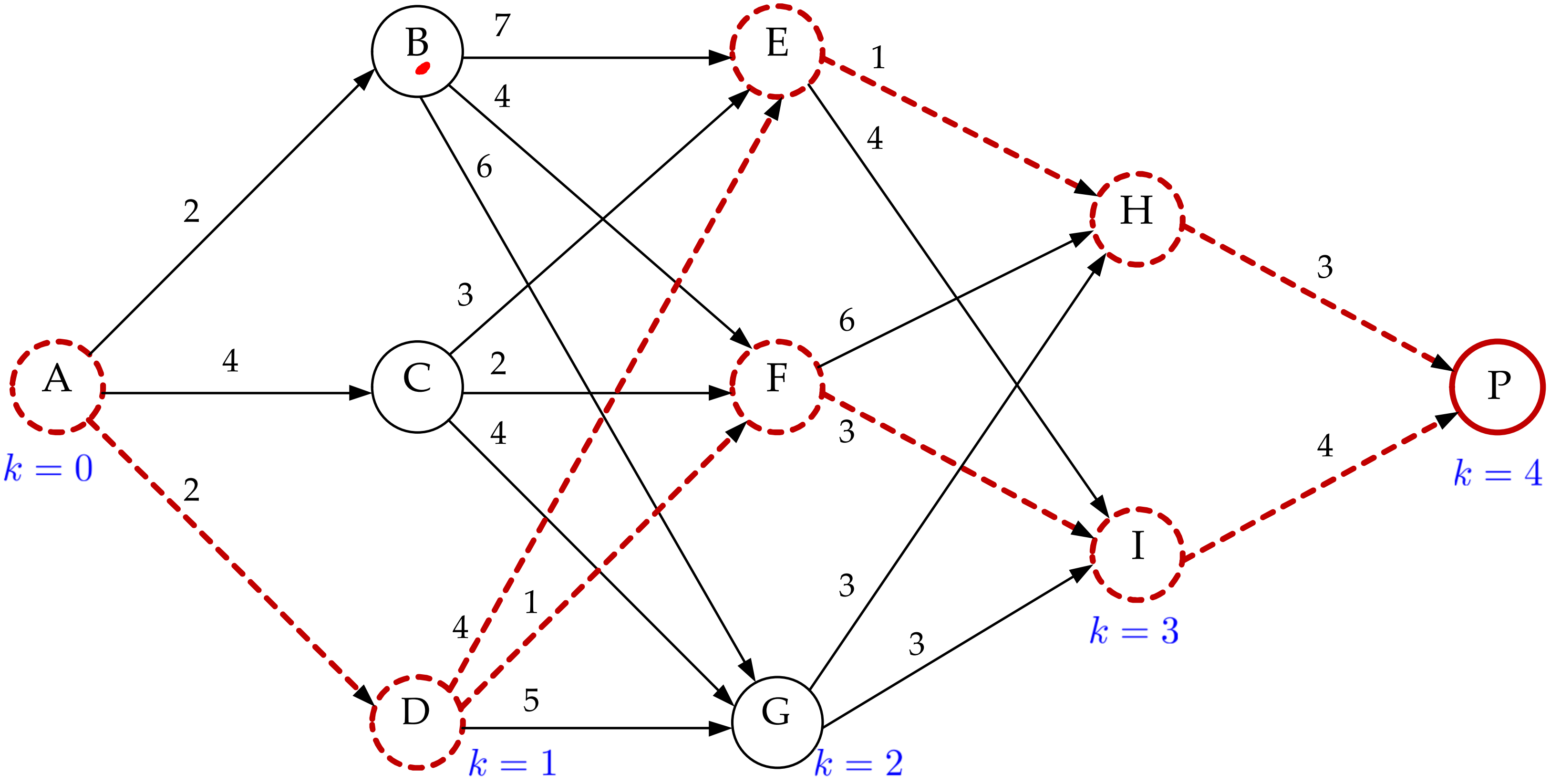
Backwards Dynamic Programming – Shortest path Example

- Dynamic Programming summary for shortest path problem!

Step 3: Go to initial point, $k = 0$ and reconstruct optimal path

$$J_0(A) = \min_{\mu} \{l_0(A, \mu_0(A)) + J_1(B,\ C,\ D)\} = 10 \qquad \mu_0^*(A) = AD$$

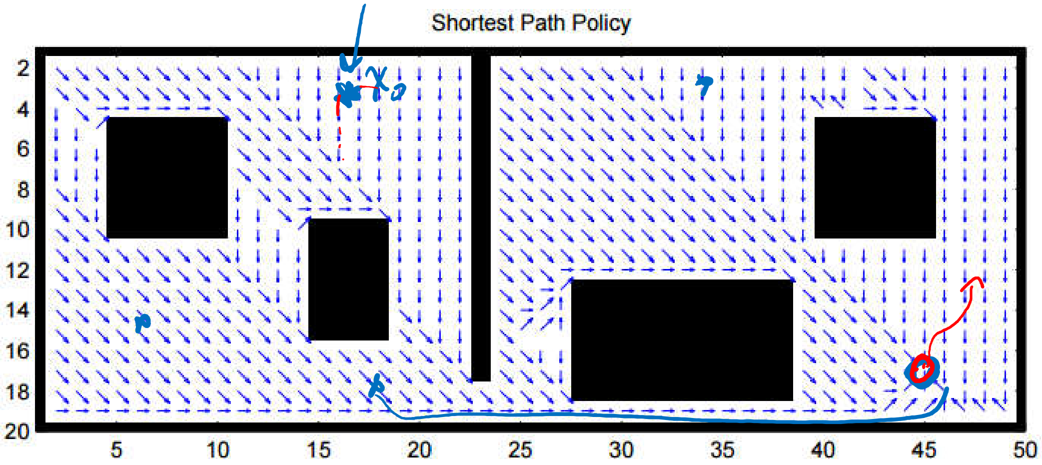
$$\pi^* = \{\mu_0^*(A),\ \mu_1^*(D),\ \mu_2^*,\ \mu_3^*\} = \begin{cases} \{AD,\ DE,\ EH,\ HP\} \\ \{AD,\ DF,\ FI,\ IP\} \end{cases} \qquad J_{\pi^*}(A) = \underline{10}$$

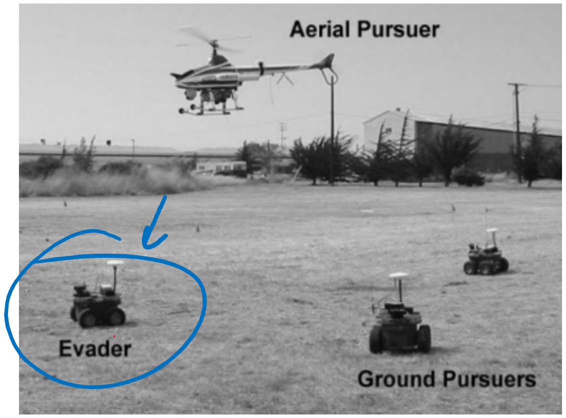


Applications of Dynamic Programming

- Power systems – unit commitment, optimization* (will study in detail)

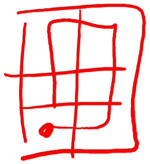
- **Route planning**
  - Minimize time – "shortest path problems"
  - Pursuit-evasion games (UAVs), optimal control theory

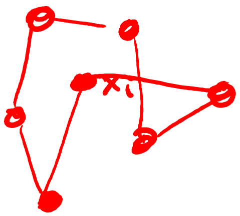




- *Communication*, **pattern recognition** → Viterbi Algorithm
- ***Optimal control** (finding optimal paths)

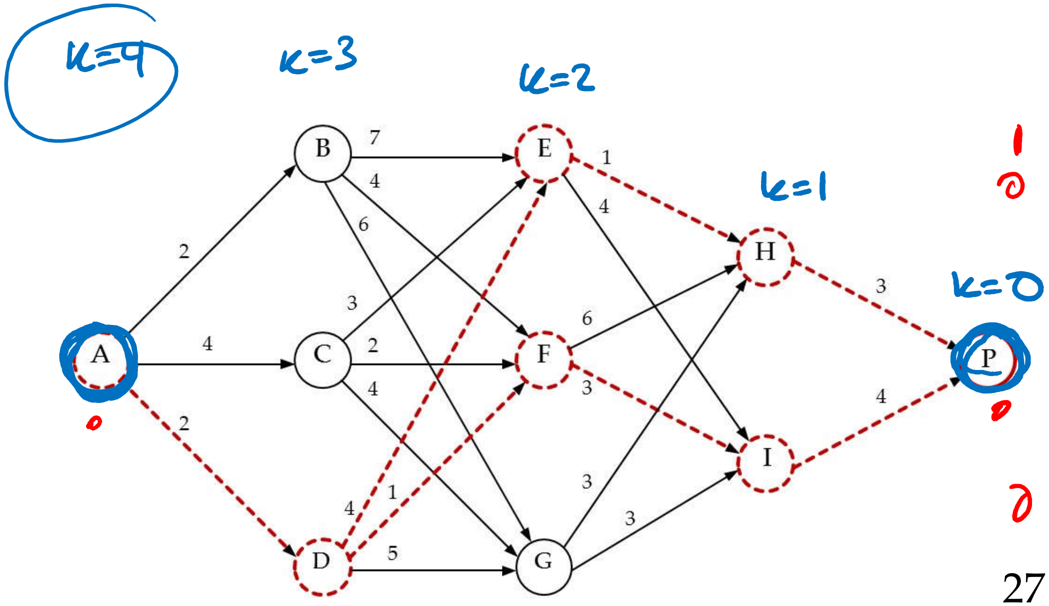
• Traveling Salesman





Forward Dynamic Programming

- **Principle/idea**
  if a path from *A-to-P* is **optimal,** then going from *P-to-A* through the **same path is also optimal**
  - We can think of this problem then as starting at *P* and going to *A*
  - Use regular DP algorithm for this problem
- Both Forwards or Backwards DP will yield the same optimal policy

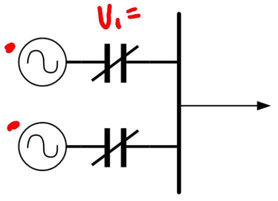


Unit Commitment Example

- Consider, a unit commitment problem for 3 hours
- We have start-up costs only (assume shut-down cost is 0) * (not always case)
- * Assume, both **generators are initially on (important)** → initial condition

Gen.	$P_{\min}$	$P_{\max}$	Cost/power	Start-up cost
1	1.5	7	7.2	4
2	1	6	7.8	4

Hour	1	2	3
P_L	5.5	8	4

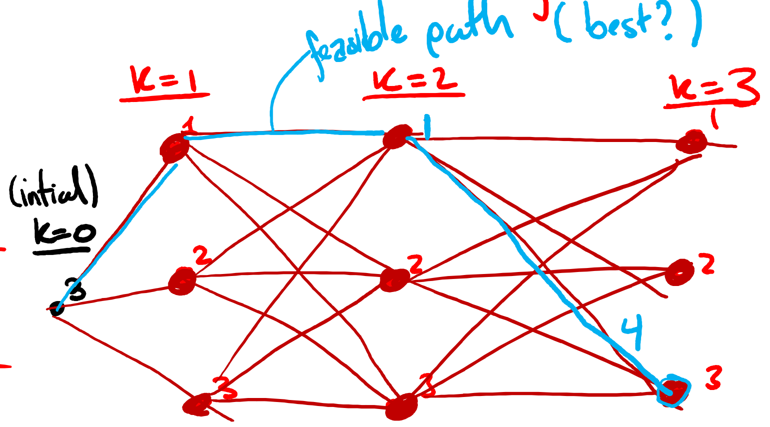


- **Set up the DP algorithm**
- How many modes (states) are there? (ignore the mode where both generators are off)

How many combinations of generator status are there at every hour?

$$\left(2^{\overset{\#\text{gens}}{2}}-1\right)^{\overset{\#\text{hours}}{3}} = 27$$

State	Gen 1-status	Gen 2-status
1	1	0
2	0	1
3	1	1



- These can be considered the "states" or x $x_k \in \{1,2,3\}$ $\forall k \geq 1$

Unit Commitment Example

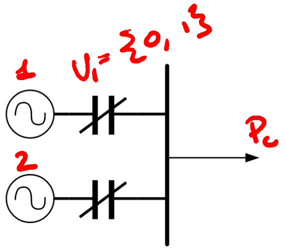
- Consider, a unit commitment problem for 3 hours
- Initial mode is 3 **(both generators are on)**

