Assignment 9

March 25, 2024 2:29 PM

MIE 376 **Assignment 9** Winter 2024

In this assignment, you solve problems for Markov Decision Processes (MDP) - Discounted. You need to formulate the mathematical model and solve it by hand and with Python and Gurobi.

1. **Submission Instructions**

Submit a PDF file describing a Markov Decision Processes (MDP) model and reporting the solution to the problem instance. Also, submit a program (a Python script or a Jupyter notebook) using Gurobi to solve the problem instance. In the MDP formulation, clearly define the stages, states, and actions and formulate the Bellman equations. In the Linear Programming (LP) formulation, clearly define decision variables and state the objective function and constraints. For the problem instance, report the values of the objective and soluions.

2. **Problem**

Question 1: At the beginning of each year, an aircraft engine is in good, fair, or poor condition. It costs $500,000 to run a good engine for a year, $1 million to run a fair engine for a year, and $2 million to run a poor engine for a year. A fair engine can be overhauled for $2 million, and it immediately becomes a good engine. A poor engine can be replaced for $3 million, and it immediately becomes a good engine. The transition probability matrix for an engine is as follows:

	Good	**Fair**	**Poor**
Good	0.7	0.2	0.1
Fair	0.0	0.6	0.4
Poor	0.0	0.0	1.0

Suppose the discount rate $\beta = 0.9$. and our goal is to minimize expected discounted cost.

a) Use the policy iteration method to determine an optimal policy.

b) Use linear programming to determine an optimal policy.

c) Perform two iterations of value iteration.

d) Find a policy that minimizes average cost.

→ SOLUTION BELOW AND ON PYTHON - GUROBI ↓

Question 2: At any time, the size of tree is 0, 1, 2, or 3. We must decide when to harvest the tree. Each year, it costs $1 to maintain the tree. It costs $5 to harvest a tree. The sales price for a tree of each size is as follows:

Tree Size	**Sale Price**
0	$20
1	$30
2	$45
3	$49

The transition probability matrix for the size of the tree is as follows:

	0	**1**	**2**	**3**
0	0.8	0.2	0.0	0.0
1	0.0	0.0	0.9	0.1
2	0.0	0.0	0.7	0.3
3	0.0	0.0	0.0	1.0

For example, 80% of all size 0 trees begin the next year as size 0 trees, and 20% of all size 0 trees begin the next year as size 1 trees. Assuming the discount factor for cash flows is 0.9 per year, determine an optimal harvesting strategy.

a) Use the policy iteration method to determine an optimal policy.

b) Use linear programming to determine an optimal policy.

c) Perform two iterations of value iteration.

d) Find a policy that minimizes average cost.

→ SOLUTION BELOW AND ON PYTHON - GUROBI ↓

Problem 1

a) Policy Iteration

Stages: time period (need to make decision at each stage)

States: $s \in \{good, fair, poor\}$: engine condition

Decision: $x = \{0, 1\}$ to run or overhaul (run = dont overhaul)

General Bellman → 0.9 discount rate

$$V_s = \min_x \{ r_{sx} + \beta \sum_j P_{sjx} V_j \} \quad \forall s$$

↳ transition probabilities given

Bellman Equations (K = thousand)

$V_{good} = \min_x \{ 500K + 0.9(0.7V_{good} + 0.2V_{fair} + 0.1V_{poor}) \}$: no overhaul for good engine.

