Electric Generator Control
MDP Example

This Hypercard stack was prepared by:
Dennis L. Bricker,
Dept. of Industrial Engineering,
University of Iowa,
Iowa City, Iowa 52242
e-mail: dbricker@icaen.uiowa.edu

author

An electricity plant has two generators for generating electricity.
The demand fluctuates during the day, which is divided into 4-hour blocks of time.
d_k = demand during block k, k=1,2,3,4,5,6

An excess of electricity produced cannot be stored.
c_j = capacity (kWh /4hr) of electricity of generator j
r_j = operating cost/4hr of generator j
S_j = setup cost to turn on an idle generator

What control rule will minimize the daily cost?

Data

k	1	2	3	4	5	6
d_k	20	40	60	90	70	30

j	c_j	S_j	r_j
1	40	500	1000
2	60	300	1100

States (t, m) where

t = period of the day t=1,2,...6;

m indicates which generators are on:

- 1: generator 1 only
- 2: generator 2 only
- 3: both generators

Actions k =

- 1: generator 1 only
- 2: generator 2 only
- 3: both generators

States

i	name	
1	1	1
2	1	2
3	1	3
4	2	1
5	2	2
6	2	3
7	3	1
8	3	2
9	3	3
10	4	1
11	4	2
12	4	3
13	5	1
14	5	2
15	5	3
16	6	1
17	6	2
18	6	3

actions

k	name
1	Use generator 1
2	Use generator 2
3	Use both generators

action:

1	Use generator 1

Transition Matrix

from \ to	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
16	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

action:

2	Use generator 2

Transition Matrix

from \ to	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
16	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

action:

3	Use both generators

Transition Matrix

from \ to	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
16	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

k:	1	2	3	1	2	3	1	2	3	1	2	3	1	2
i:	1	1	1	2	2	2	3	3	3	4	4	4	5	5
Min	1000	1400	2400	1500	1100	2600	1000	1100	2100	1000	1400	2400	1500	1100
	1	1	1	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	1	1	1	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	1	1	1	0	0	0	0	0
	¯1	0	0	¯1	0	0	¯1	0	0	1	1	1	0	0
	0	¯1	0	0	¯1	0	0	¯1	0	0	0	0	1	1
	0	0	¯1	0	0	¯1	0	0	¯1	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	¯1	0	0	¯1	0
	0	0	0	0	0	0	0	0	0	0	¯1	0	0	¯1
	0	0	0	0	0	0	0	0	0	0	0	¯1	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	1	1	1	1	1	1	1	1	1	1	1	1	1

3	1	2	3	2	3	2	3	2	3	3	3	3	3	3	3
5	6	6	6	7	7	8	8	9	9	10	11	12	13	14	15
2600	1000	1100	2100	1400	2400	1100	2600	1100	2100	2400	2600	2100	2400	2600	2100
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
0	¯1	0	0	1	1	0	0	0	0	0	0	0	0	0	0
0	0	¯1	0	0	0	1	1	0	0	0	0	0	0	0	0
¯1	0	0	¯1	0	0	0	0	1	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
0	0	0	0	¯1	0	¯1	0	¯1	0	0	1	0	0	0	0
0	0	0	0	0	¯1	0	¯1	0	¯1	0	0	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	0	¯1	¯1	¯1	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

1 16	2 16	3 16	1 17	2 17	3 17	1 18	2 18	3 18	RHS
1000	1400	2400	1500	1100	2600	1000	1100	2100	
¯1	0	0	¯1	0	0	¯1	0	0	0
0	¯1	0	0	¯1	0	0	¯1	0	0
0	0	¯1	0	0	¯1	0	0	¯1	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0	0
0	0	0	1	1	1	0	0	0	0
1	1	1	1	1	1	1	1	1	1

Policy

State		action
1	1	Use both generators
1	2	Use both generators
1	3	Use both generators
2	1	Use both generators
2	2	Use both generators
2	3	Use both generators
3	1	Use both generators
3	2	Use both generators
3	3	Use both generators
4	1	Use both generators
4	2	Use both generators
4	3	Use both generators
5	1	Use both generators
5	2	Use both generators
5	3	Use both generators
6	1	Use both generators
6	2	Use both generators
6	3	Use both generators

The average cost/period in steady state is 2100

# of states: 18
# of actions: 3

Cost Matrix

k	name	1	2	3	4	5	6	7	8	9
1	Use generator 1	1000	1500	1000	1000	1500	1000	1001000	1001500	1001000
2	Use generator 2	1400	1100	1100	1400	1100	1100	1400	1100	1100
3	Use both generators	2400	2600	2100	2400	2600	2100	2400	2600	2100

k	10	11	12	13	14	15	16	17	18
1	1001000	1001500	1001000	1001000	1001500	1001000	1000	1500	1000
2	1001400	1001100	1001100	1001400	1001100	1001100	1400	1100	1100
3	2400	2600	2100	2400	2600	2100	2400	2600	2100

(Rows ~ actions, Columns ~ states)

A value of 999999 above signals
an infeasible action in a state.