$x_k = \{1, 2, 3\}$

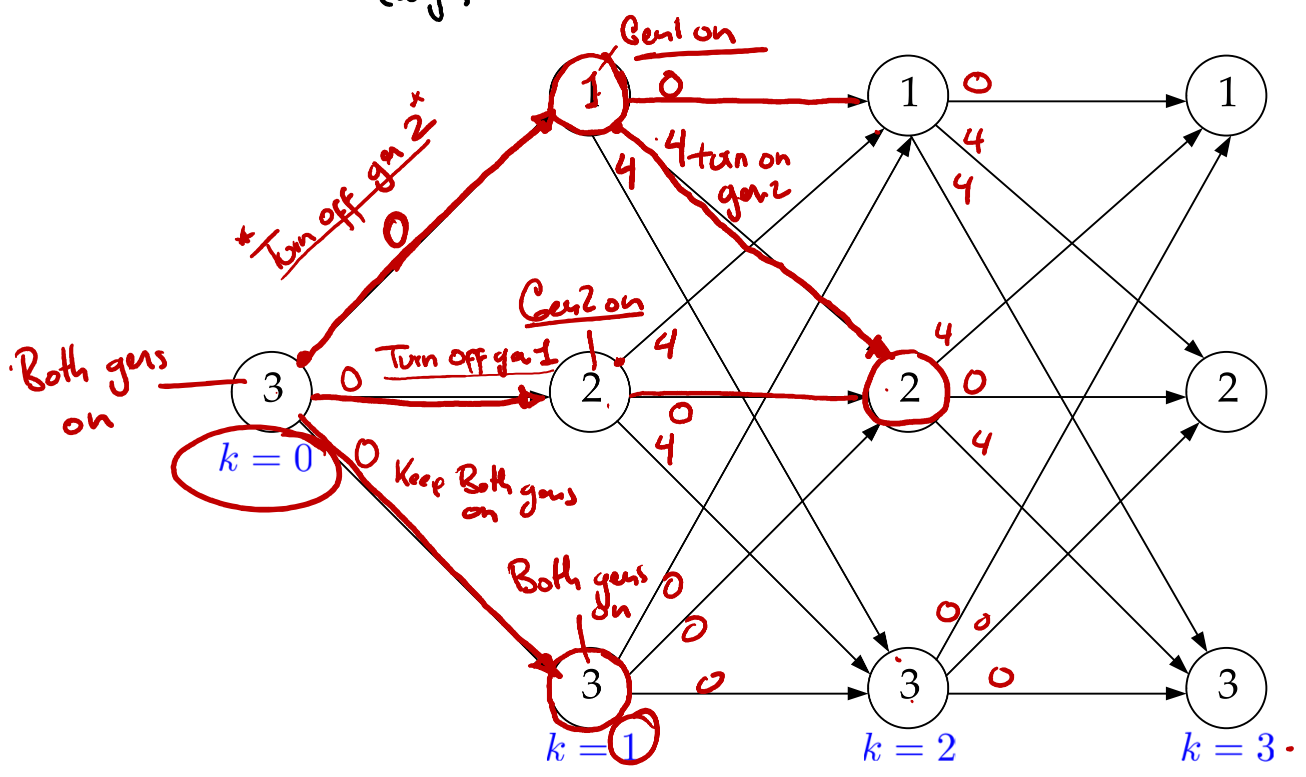
$U_i = \{0, 1\}$

Gen.	$P_{\min}$	$P_{\max}$	Cost/power	* Start-up cost *
1	1.5	7	7.2	4
2	1	6	7.8	4

Hour	1	2	3
P_L	5.5	8	4



- **What are the transition costs?** (Not running cost)
(edge)



Unit Commitment Example

Gen.	$P_{\min}$	$P_{\max}$	Cost/power	Start-up cost
1	1.5	7	7.2	4
2	1	6	7.8	4

Hour	1	2	3
P_L	5.5	8	4

- Define costs functions (final, middle, initial)

$$J_N(x_N) = \underline{g_N(x_N)}$$

$$* \; J_k(x_k) = \min_{\mu} \{ \underbrace{l_k(x_k, \mu(x_k))}_{\text{running cost}} + J_{k+1}(x_{k+1}) \}$$

$$\forall k = 0, \; ..., \; N-1$$

- * Define $\underset{\text{hour } k = 1, 2, 3}{g_k^{\text{hour}}}(\overset{\text{mode/state at } k^{\text{th}} \text{ hour}}{x_k}) = \min_{\{P_1, P_2\}} \text{Total cost } \{P_1, P_2\}$ s.t. $\{P_1, P_2, P_1 + P_2\} = P_L(k)$, $1.5 \leq P_1 \leq 7$, $1 \leq P_2 \leq 6$

(Economic Dispatch problem)

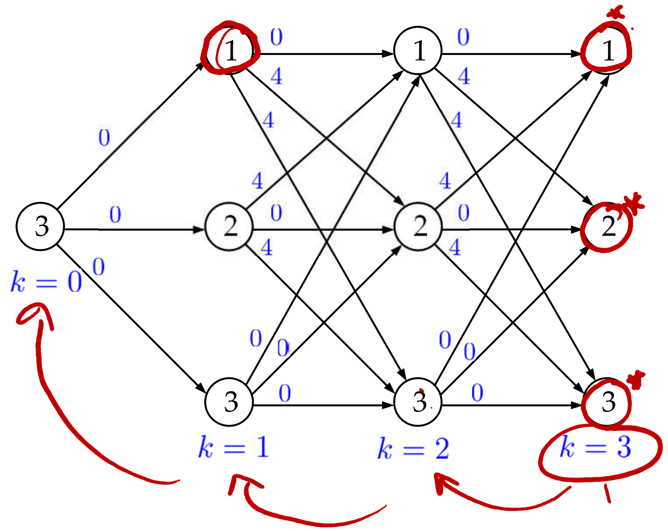
- Final cost $J_N(x_N) = J_3(x_3) = \underline{g_3(x_3)}$

- $J_k(x_k) = \min_{\mu} \{ \underbrace{l_k(x_k, \mu_k(x_k))}_{} + J_{k+1}(x_{k+1}) \}$

- $\underbrace{l_k(x_k, \mu_k(x_k))}_{\text{* running cost *}} = \underbrace{g_k(x_k)}_{\text{Econ. Dispatch *}} + \underbrace{a_{x_k - x_{k+1}}}_{\text{transition cost (prev slide)}}$

↓

startup/shutdown cost



Unit Commitment Example

Gen.	$P_{\min}$	$P_{\max}$	Cost/power	Start-up cost
1	1.5	7	7.2	4
2	1	6	* 7.8 *	4

Hour	1	2	3
P_L	5.5	8	4

$$G = * \begin{pmatrix} 39.6 & \infty & 28.8 \\ 42.9 & \infty & 31.2 \\ 40.2 & 58.2 & 29.4 \end{pmatrix}$$

- Compute the costs at k=3 $X_3 = \{1, 2, 3\}$

$$* \boxed{J_3(x_3) = g_3(x_3)} *$$

$$J_k(x_k) = \min_{\mu} \{g_k(x_k) + a_{x_k \to j} + J_{k+1}(j)\}$$

$$\forall k = 1, \ldots, 2$$

$$\underbrace{J_0(x_0)}_{\text{initial cost}} = \min_{\mu} \{0 + a_{x_0 \to j} + J_1(j)\}$$

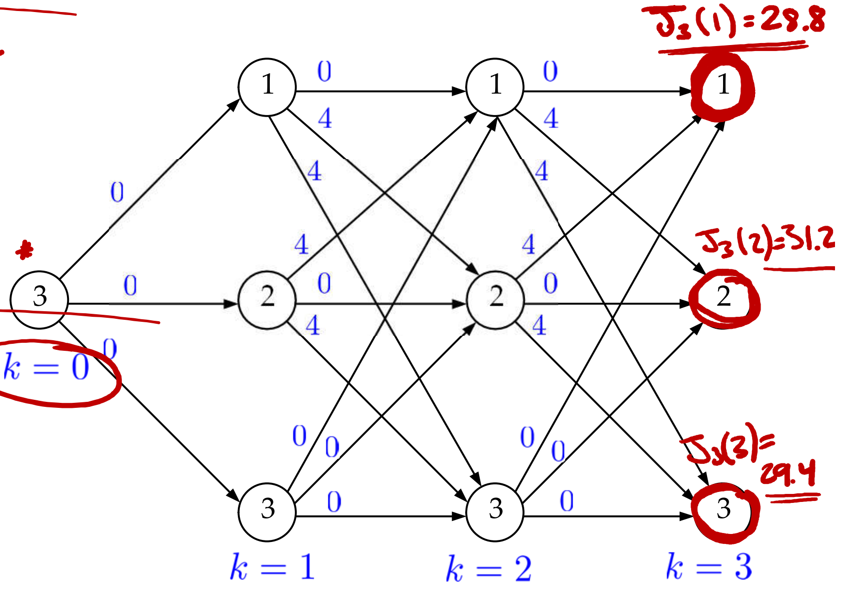
$$x_3 = 1 \Rightarrow J_3(1) = \underbrace{g_3(1)}_{\text{gen 1 on}} = \underbrace{\boxed{\begin{array}{l} \min_{P_1} \ 7.2P_1 \\ \text{s.t.} \\ P_1 = 4 \\ 1.5 \le P_1 \le 7 \end{array}}}_{\boxed{P_1^* = 4}} = \underbrace{(7.2)(4) = 28.8}$$

$$x_3 = 2 \Rightarrow J_3(2) = g_3(2) = \underbrace{\begin{array}{l} \min_{P_2} \ 7.8P_2 \\ \text{s.t.} \\ P_2 = 4 \\ 1 \le P_2 \le 6 \end{array}}_{P_2^* = 4} = (7.8)(4) = 31.2$$

(Gen 2 on)

$$x_3 = 3 \Rightarrow J_3(3) = g_3(3) = \underbrace{\begin{array}{l} \min_{P_1, P_2} (7.2\underset{3}{P_1} + 7.8\underset{1}{P_2}) = 29.4 \\ 1.5 \le P_1 \le 7 \\ 1 \le P_2 \le 6 \\ P_1 + P_2 = 4 \end{array}}_{\text{cvx} \ \boxed{\begin{array}{l} P_1^* = 3 \\ P_2^* = 1 \end{array}}}$$

Both gens on



Unit Commitment Example

Gen.	$P_{\min}$	$P_{\max}$	Cost/power	Start-up cost
→ 1	1.5	$\boxed{7}$	7.2	4
→ 2	1	$\boxed{6}$	7.8	4

Hour	1	2	3
P_L	5.5	8	4

$$G = \begin{array}{cc} & \begin{array}{ccc} k=1 & \underline{2} & 3 \end{array} \\ \begin{array}{c} \text{Gen 1 } \$ \\ \text{Gen 2 } \$ \\ \text{Gen 1 and 2 } \$ \end{array} & \begin{pmatrix} \underline{39.6} & \boxed{\infty} & 28.8 \\ \underline{42.9} & \boxed{\infty} & 31.2 \\ 40.2 & \underline{58.2} & 29.4 \end{pmatrix} \end{array}$$

- Compute the costs at $k=\underline{2}$ → $x_2=\{1,2,3\}$

$x_2=1 \Rightarrow J_2(1) = \min\limits_{\underline{\mu}} \{ \underbrace{\overset{\infty}{g_2(1)}}_{*} + \underset{\downarrow \; 0,4,4}{a_{1\to\{1,2,3\}}} + \underline{J_3(\{1,2,3\})} \}$

Gen 1 on

$$g_2(\underline{1}) = \underbrace{\left[\begin{array}{l} \min\limits_{P_1} \; 7.2 P_1 \\ \text{s.t.} \\ \quad \boxed{P_1 = 8} \\ \quad 1.5 \le P_1 \le 7 \end{array}\right]}_{P_1^* = \text{No solution / infeasible}} = \boxed{\infty}$$

$x_2=2 \Rightarrow J_2(2) = \min\limits_{\mu} \{ \underline{g_2(2)} + a_{2\to\{1,2,3\}} + J_3(\{1,2,3\}) \}$

Gen 2 on

$\to \min 7.8 P_2 = \boxed{\infty}$

s.t.

$P_2 = 8$

$1 \le P_2 \le 6$

$x_2=3 \Rightarrow J_2(3) = \min\limits_{\mu} \{ \underline{g_2(3)} + \underline{a_{2\to\{1,2,3\}}} + \underline{J_3(\{1,2,3\})} \}$

Both gens on

$g_2(3) = \underline{58.2}$

$$\left\{ \begin{array}{l} \min\limits_{P_1,P_2} \; \underline{7.2P_1 + 7.8P_2}^{\,*} \\ \text{s.t.} \\ \quad P_1 + P_2 = 8 \\ \quad P_1 \in [1.5, 7] \\ \quad P_2 \in [1, 6] \end{array} \right. \quad \boxed{\begin{array}{l} P_1^* = 7 \\ P_2^* = 1 \end{array}}$$

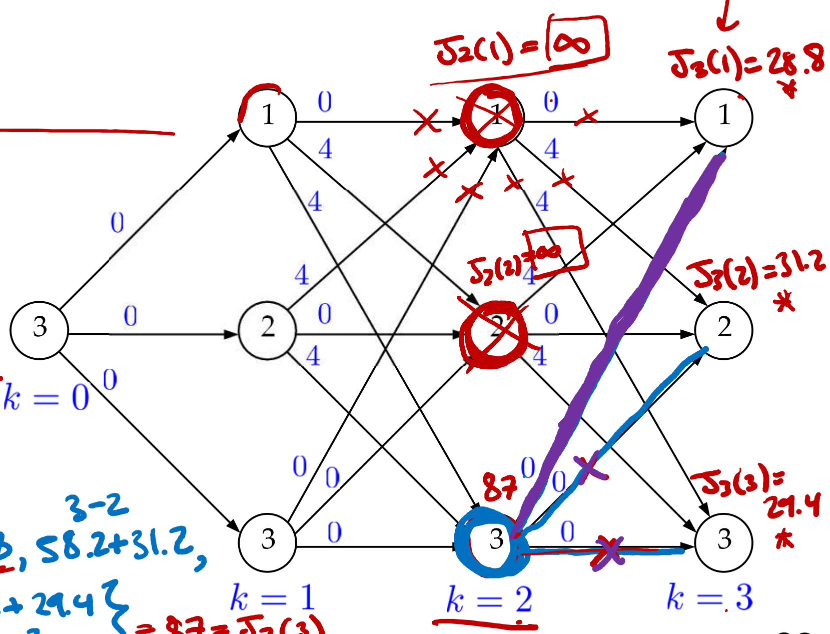
$$\min_{\mu} \{ \underline{\overset{3-1}{58.2+28.8}}, \overset{3-2}{58.2+31.2}, \underset{3-3}{58.2+29.4} \} = \underline{87} = J_2(3)$$

$$J_3(x_3) = (g_3(x_3))$$

$$J_k(x_k) = \min_{\mu} \{ (g_k(x_k) + a_{x_k\to j}) + J_{k+1}(j) \}$$

$$\forall k = 1, \ldots, 2$$

$$J_0(x_0) = \min_{\mu} \{ 0 + a_{x_0\to j} + J_1(j) \}$$



Unit Commitment Example

Gen.	$P_{\min}$	$P_{\max}$	Cost/power	Start-up cost
1	1.5	7	7.2	4
2	1	6	7.8	4

Hour	1	2	3
P_L	5.5	8	4

$$G = \begin{pmatrix} 39.6 & \infty & 28.8 \\ 42.9 & \infty & 31.2 \\ 40.2 & 58.2 & 29.4 \end{pmatrix}$$

- Compute the costs at *k=1*

$x_2 = \{1, 2, 3\}$

state @ hour 1 ⇒ gen 1 is on

Hour

$$J_1(1) = \min_{\mu} \{ \underbrace{g_1(1)}_{} + \underbrace{a_{1-x_2}}_{\text{transition cost}} + J_2(x_2) \}$$

$$\left(\min_{P_1} 7.2P_1 \text{ s.t. } P_1 = 5.5,\ 1.5 \le P_1 \le 7 \right)$$

$P_1^* = 5.5$

$(7.2)5.5 = 39.6$

$$= \{ 39.6 + 0 + \overset{\infty}{J_2(1)},\ 39.6 + 4 + J_2(2) = \infty,\ 39.6 + 4 + \underset{87}{J_2(3)} \}$$

$$\boxed{J_1^*(1) = 130.6} \qquad \mu_1^*(1) = 1-3$$

$$J_1(2) = \min_{\mu} \{ g_1(2) + a_{2\to x_2} + J_2(x_2) \}$$

$$J_1^*(2) = 42.9 + \underset{2-3}{4} + \underset{87}{J_2(3)} = 133.9 \qquad \mu_1^*(2) = 2-3$$

$$* J_1(3) = \min_{\mu} \{ g_1(3) + a_{3-x_2} + J_2(x_2) \}$$

$$\min_{P_1, P_2} 7.2P_1 + 7.8P_2 \text{ s.t. } P_1 + P_2 = 5.5$$

$P_1 \in [1.5, 7],\ P_2 \in [1, 6]$

$P_1^* = 4.5$

$P_2^* = 1 \Rightarrow 40.2$

$$J_1^*(3) = \{ 40.2 + 0 + 87 \}$$

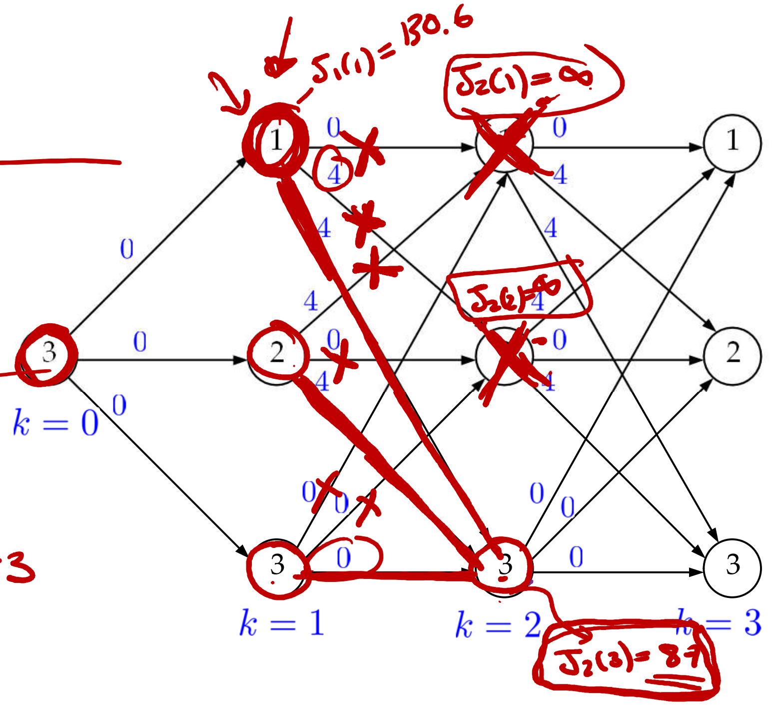
$J_1^{(3)} = 127.2 \qquad \mu_1^*(3) = 3-3$

$$J_3(x_3) = g_3(x_3)$$

$$* J_k(x_k) = \min_{\mu} \{ g_k(x_k) + a_{x_k \to j} + J_{k+1}(j) \}$$

$$\forall k = 1,\ \ldots,\ 2$$

$$J_0(x_0) = \min_{\mu} \{ 0 + a_{x_0 \to j} + J_1(j) \}$$



Unit Commitment Example

Gen.	$P_{\min}$	$P_{\max}$	Cost/power	Start-up cost
1	1.5	7	7.2	4
2	1	6	7.8	4

0	Hour	1	2	3
0	P_L	5.5	8	$\underline{4}$

$$G = \begin{pmatrix} 39.6 & \infty & 28.8 \\ 42.9 & \infty & 31.2 \\ 40.2 & 58.2 & 29.4 \end{pmatrix}$$

- Compute the costs at *k=0*
- Reconstruct Policy

$$J_3(x_3) = g_3(x_3)$$

$$J_k(x_k) = \min_\mu \{g_k(x_k) + a_{x_k \to j} + J_{k+1}(j)\}$$

$$\forall k = 1, \ldots, 2$$

→ $$J_0(x_0) = \min_\mu \{0 + a_{x_0 \to j} + J_1(j)\}$$

$$\boxed{J_0(3)} = \min_\mu \{ \underset{0}{\underset{\parallel}{\cancel{g_0(3)}}} + \underline{a_{3\to x_1}} + \underline{J_1(x_1)} \}$$

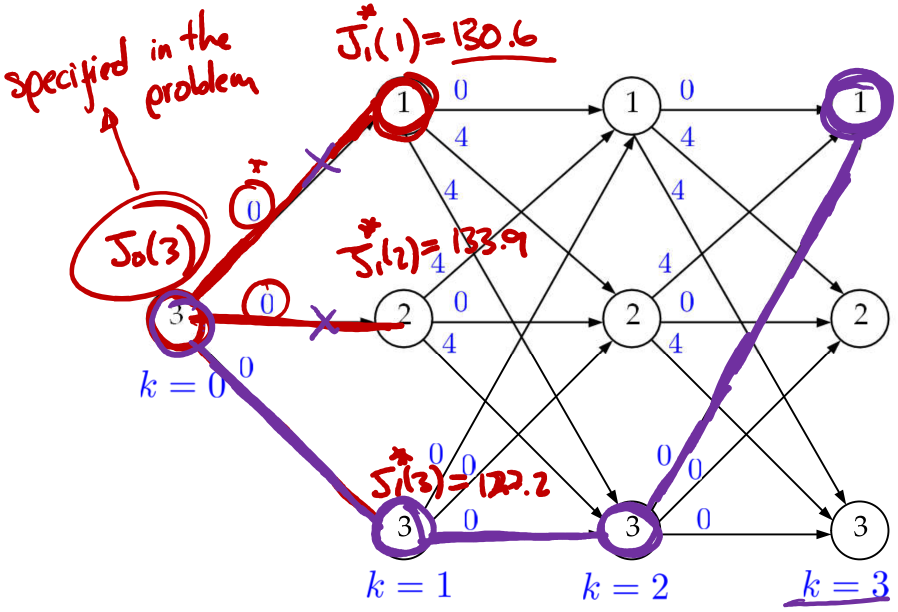
$$= \min_\mu \{ \underline{\overset{3-1}{0 + 0 + 130.6}}, \underline{\overset{3-2}{0 + 0 + 133.9}}, \underline{\overset{3-3}{0 + 0 + 127.2}} \}$$

$$\boxed{J_0^*(3) = 127.2} \quad \mu_0^*(3) = 3-3$$

Transition cost may not always be 0 in this case

e.g. - if we have shutdown cost

- if we have different initial condition



Unit Commitment Example

Gen.	$P_{\min}$	$P_{\max}$	Cost/power	Start-up cost
1	1.5	7	7.2	4
2	1	6	7.8	4

Hour	1	2	3
P_L	5.5	8	4

$$G = \begin{pmatrix} 39.6 & \infty & 28.8 \\ 42.9 & \infty & 31.2 \\ 40.2 & 58.2 & 29.4 \end{pmatrix}$$

$$G = \begin{pmatrix} g_1(1) & g_2(1) & g_3(1) \\ g_1(2) & g_2(2) & g_3(2) \\ g_1(3) & g_2(3) & g_3(3) \end{pmatrix}$$

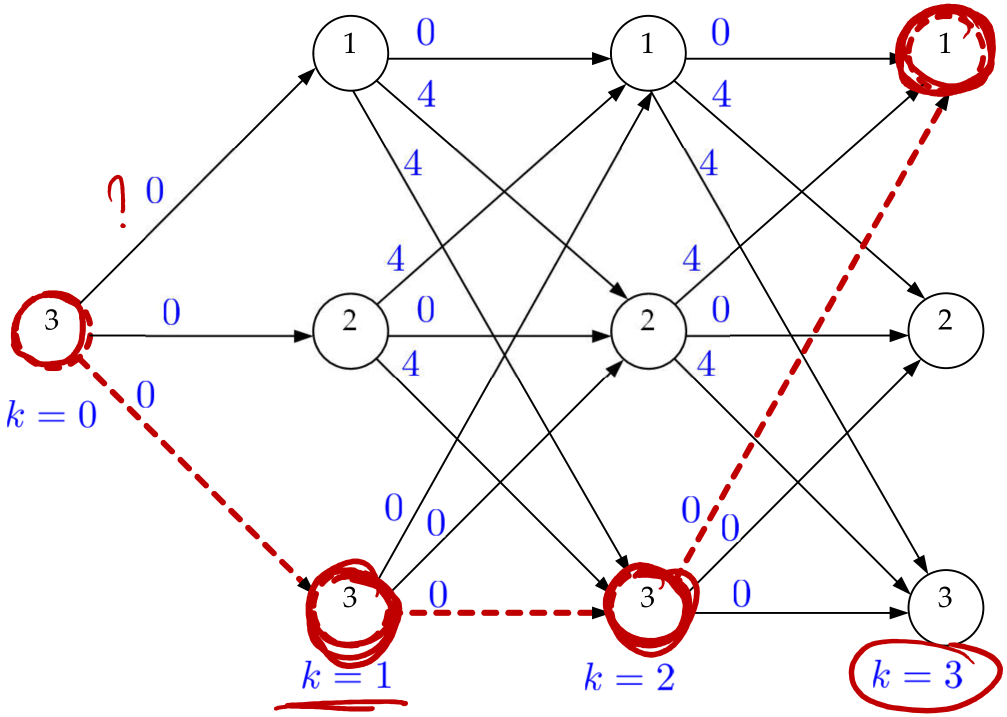
- Compute the costs at *k=0*
- Reconstruct Policy

$$\boxed{J_0^*(x_0) = \$127.2}$$

Hour

	1	2	3
P_1^*	4.5	7	4
P_2^*	1	1	0
U_1^*	1	1	1
U_2^*	1	1	0

status of generator

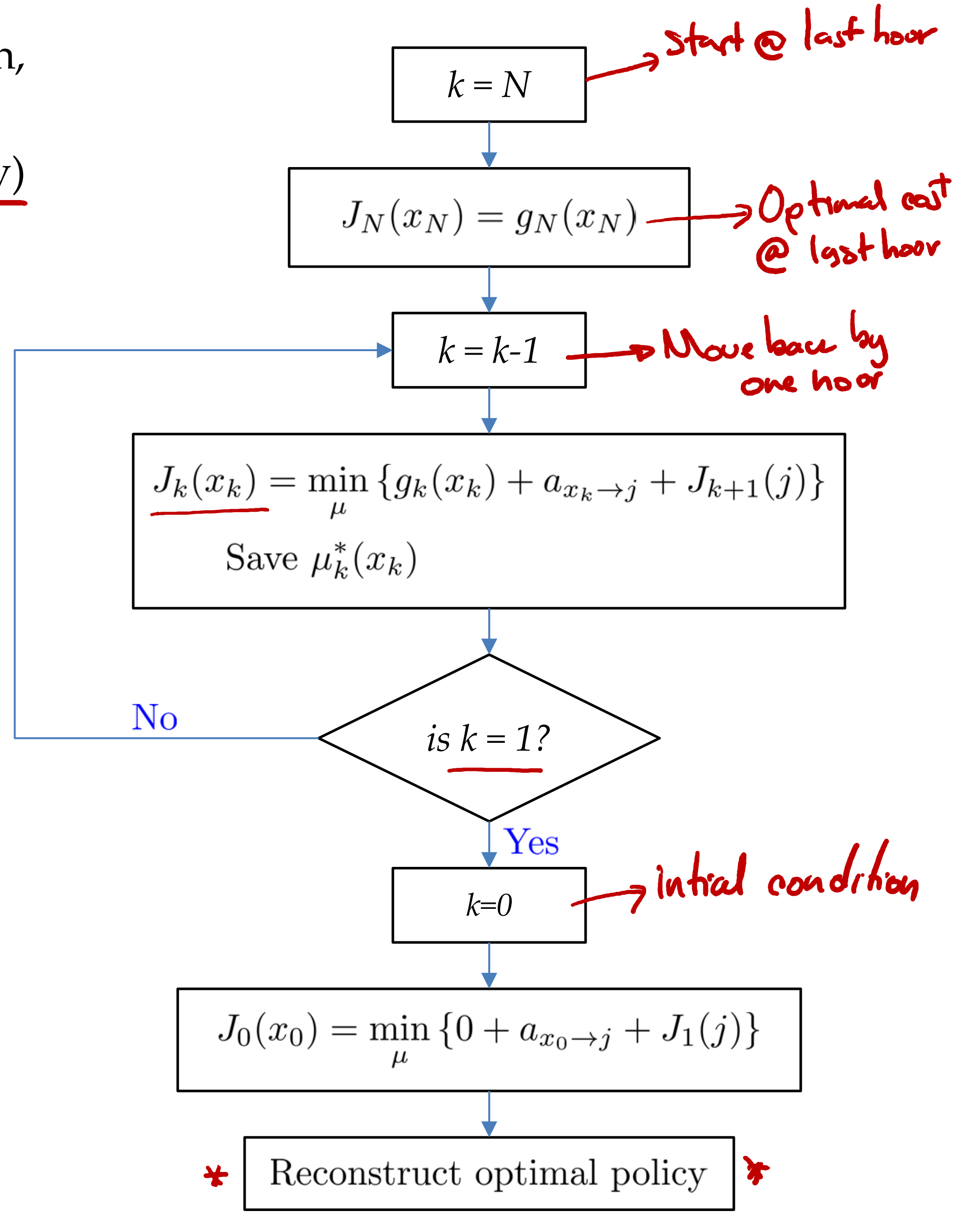


- We can also consider **shut-down costs**

(Backwards) Dynamic Programming – UC Flow Chart

- Define transition costs (start-up/shut-down, and
  static costs (matrix G or compute on the fly)
  $g_k(x_k)$
- Define initial condition x_0

• Forward Dynamic Programming (similar)
flowchart on page 174 of Book



$k = N$ → start @ last hour

$J_N(x_N) = g_N(x_N)$ → Optimal cost @ last hour

$k = k-1$ → Move back by one hour

$J_k(x_k) = \min_{\mu} \{g_k(x_k) + a_{x_k \to j} + J_{k+1}(j)\}$

Save $\mu_k^*(x_k)$

is $k = 1$?

No

Yes

$k=0$ → initial condition

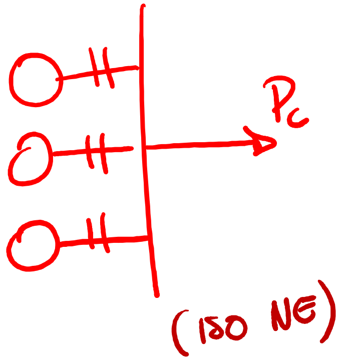
$J_0(x_0) = \min_{\mu} \{0 + a_{x_0 \to j} + J_1(j)\}$

* Reconstruct optimal policy *

Example 2 Dynamic Programming Unit Commitment

- Suppose that we have three generation units with the following characteristics:

	Gen.	$P_{\min}$	$P_{\max}$	Cost/power	Start-up cost	Shut-down costs
Cheap	1	1.5	7	7	30	1
Expensive	2	1	5	9	20	2
Middle	3	2	7	8	40	0.5

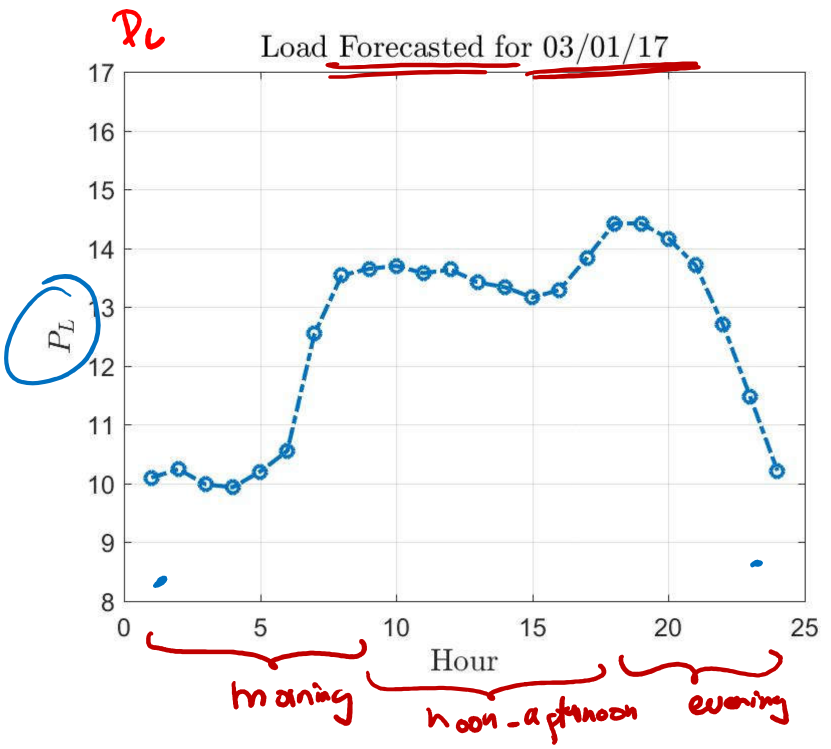


and suppose we wanted to define the commitment of the units for the day 03/01/17:

$$\boxed{(2^3-1)^{24} = 1.91\times10^{20}}$$

3 → generators

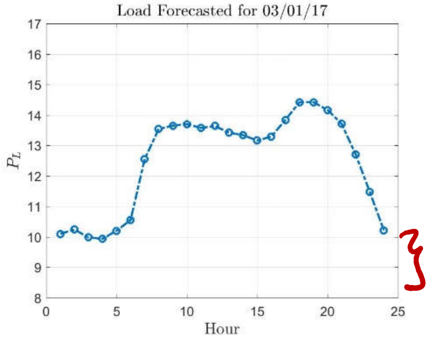
24 → Hours



Example 2 Dynamic Programming Unit Commitment

- Suppose that we have three generation units with the following characteristics:

Gen.	P_{min}	P_{max}	Cost/power	Start-up cost	Shut-down costs
1	1.5	7	7	30	1
2	1	5	9	20	2
3	2	7	8	40	0.5

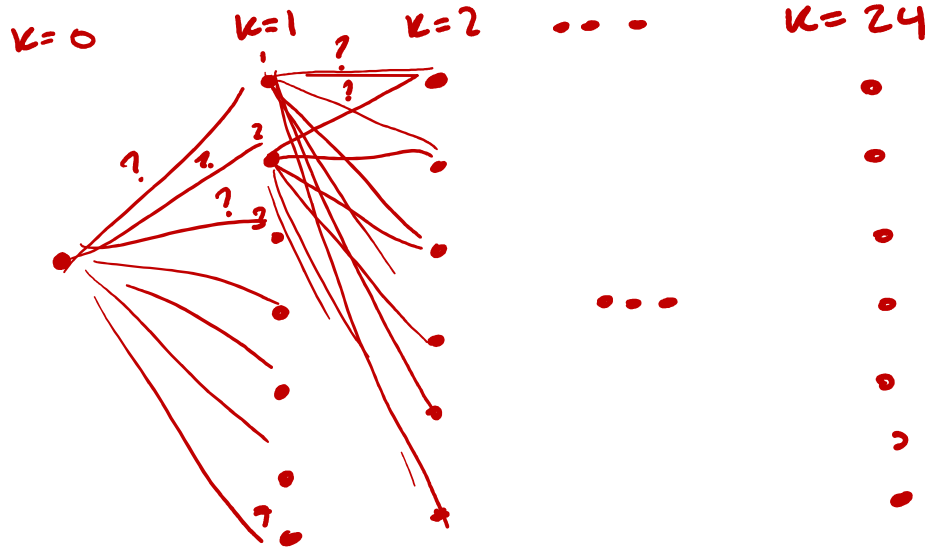


- How many different modes/states are there? (ignore mode where all units are off)

$(2^3-1) = 7$

Mode	Gen3	Gen2	Gen1
1	0	0	1
2	0	1	0
3	0	1	1
→ 4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

7 modes/states at each hour

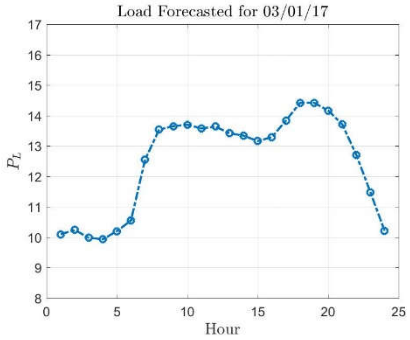


1.91×10^{20}

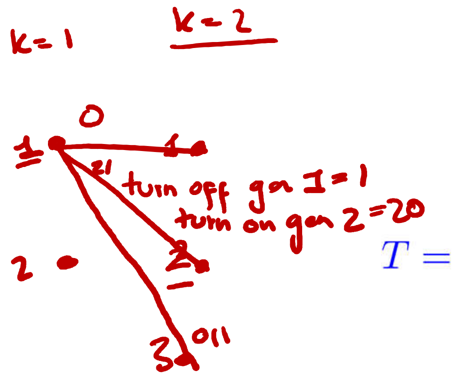
Example 2 Dynamic Programming Unit Commitment

- Suppose that we have three generation units with the following characteristics:

Gen.	P_{min}	P_{max}	Cost/power	Start-up cost	Shut-down costs
1	1.5	7	7	30	1
2	1	5	9	20	2
3	2	7	8	40	0.5



- How many different modes/states are there? (ignore mode where all units are off)
- Define a transition matrix (transition costs from one mode to another)



Begin ↓ End →

$$T = \begin{array}{c|ccccccc} & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ \hline 1 & 0 & 21 & 20 & 41 & 40 & 61 & 60 \\ 2 & 32 & 0 & 30 & 42 & 72 & 40 & 70 \\ 3 & 2 & 1 & 0 & 43 & 42 & 41 & 40 \\ 4 & 30.5 & 20.5 & 50.5 & 0 & 30 & 20 & 50 \\ 5 & 0.5 & 21.5 & 20.5 & 1 & 0 & 21 & 20 \\ 6 & 32.5 & 0.5 & 30.5 & 2 & 32 & 0 & 30 \\ 7 & 2.5 & 1.5 & 0.5 & 3 & 2 & 1 & 0 \end{array}$$

Based on startup/ shutdown cost

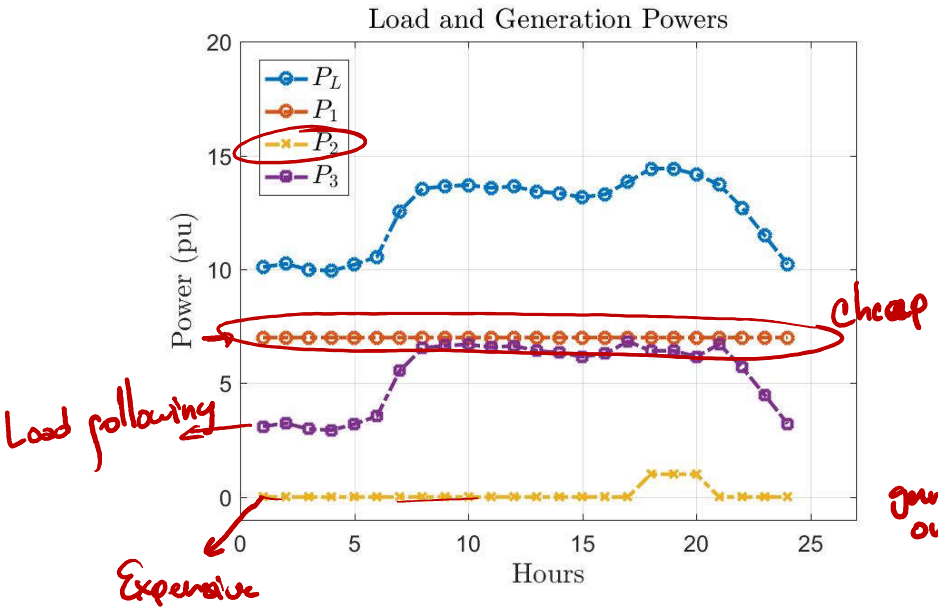
Example 2 Dynamic Programming Unit Commitment

- Suppose that we have three generation units with the following characteristics:

	Gen.	$P_{\min}$	$P_{\max}$	Cost/power	Start-up cost	Shut-down costs
Cheap	1	1.5	7	7	30	1
Expensive	2	1	5	9	20	2
Middle	3	2	7	8	40	0.5

- How many different modes/states are there? (ignore mode where all units are off)
- Assume initial state is only generator 1 on (mode 001 or 1)

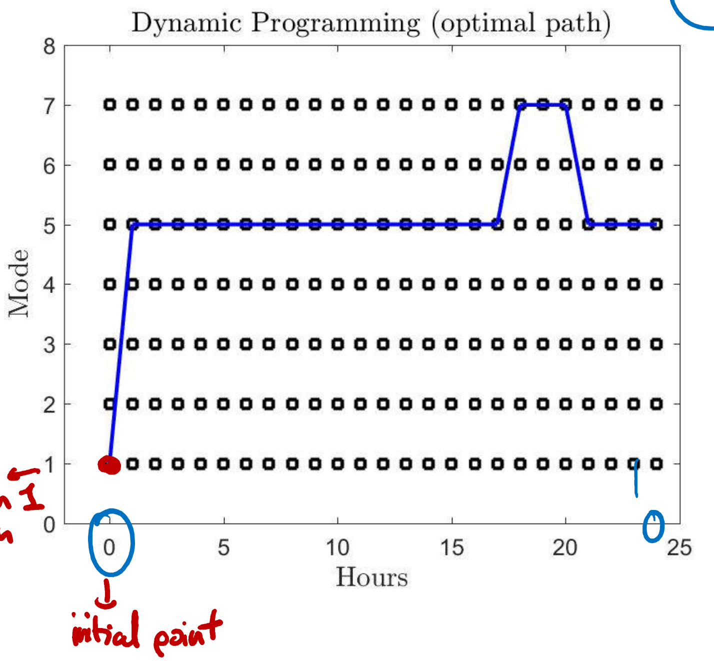
10^{20}



Load following

Cheap

Expensive



gen 1 on

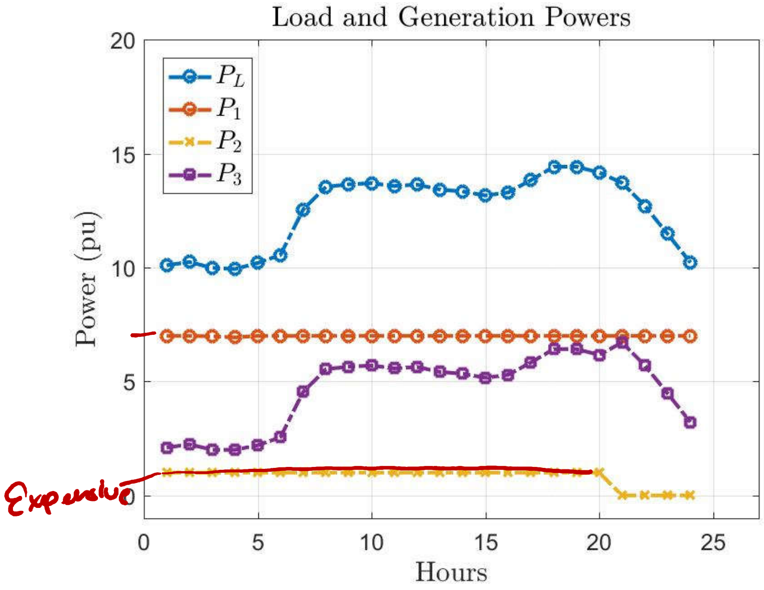
initial point

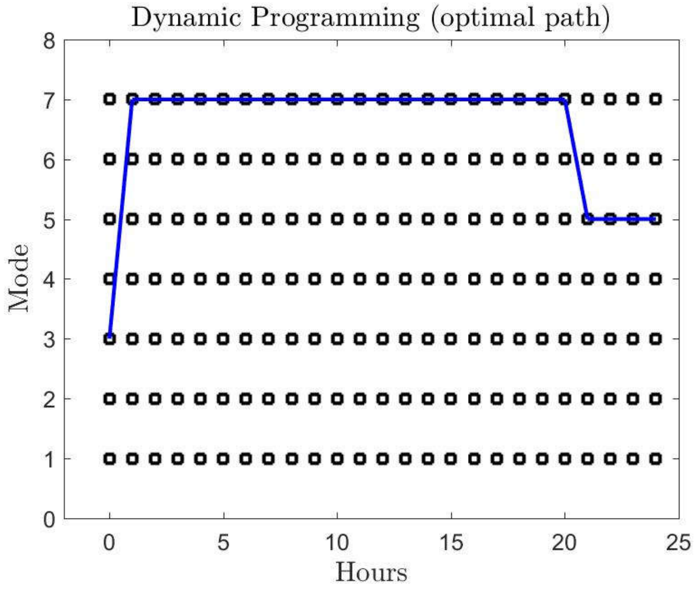
Example 2 Dynamic Programming Unit Commitment

- Suppose that we have three generation units with the following characteristics:

Gen.	$P_{\min}$	$P_{\max}$	Cost/power	Start-up cost	Shut-down costs
1	1.5	7	7	30	1
2	1	5	9	20	2
3	2	7	8	40	0.5

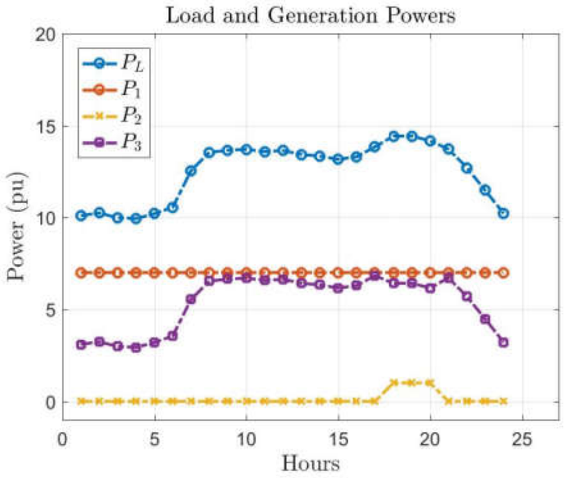
- How many different modes/states are there? (ignore mode where all units are off)
- Assume initial state is generators 1 and 2 on (mode 011 or 3)

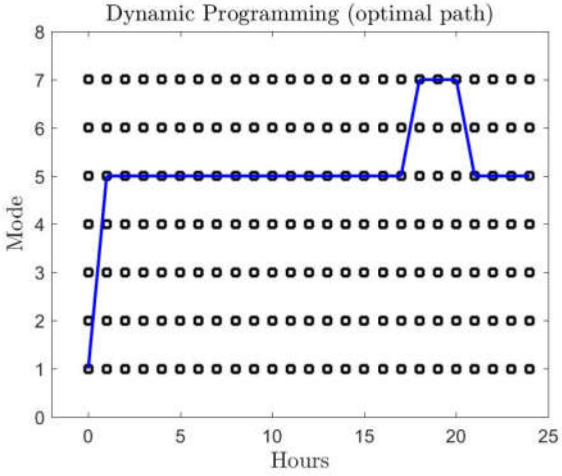




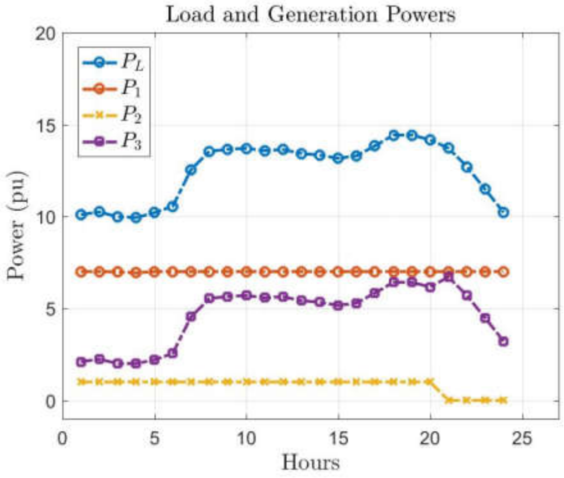
Example 2 Dynamic Programming Unit Commitment

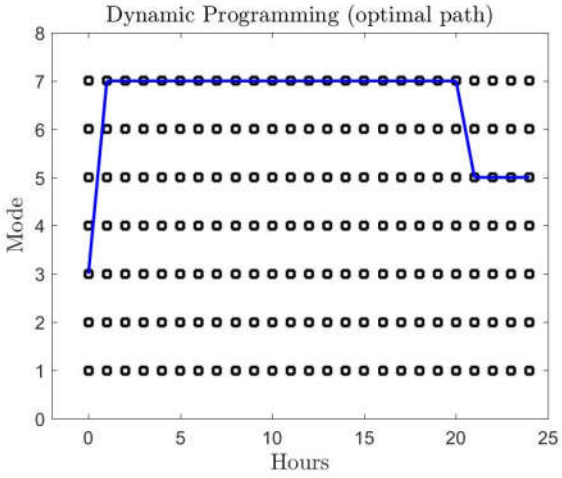
- Assuming initial condition is only generator 1 on:





- Initial condition is generator 1 and 2 on (mode 011 or 3)





Final Comments on Dynamic Programming

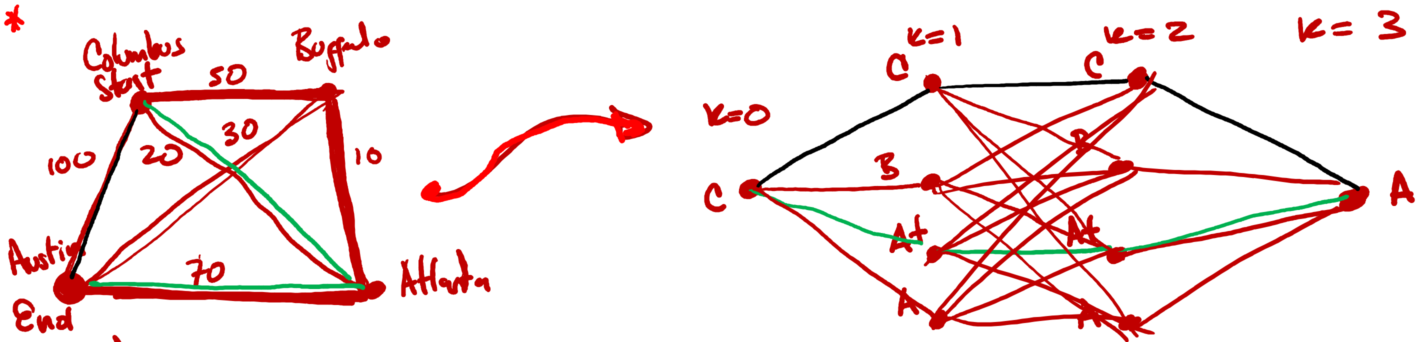
- **Principle of optimality:** An optimal path (policy) contains optimal sub-paths (policies)

- Components of dynamic programming (final costs, transition costs, policies, etc.)

- (Backwards) dynamic programming Start k=N
- Forward dynamic programming Start k=0

- *Not all problems are ready to use DP, you may have to transform it in a way to use DP*

*



(Problem 1 HW)

Dynamic Programming to find optimal cost to go from Columbus to Austin

Topics that Will be Covered

- *Unit Commitment Problem Formulation*
- *Dynamic Programming*
- ***Mixed Integer LP Formulation***
- *Other Solution Methods*
- *Examples*

What is a Mixed Integer Linear Optimization Problem (MILP)?

$\mathbb{R} = (-\infty, \infty)$

$\mathbb{Z} = \{\ldots, -2, -1, \ldots\}$

- Let's separate the set of "unknown" variables into two components:

$$x = \begin{pmatrix} x_r \\ x_i \end{pmatrix} \qquad \text{where } x_r \in \mathbb{R}^m,\ x_i \in \mathbb{Z}^p$$

real variables

integer variables

integers

$$x_r = \begin{pmatrix} x_{r_1} \\ \vdots \\ x_{r_m} \end{pmatrix} \qquad x_i = \begin{pmatrix} x_{i_1} \\ \vdots \\ x_{i_p} \end{pmatrix}$$

$c_1 \in \mathbb{R}^m \quad c_2 \in \mathbb{R}^p$

- Then, a Mixed Integer Linear Program can be formulated as follows:

$$\min_x c^T x$$

s.t.

$$A^T x \le b$$

$$A_{eq} x = b_{eq}$$

$$(x \ge 0)$$ * not used

Regular LP opt.

$$\min_x \begin{pmatrix} c_1^T & c_2^T \end{pmatrix}^T \begin{pmatrix} x_r \\ x_i \end{pmatrix} \Rightarrow \text{linear cost} = c_1^T x_r + c_2^T x_i$$

s.t.

$$\begin{pmatrix} A_1 & A_2 \end{pmatrix} \begin{pmatrix} x_r \\ x_i \end{pmatrix} \le b \Rightarrow \text{linear inequalities: } A_1 x_r + A_2 x_i \le b$$

$$\begin{pmatrix} A_{eq-1} & A_{eq-2} \end{pmatrix} \begin{pmatrix} x_r \\ x_i \end{pmatrix} = b_{eq} \Rightarrow \text{linear equalities: } A_{eq-1} x_r + A_{eq-2} x_i = b_{eq}$$

$$* \left(\begin{pmatrix} x_r \\ x_i \end{pmatrix} \ge \begin{pmatrix} 0 \\ 0 \end{pmatrix} \right)$$

MILP

$\mathbb{R}^2 \quad x_r \in \mathbb{R}^2 \quad c_1 \in \mathbb{R}^2$

$$c_1^T x_r = \begin{pmatrix} c_{1,1} \\ c_{1,2} \end{pmatrix}^T \begin{pmatrix} x_{r_1} \\ x_{r_2} \end{pmatrix} = c_{1,1} x_{r_1} + c_{1,2} x_{r_2}$$

What is a Mixed Integer Linear Optimization Problem (MILP)?

- MILP is a linear optimization problem for which the variables belong to the set of reals and/or integers

$$x = \begin{pmatrix} x_r \\ x_i \end{pmatrix} \qquad \text{where } x_r \in \mathbb{R}^m,\ x_i \in \mathbb{Z}^p$$

$$\begin{aligned} &\min_x \left(c^T x\right) \\ &\text{s.t.} \\ &\left(A^T x \le b\right) \\ &\left(A_{eq} x = b_{eq}\right) \\ &\left[x \ge 0\right] \text{ * not required} \end{aligned}$$

$$\begin{aligned} &\min_x c_1^T x_r + c_2^T x_i \\ &\text{s.t.} \\ &A_1 x_r + A_2 x_i \le b \\ &A_{eq-1} x_r + A_{eq-2} x_i = b_{eq} \\ &\left[\begin{pmatrix} x_r \\ x_i \end{pmatrix} \ge \begin{pmatrix} 0 \\ 0 \end{pmatrix}\right] \text{ * not required} \\ &\underbrace{x_r \in \mathbb{R}^m}_{\text{real}},\ \underbrace{x_i \in \mathbb{Z}^p}_{\text{integer}} \end{aligned}$$

- * **Note:** This problem is not as simple to solve as a regular linear program (non-convex)

  * we can't use CVX

- (**Solution methods:** (Algorithms)
  - **Enumeration techniques:** Branch and bound ✓
  - Cutting plane technique ✓
  - Group-theoretic techniques ✓
  - Sphere decoding algorithm.
  - Dynamic Programming.

* Matlab: intlinprog(...)

(CPLEX, GAMS)

Why MILP for Unit Commitment?

- In the Unit Commitment problem, we can consider generators that can turn on/off

$\mathbb{Z} = \{\dots, -1, 0, 1, 2, 3, \dots\}$

ON/OFF condition

- This condition can be represented by an integer variable for each generator

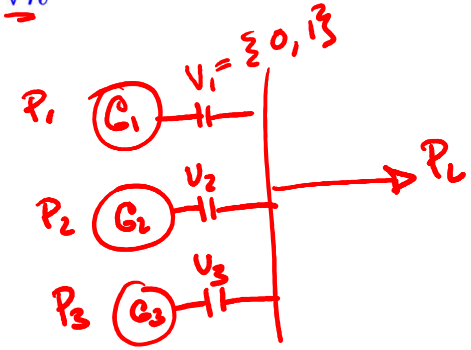
hours

$$v_i(k) \in \{0,\ 1\}, \quad \text{for } i = 1,\ \dots,\ N_g, \quad \forall k$$

\# of gens.

Status (off/on) of generator i at time k

$$v_i(k) = \begin{cases} 0 & \text{if gen } i \text{ is off @ time } k \\ 1 & \text{if gen } i \text{ is on @ time } k \end{cases}$$



- Constrain $v_i(k)$ $\forall i, \forall k$, to be between 0 and 1:

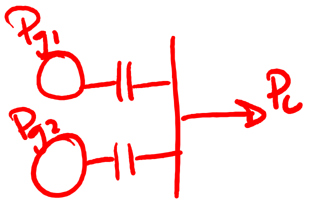
$$\boxed{\begin{array}{l} 0 \le v_i(k) \le 1 \\ v_i(k) \in \mathbb{Z} \end{array}} \Rightarrow v_i(k) \in \{0, 1\}$$

Example UC Without Start-up Costs or Shutdown Cost

- Let's consider the same problem as before but **without** start-up costs / we can still turn on/off generators

Gen.	$P_{\min}$	$P_{\max}$	Cost/power	
1	1.5	7	7.2	(Cheaper)
2	1	6	7.8	(Expensive)

Hour	1	2	3
P_L	5.5	8	4



- Formulate this problem using MILP

(Cannot use CVX)

(can use CPLEX, Matlab)

$$* \quad \min \left(\sum_{k=1}^{3} \underbrace{\underline{7.2}\, P_{g1}(k)} + \underbrace{\underline{7.8}\, P_{g2}(k)} \right) = c^T x$$

s.t.

$$\checkmark \; \left\{ P_{g1}(k) + P_{g2}(k) = P_L(k) \right. \qquad k = 1, 2, 3$$

$$\begin{cases} P_{g1}(1) + P_{g2}(1) = \underline{5.5} \\ \vdots \\ P_{g1}(3) + P_{g2}(3) = \underline{4} \end{cases}$$

$$\left[\begin{array}{l} \underline{v_1(k)}\, 1.5 \le \underline{P_{g1}(k)} \le \underline{v_1(k)}\, 7 \\ \underline{v_2(k)}\, 1 \le P_{g2}(k) \le \underline{v_2(k)}\, 6 \end{array} \right] \quad k = 1, 2, 3$$

$$\left[\begin{array}{l} 0 \le v_1(k) \le 1 \\ 0 \le v_2(k) \le 1 \end{array} \quad k = 1, 2, 3 \right]$$

$$v_1(k), v_2(k) \in \mathbb{Z}$$

Unknown variables:

$$\underbrace{\begin{array}{l} P_{g1}(1), P_{g1}(2), P_{g1}(3) \\ P_{g2}(1), P_{g2}(2), P_{g2}(3) \\ v_1(1), \ldots, v_1(3) \\ v_2(1), \ldots, v_2(3) \end{array}}_{\boxed{12 \text{ unknown}}}$$

intlinprog( )

$$x = [\underbrace{P_{g1}(1) \ldots P_{g1}(3), P_{g2}(1), \ldots P_{g2}(3)}_{6}, \underbrace{v_1(1), \ldots, v_1(3), v_2(1), \ldots v_2(3)}_{6}]^T$$

$\min c^T x$

s.t.

$A x = b_{eq}$

Example UC Without Start-up Costs

- Let's consider the same problem as before but **without** start-up costs

Gen.	$P_{\min}$	$P_{\max}$	Cost/power
1	1.5	7	7.2
2	1	6	7.8

Hour	1	2	3
P_L	5.5	8	4

$$\min \sum_{k=1}^{3} 7.2P_1(k) + 7.8P_2(k)$$

s.t.

$$1.5v_1(k) \le P_1(k) \le 7.2v_1(k)$$

$$1v_2(k) \le P_2(k) \le 6v_2(k)$$

$$P_1(k) + P_2(k) = P_L(k)$$

$$v_1(k) \in \{0,\ 1\},\ \ v_2(k) \in \{0,\ 1\} \qquad \text{for } k = 1, 2, 3$$

- Place it in vector form (for intlinprog)

$$\min \ c^T x$$

s.t.

$$Ax \le b \ ?$$

$$\boxed{A_{eq}x = b_{eq}} \ ✓$$

$$lb \le x \le ub$$

*

Step 1: Define the order of unknown variables in x

$$x = [\overbrace{\underset{1}{P_1(1)}, \ldots, \underset{2}{} \underset{3}{P_1(3)}, \underset{4}{P_2(1)} \ldots \underset{5}{} \underset{6}{P_2(3)}}^{x_r}, \overbrace{\underset{7}{v_1(1)}, \ldots, \underset{8}{} \underset{9}{v_1(3)}, \underset{10}{v_2(1)}, \ldots \underset{11}{} \underset{12}{v_2(3)}}^{x_i}]^T$$

6 zeros

$$\min_x \underbrace{[7.2\ \ 7.2\ \ 7.2 \mid 7.8\ \ 7.8\ \ 7.8\ \ 0 \cdots 0]}_{c^T} x \quad \text{(compare to prev slide)}$$

s.t.

$$\underbrace{\begin{array}{c} \overbrace{\quad}^{P_{g1}(k)} \ \overbrace{\quad}^{P_{g2}(k)} \ 12 \\ \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & \cdots & 0 \end{bmatrix} \end{array}}_{Aeq} x = \underbrace{\begin{bmatrix} 5.5 \\ 8 \\ 4 \end{bmatrix}}_{beq} \rightarrow$$

1 $P_{g1}(1) + P_2(1) = 5.5$ ✓

2 $P_{g1}(2) + P_{g2}(2) = 8$

3 $P_{g1}(3) + P_{g2}(3) = 4$

$$A_{eq} = [\,I_{3\times3} \quad I_{3\times3} \quad O_{3\times6}\,]$$

$Ax \le b$ (inequalities on next slide)

$lb \le x \le ub$ (next slide)

Example UC Without Start-up Costs

- Let's consider the same problem as before but **without** start-up costs

Gen.	$P_{\min}$	$P_{\max}$	Cost/power
1	1.5	7	7.2
2	1	6	7.8

Hour	1	2	3
P_L	5.5	8	4

$$\min_x c_1^T x_r + c_2^T x_i$$

s.t.

$$A_1 x_r + A_2 x_i \le b$$

$$A_{eq-1} x_r + A_{eq-2} x_i = b_e$$

$$\begin{pmatrix} x_r \\ x_i \end{pmatrix} \ge \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$x_r \in \mathbb{R}^m, \quad x_i \in \mathbb{Z}^p$$

- Place it into general form: →

min $c^T x$ ✓

s.t.

$A_{eq} x = b_{eq}$ ✓

* $Ax \le b$ ✓

$\boxed{lb \le x \le ub}$

rows

$$A = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \end{bmatrix}$$

Example:

1st hour for gen 1

$P_1(1) \le 7.2 v_1(1)$ ✓

$1.5 v_1(1) \le P_1(1)$

$$\min \left(\sum_{k=1}^{3} 7.2 P_1(k) + 7.8 P_2(k) \right) ✓$$

s.t.

* $1.5 v_1(k) \le P_1(k) \le 7.2 v_1(k)$

$1 v_2(k) \le P_2(k) \le 6 v_2(k)$

$\big(P_1(k) + P_2(k) = P_L(k)\big)$ ✓ ✓

$\boxed{v_1(k) \in \{0,\ 1\},\ \ v_2(k) \in \{0,\ 1\}}$

for $k = 1, 2, 3$

$P_1(1) - 7.2 v_1(1) \le 0$ ✓

$a_1 = [1\ 0\ 0\ 0\ 0\ 0\ -7.2\ 0\ 0\ 0\ 0\ 0]x \le 0$

$-P_1(1) + 1.5 v_1(1) \le 0$

$a_2 = [-1\ 0\ 0\ 0\ 0\ 0\ 1.5\ 0\ 0\ 0\ 0\ 0]x \le 0$

$$\underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & -7.2 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 1.5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & -7.2 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1.5 & 0 & 0 & 0 & 0 \\ & & & & & & \vdots & & & & & \end{bmatrix}}_{A} x \le \underbrace{\begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}}_{b}$$

$$\underbrace{\begin{pmatrix} -\infty \\ -\infty \\ \vdots \\ -\infty \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}}_{lb} \le x = \begin{pmatrix} P_1(1) \\ P_1(2) \\ P_1(3) \\ P_2(1) \\ P_2(2) \\ P_2(3) \\ v_1(1) \\ v_1(2) \\ v_1(3) \\ v_2(1) \\ v_2(2) \\ v_2(3) \end{pmatrix} \le \underbrace{\begin{pmatrix} \infty \\ \infty \\ \vdots \\ \infty \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}}_{ub}$$

Example UC Without Start-up Costs

- Let’s consider the same problem as before but **without** start-up costs

Gen.	$P_{\min}$	$P_{\max}$	Cost/power	Start-up cost
1	1.5	7	7.2	4
2	1	6	7.8	4

Hour	1	2	3
P_L	5.5	8	4

will upload to UBLearns

- Solve using Matlab (intlinprog)!
- Solution:

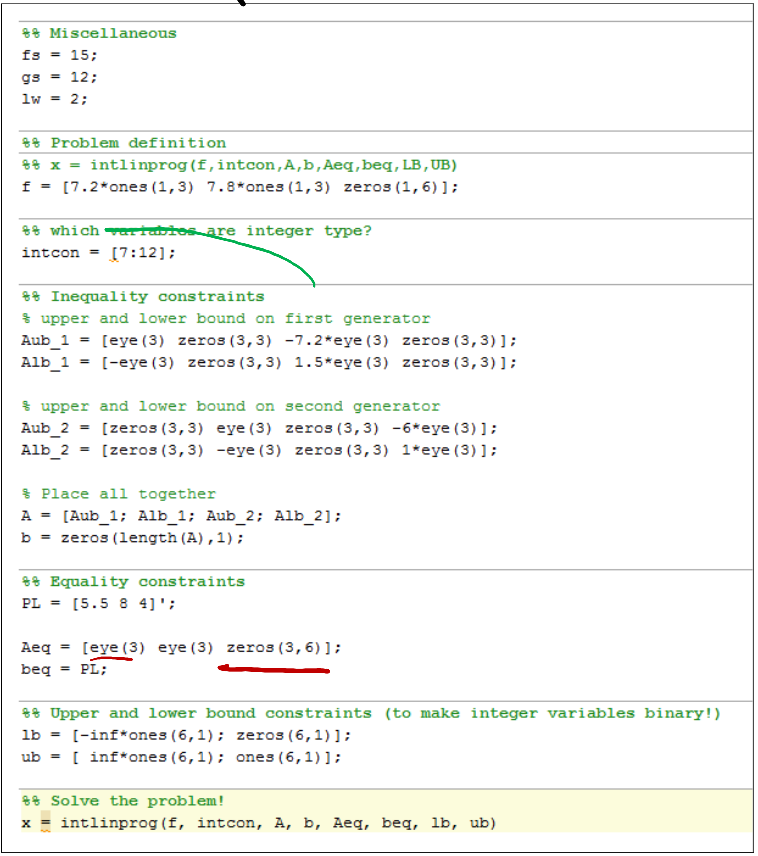
$$x = \begin{pmatrix} P_1(1) \\ P_1(2) \\ P_1(3) \\ P_2(1) \\ P_2(2) \\ P_2(3) \\ v_1(1) \\ v_1(2) \\ v_1(3) \\ v_2(1) \\ v_2(2) \\ v_2(3) \end{pmatrix} = \begin{pmatrix} 5.5 \\ 7 \\ 4 \\ 0 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

$$\text{eye}(3) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\text{eye}(2) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$\text{zeros}(\underset{\text{rows}}{3},\underset{\text{columns}}{6}) = \begin{bmatrix} 0 & \cdots & 0 \\ 0 & \cdots & 0 \\ 0 & \cdots & 0 \end{bmatrix}$$

$$\text{ones}(\underset{\text{rows}}{1},\underset{\text{columns}}{3}) = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$$

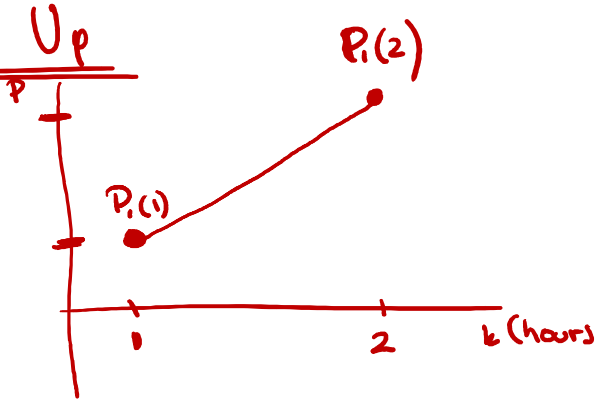


```
%% Miscellaneous
fs = 15;
gs = 12;
lw = 2;

%% Problem definition
%% x = intlinprog(f,intcon,A,b,Aeq,beq,LB,UB)
f = [7.2*ones(1,3) 7.8*ones(1,3) zeros(1,6)];

%% which variables are integer type?
intcon = [7:12];

%% Inequality constraints
% upper and lower bound on first generator
Aub_1 = [eye(3) zeros(3,3) -7.2*eye(3) zeros(3,3)];
Alb_1 = [-eye(3) zeros(3,3) 1.5*eye(3) zeros(3,3)];

% upper and lower bound on second generator
Aub_2 = [zeros(3,3) eye(3) zeros(3,3) -6*eye(3)];
Alb_2 = [zeros(3,3) -eye(3) zeros(3,3) 1*eye(3)];

% Place all together
A = [Aub_1; Alb_1; Aub_2; Alb_2];
b = zeros(length(A),1);

%% Equality constraints
PL = [5.5 8 4]';

Aeq = [eye(3) eye(3) zeros(3,6)];
beq = PL;

%% Upper and lower bound constraints (to make integer variables binary!)
lb = [-inf*ones(6,1); zeros(6,1)];
ub = [ inf*ones(6,1); ones(6,1)];

%% Solve the problem!
x = intlinprog(f, intcon, A, b, Aeq, beq, lb, ub)
```

Ramping Up/Down Constraints

- A new type of constraint that can be included is the ramp-up and/or ramp-down constraint
- * Limit how fast a generator can increase or lower its output power, e.g. a generator cannot increase/decrease its power by more than 200 MW/hour
- How can we take these constraints into account? (in the MILP)

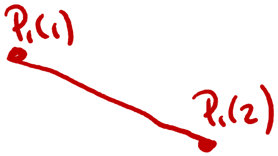
Ramp Up



$$P_i(k+1) - P_i(k) \le 200\text{ MW} \qquad k=2,3,\dots,N$$

$$P_i(1) - \underbrace{P_i(0)}_{\text{initial condition - known beforehand}} \le 200\text{ MW}$$

Ramp down

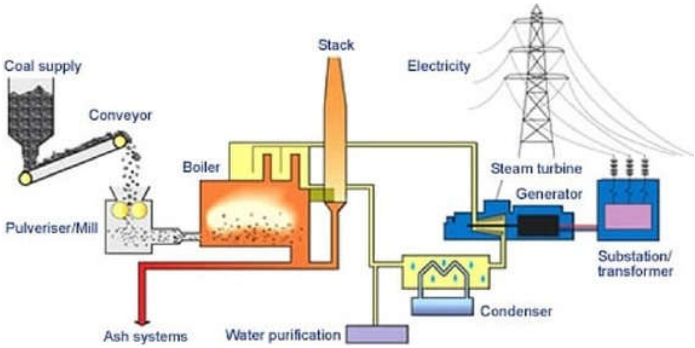


$$P_i(k) - P_i(k+1) \le 200 \qquad k=2,\dots,N$$

$$\underbrace{P_i(0)}_{\text{initial condition}} - P_1(1) \le 200$$

Start-up Costs

- Recall, that to turn on a thermal generating unit there is a costs associated with it



- How can we include this cost in the Unit Commitment problem?

• Define a new variable $s_{\underset{\text{generator}}{i}}(\overset{\text{hour}}{k}) = \begin{cases} 0 & \text{o.w} \\ 1 & \text{if gen } i \text{ turns on @ hour } k \end{cases}$

Status of gen i		
$v_i(k-1)$	$v_i(k)$	$s_i(k)$
0	0	0
0	1	1
1	0	0
1	1	0 *

Goal: write $s_i(k) = f(v_i(k-1), v_i(k))$

Attempt 1: $s_i(k) = v_i(k)\,(1 - v_i(k-1))$

work? yes is function linear or nonlinear? can't use MILP

* Attempt 2: linear $\begin{cases} s_i(k) \geq v_i(\underset{0}{k}) - v_i(\underset{0}{k}-1) \\ s_i(k) \geq 0 \\ s_i(k) \leq 1 \end{cases}$

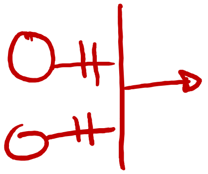
linear ✓

- **Can we formulate shut-down costs similarly?** yes (make a truth table and formulate)

Example with Start-up Costs

- Formulate the previous problem, including start-up costs, as a MILP (compare with the Dynamic Programming solution)
- **Initial condition is all generators are on**

$V_1(0) = 1$
$V_2(0) = 1$



Gen.	$P_{\min}$	$P_{\max}$	Cost/power	Start-up cost
1	1.5	7	7.2	4
2	1	6	7.8	4

Hour	1	2	3
P_L	5.5	8	4

Write opt. Problem

$$\min_{x} \sum_{k=1}^{3} 7.2 P_1(k) + 7.8 P_2(k) + \underbrace{4 S_1(k) + 4 S_2(k)}_{\text{startup cost}}$$

$$V_1(k) 1.5 \le P_1(k) \le V_1(k) 7$$
$$V_2(k) 1 \le P_2(k) \le V_2(k) 6 \qquad k = 1,2,3$$

$$P_1(k) + P_2(k) = P_L(k)$$

gen 1 $\left[\begin{array}{l} S_1(k) \ge V_1(k) - V_1(k-1) \quad k = 2,3 \\ S_1(1) \ge V_1(1) - \underbrace{V_1(0)}_{1} \end{array}\right.$

gen2 $\left[\begin{array}{l} S_2(k) \ge V_2(k) - V_2(k-1) \quad k = 2,3 \\ S_2(1) \ge V_2(1) - V_2(0) \end{array}\right.$

$V_1(k), V_2(k), S_1(k), S_2(k) \in \{0,1\}$ $(lb \le x \le ub)$

Step 1 Define x

How many unknowns? 18

$$x = [P_1(1) \ldots P_1(3)\ P_2(1) \ldots P_2(3)\ V_1(1) \ldots V_1(3)\ V_2(1) \ldots V_2(3)\ S_1(1) \ldots S_1(3)\ S_2(1) \ldots S_2(3)]^T$$

$$\min\ c^T x$$
s.t
$$A_{eq} x = b_{eq}$$
$$Ax \le b$$
$$lb \le x \le ub$$

Example with Start-up Costs

- Formulate the previous problem, including start-up costs, as a MILP (compare with the Dynamic Programming solution)
- **Initial condition is all generators are on**

Gen.	$P_{\min}$	$P_{\max}$	Cost/power	Start-up cost
1	1.5	7	7.2	4
2	1	6	7.8	4

Hour	1	2	3
P_L	5.5	8	4

- Problem formulation – define variables

$$\min \sum_{k=1}^{3} 7.2P_1(k) + 7.8P_2(k) + 4s_1(k) + 4s_2(k)$$

s.t.

$$1.5v_1(k) \leq P_1(k) \leq 7.2v_1(k)$$

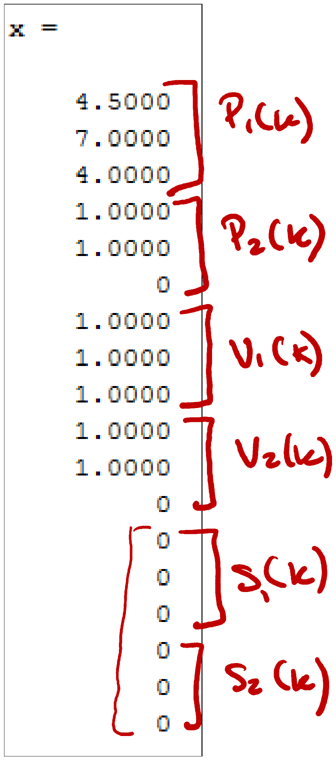
$$1v_2(k) \leq P_2(k) \leq 6v_2(k)$$

$$P_1(k) + P_2(k) = P_L(k)$$

$$s_1(k) \geq v_1(k) - v_1(k-1), \quad s_2(k) \geq v_2(k) - v_2(k-1) \quad \text{for } k = 2, 3$$

$$s_1(1) \geq v_1(1) - v_1(0), \quad s_2(1) \geq v_2(1) - v_2(0) \quad \text{for } k = 2, 3$$

$$v_1(k) \in \{0,\ 1\},\ \ v_2(k) \in \{0,\ 1\} \qquad \text{for } k = 1, 2, 3$$



```
x =

    4.5000
    7.0000
    4.0000
    1.0000
    1.0000
         0
    1.0000
    1.0000
    1.0000
    1.0000
    1.0000
         0
         0
         0
         0
         0
         0
         0
```

Example with Start-up Costs

- Formulate the previous problem, including start-up costs, as a MILP (compare with the Dynamic Programming solution)
- **What if initial condition is all generators off?**

Gen.	$P_{\min}$	$P_{\max}$	Cost/power	Start-up cost
1	1.5	7	7.2	4
2	1	6	7.8	4

Hour	1	2	3
P_L	5.5	8	4

- Problem formulation – define variables

$$\min \sum_{k=1}^{3} 7.2P_1(k) + 7.8P_2(k) + \underbrace{4s_1(k) + 4s_2(k)}_{\text{Startup}}$$

s.t.

$$1.5v_1(k) \le P_1(k) \le 7.2v_1(k)$$

$$1v_2(k) \le P_2(k) \le 6v_2(k)$$

$$P_1(k) + P_2(k) = P_L(k)$$

$$s_1(k) \ge v_1(k) - v_1(k-1), \quad s_2(k) \ge v_2(k) - v_2(k-1) \quad \text{for } k = 2, 3$$

$$s_1(1) \ge v_1(1) - v_1(0), \quad s_2(1) \ge v_2(1) - v_2(0) \quad \text{for } k = 2, 3$$

$$v_1(k) \in \{0,\ 1\}, \quad v_2(k) \in \{0,\ 1\} \qquad \text{for } k = 1, 2, 3$$

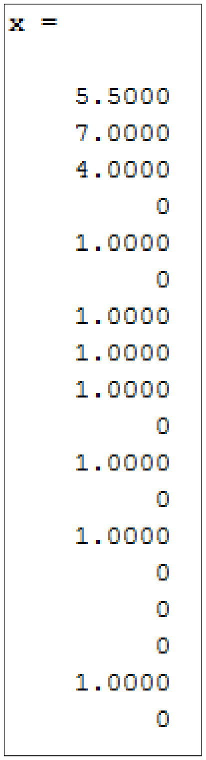
$\min\ c^T x$

s.t.

$Ax \le b$

$A_{eq}x = b_{eq}$

$lb \le x \le ub$



```
x =

    5.5000
    7.0000
    4.0000
         0
    1.0000
         0
    1.0000
    1.0000
    1.0000
         0
    1.0000
         0
    1.0000
         0
         0
         0
    1.0000
         0
```

Start-up and Shut-down Costs

- How can we take both start-up and shut-down costs into account?

- If we have both startup and shutdown cost, logic can be simplified
- Define $S_i(k)$ startup and $Sh_i(k)$ shut-down

*

$$S_i(k) - Sh_i(k) = V_i(k) - V_i(k-1) \quad k=2,\dots N$$

$$S_i(1) - Sh_i(1) = V_i(1) - \underbrace{V_i(0)}_{\text{known}} \quad k=1$$

*

$$\underbrace{S_i(k), Sh_i(k)}, V_i(k) \in \{0,1\}$$

$V_i(k-1)$	$V_i(k)$	$S_i(k)$	$Sh_i(k)$
0	0	0	0
0	1	1	0
1	0	0	1
1	1	0	0

Summary of Start-up and/or Shut-down Costs

- If we have **start-up** costs **only:**

$S_i(k) \geq v_i(k) - v_i(k-1)$ $k=2,\dots,N$

$S_i(1) \geq v_i(1) - \underbrace{v_i(0)}_{\text{known}}$

and add startup cost to cost function

- If we have **shut-down** costs **only:**

$Sh_i(k) \geq v_i(k-1) - v_i(k)$ $k=2,\dots,N$

$Sh_i(1) \geq \underbrace{v_i(0)}_{\text{known}} - v_i(1)$

" " shutdown cost

- If we have **both start-up and shut-down** costs:

prev slide

and add startup and shutdown cost to the cost function

Topics that Will be Covered

- *Unit Commitment Problem Formulation*
- *Dynamic Programming*
- *Mixed Integer LP Formulation*
- ***Other Solution Methods – Final Comments***

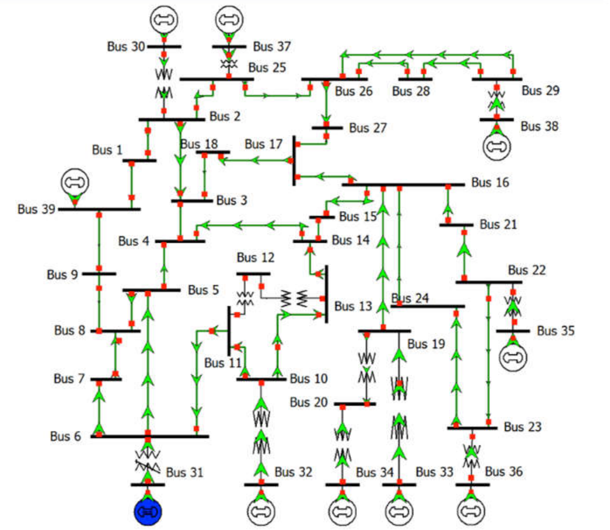
Unit Commitment – Solution Methods

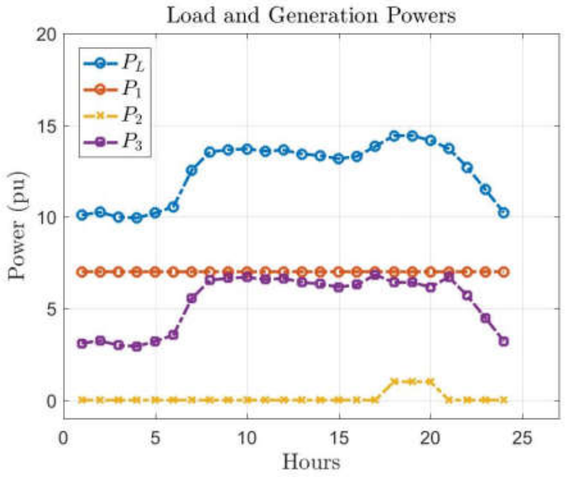
- The goal of the UC is to "commit" generation units (on-off) and optimize their output power for a certain period of time (e.g. 24 hours)

- So far, we have studied two methods (ways) that can be used for solving this problem:
  1. Dynamic Programming (algorithm) used to solve
  2. MILP (formulation)

  ↓ use software (matlab, CPLEX) to give a solution

- Other solution ~~methods~~ (algorithms) utilized:
  - Lagrange relaxation
  - Benders decomposition





Final Thoughts – Unit Commitment

- The goal of the UC is to "commit" generation units (on-off) and optimize their output power for a certain period of time

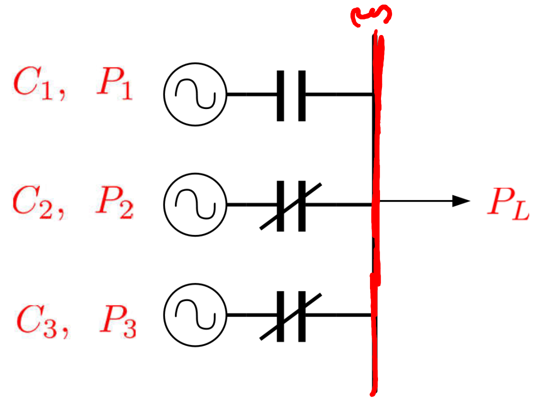
- We have studied two methods that can be used for solving this problem:
  1. Dynamic Programming
  2. Mixed Integer Linear Program

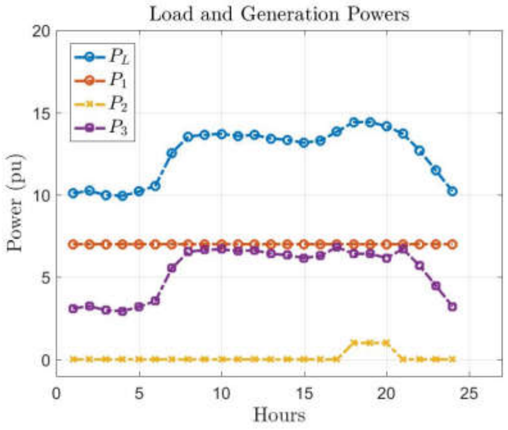
- Unit Commitment is used mainly in US electricity markets **(see NE ISO document on UBLearns)**

- PowerGlobe forum: (General) http://www.ece.mtu.edu/faculty/ljbohman/peec/globe/

- We did not take network constraints into consideration:

  Unit commitment + network/security constraints
  { = Security Constrained Unit Commitment (SCUC)}





Homework 5 (Problem 4) placing problems in matrix forms.

1) Define

$$\underline{\underline{x}} = [\underbrace{\overset{1}{P_1(1)}\ \overset{2}{P_1(2)}\ \overset{3}{P_1(3)}}\ \underbrace{\overset{4}{P_2(1)}\ \overset{5}{P_2(2)}\ \overset{6}{P_2(3)}}\ \overset{7}{V_1(1)}\ \overset{8}{V_1(2)}\ \overset{9}{V_1(3)}\ \overset{10}{V_2(1)}\ \overset{11}{V_2(2)}\ \overset{12}{V_2(3)}\ \underbrace{\overset{13}{S_1(1)}\ \overset{14}{S_1(2)}\ \overset{15}{S_1(3)}} \ldots \underbrace{\overset{16}{S_2(1)}\ \overset{17}{S_2(2)}\ \overset{18}{S_2(3)}}]^T$$

$$C^T = [7.2\ 7.2\ 7.2\ 7.8\ 7.8\ 7.8\ 0\ 0\ 0\ 0\ 0\ 0\ \underbrace{\overset{13}{4}\ \overset{14}{4}\ \overset{15}{4}\ \overset{16}{4}\ \overset{17}{4}\ \overset{18}{4}}]$$

$C^T x$

18 unknowns $x \in \mathbb{F}^{\textcircled{18}}$

integer variables = $\underline{7 - 18}$

Example with startup cost only

$$\underline{\underline{\min}} \sum_{k=1}^{\textcircled{3}} 7.2\underline{P_1(k)} + 7.8\underline{P_2(k)} + 4\underline{s_1(k)} + 4\underline{s_2(k)} = 7.2P_1(1) + \ldots + 7.2P_1(3) + 7.8P_2(1) + \ldots + 7.8P_2(3) + 4s_1(1) + \ldots + 4s_1(3) + 4s_2(1) + \ldots + 4s_2(3)$$

$$= \underset{1\times 18}{C^T} \underset{18\times 1}{\underline{\underline{x}}}$$

s.t.

$$1.5\,v_1(k) \le P_1(k) \le 7.2\,v_1(k)$$

$$1v_2(k) \le P_2(k) \le 6v_2(k)$$

$$\boxed{P_1(k) + P_2(k) = P_L(k)}$$

$$\left\{\begin{array}{l} P_1(1) + P_2(1) = P_L(1) \\ \vdots \\ P_1(3) + P_2(3) = P_L(3) \end{array}\right. \rightarrow A_{eq}\underline{x} = b_{eq}$$

$s_1(k) \ge v_1(k) - v_1(k-1)$, $s_2(k) \ge v_2(k) - v_2(k-1)$ for $k = 2, 3$

$s_1(1) \ge v_1(1) - v_1(0)$, $s_2(1) \ge v_2(1) - v_2(0)$ for $k = 2, 3$

$v_1(k) \in \{0, 1\}$, $v_2(k) \in \{0, 1\}$ for $k = 1, 2, 3$

min $\underline{c}^T x$

s.t.

$\underline{Ax \le b}$

$\underline{A_{eq}x = b_{eq}}$

$\underline{lb \le x \le ub}$

- **Example**

$$x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} \qquad f(x) = 3x_2 + 5x_4$$

$$f(x) = c^T x = (0\ 3\ 0\ 5) \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = 3x_2 + 5x_4$$

Example 2

$$\text{2eq. const} \left\{ \begin{aligned} 3x_1 + 2 &= 5x_4 \\ 2x_3 + 1x_1 &= 3 \end{aligned} \right\} \longrightarrow A_{eq} x = b_{eq}$$

$$\begin{aligned} 3x_1 - 5x_4 &= -2 \\ x_1 + 2x_3 &= 3 \end{aligned} \longrightarrow \begin{aligned} 3x_1 + 0x_2 + 0x_3 - 5x_4 &= -2 \\ x_1 + 0x_2 + 2x_3 + 0x_4 &= 3 \end{aligned}$$

$$\downarrow$$

$$\underset{\substack{2\times 4 \\ A_{eq}}}{\begin{bmatrix} 3 & 0 & 0 & -5 \\ 1 & 0 & 2 & 0 \end{bmatrix}} \underset{\substack{4\times 1 \\ x}}{\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}} = \underset{\substack{2\times 1 \\ b_{eq}}}{\begin{bmatrix} -2 \\ 3 \end{bmatrix}}$$

Example 3

$$\begin{aligned} 3x_1 &\geq 5x_2 \\ 2x_3 - 4x_4 &\leq 5 \end{aligned}$$

$$\downarrow$$

$$\begin{aligned} -3x_1 + 5x_2 &\leq 0 \\ 2x_3 - 4x_4 &\leq 5 \end{aligned}$$

$$\downarrow$$

$$\overbrace{\begin{bmatrix} -3 & 5 & 0 & 0 \\ 0 & 0 & 2 & -4 \end{bmatrix}}^{A} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \leq \overbrace{\begin{bmatrix} 0 \\ 5 \end{bmatrix}}^{b}$$

$$Ax \leq b$$

$$3\begin{cases} P_1(1) + P_2(1) = P_c(1) \\ P_1(2) + P_2(2) = P_c(2) \\ P_1(3) + P_2(3) = P_c(3) \end{cases} \longrightarrow \underline{A_{eq}\,x = b_{eq}}$$

$$x = \Big[\underbrace{P_1(1)\ \overset{1}{\dots}\ P_1(3)\ \underline{P_2(1)}\dots P_2(3)}_{1-6}\ \underbrace{\overset{7}{V_1(1)}\dots U_2(3)\ S_1(1)\dots S_2(3)}_{7-18}\Big]^T$$

12 zeros

$$\begin{array}{cccccccccccccccccc}
P_1(k) & & & P_2(k) & & & 7 & & & & & & & & & & & 18 \\
1 & 0 & 0 & 1 & 0 & 0 & 0 & \cdot & & & & & & & & & & 0 \\
0 & 1 & 0 & 0 & 1 & 0 & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & & & & & & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & 0 & \cdot & \cdot & - & - & - & & & & & & 0 \\
0 & 0 & 0 & 0 & 0 & 0, & 1 & 0 & 0, & 1 & 0 & 0, & -1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0, & 0 & \textcircled{1} & 0, & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0, & 0 & 0 & 1, & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 \\
 & & & & & & V_1(1) & \underline{V_1(2)} & V_1(3) & & & & & & & & &
\end{array}$$

$$x = \begin{bmatrix} P_c(1) \\ P_c(2) \\ P_c(3) \\ 0 \\ 0 \\ 0 \end{bmatrix} \qquad \underline{18\times1}$$

- We decided to add one more eq. constraint

$$\begin{cases} V_1(1) + V_2(1) = S_1(1) \\ V_1(2) + V_2(2) = S_1(2) \\ V_1(3) + V_2(3) = S_1(3) \end{cases}$$

$$\downarrow$$

$$\begin{cases} V_1(1) + V_2(1) - \underline{S_1(1)} = 0 \\ \textcircled{V_1(2)} + V_2(2) - S_1(2) = 0 \\ V_1(3) + V_2(3) - S_1(3) = 0 \end{cases}$$

R

1×18

$a_1 = [\qquad\qquad\qquad]$

$a_2 = [\qquad\qquad\qquad]$

$\Downarrow$

$A = [a_1;\ a_2;\ \dots]$