$V_{fair} = \min_{x \in \{0,1\}} \{ 1M + 0.9(0.6V_{fair} + 0.4V_{poor}),\ 2.5M + 0.9(0.7V_{good} + 0.2V_{fair} + 0.1V_{poor}) \}$

$V_{poor} = \min_{x \in \{0,1\}} \{ 2M + 0.9(1V_{poor}),\ 3.5M + 0.9(0.7V_{good} + 0.2V_{fair} + 0.1V_{poor}) \}$

Action

Initial policy guess : (0,1,1) : ie. $x^*(good) = 0$, $x^*(fair) = 1$, $x^*(poor) = 1$

$V_{good} = 500K + 0.9(0.7V_{good} + 0.2V_{fair} + 0.1V_{poor})$

$V_{Fair} = 2.5M + 0.9(0.7V_{good} + 0.2V_{fair} + 0.1V_{poor})$

$V_{poor} = 3.5M + 0.9(0.7V_{good} + 0.2V_{fair} + 0.1V_{poor})$

Solving gives: (solved using Wolfram eqn solver)

V_{good} = 11300000 (11.3M) (M = million)

V_{fair} = 13300000 (13.3M)

V_{poor} = 14300000 (14.3M)

Solving gives: (solved using Wolfram equations)

$V_{good} = 11300000$ (11.3M) (M = million)
$V_{fair} = 13300000$ (13.3M)
$V_{poor} = 14300000$ (14.3M)

Plug these into Bellman Equations:

$V_{good} = \min_x \{500K + 0.9(0.7(11.3M) + 0.2(13.3M) + 0.1(14.3M))\}$

$= \min_x \{11.3M\} = 11.3M \Rightarrow x^*(good) = 0$

$V_{fair} = \min_{x \in \{0,1\}} \{1M + 0.9(0.6(13.3M) + 0.4(14.3M)), 2.5M + 0.9(0.7(11.3M) + 0.2(13.3M) + 0.1(14.3M))\}$

$= \min_x \{13.3M, 13.3M\} = 13.3M \Rightarrow x^*(fair) = 1$

$V_{poor} = \min_{x \in \{0,1\}} \{2M + 0.9(14.3M), 3.5M + 0.9(0.7(11.3M) + 0.2(13.3M) + 0.1(14.3M))\}$

$= \min_x \{14.87M, 14.3M\} = 14.3M \Rightarrow x^*(poor) = 1$

So policy = (0,1,1), and since the policy does not change from the initial guess, it is an optimal policy.

Optimal policy: run if engine is good,
overhaul if engine is fair or poor

b) LP Formulation

Decision Variables: $V_{good}, V_{fair}, V_{poor}$

Objective Function $i \in \{poor, good, fair\}$

$\max \sum_i V_i$: to ensure atleast one of the V's hit the constraints to be active

Constraints

shadow prices

$V_{good} \leq 500K + 0.9(0.7V_{good} + 0.2V_{fair} + 0.1V_{poor})$ (y_1)

$V_{fair} \leq 1M + 0.9(0.6V_{fair} + 0.4V_{poor})$ (y_2)

$V_{fair} \leq 2.5M + 0.9(0.7V_{good} + 0.2V_{fair} + 0.1V_{poor})$ (y_3)

$V_{poor} \leq 2M + 0.9(1V_{poor})$ (y_4)

$V_{poor} \leq 3.5M + 0.9(0.7V_{good} + 0.2V_{fair} + 0.1V_{poor})$ (y_5)

$V_i \geq 0$, $i \in \{poor, good, fair\}$

writing each of the possibilities in Bellman equations as a constraint.

Problem Instance Solution (using Gurobi)

$V_{good} = \$11.3M$, $V_{fair} = \$13.3M$, $V_{poor} = \$14.3M$: decision variable values.

Maximized objective : $38.9M = $38900000

Shadow prices:

$y_1 = 19.9$, $y_2 = 0$, $y_3 = 6.4$, $y_4 = 0$, $y_5 = 3.7$. : only 1,3,5 tight

i.e. $x^*(good) = 0$, $x^*(fair) = 1$, $x^*(poor) = 1$

So again, optimal policy: (0,1,1)

c) Value Iteration

Let's guess: $V_i^{(0)} = 0$, $i \in \{good, fair, poor\}$: i.e. Policy = $x^*(i) = 0$
(0,0,0)

Plugging into Bellman equations we get:

Iteration 1

$V_{good}^{(1)} = \min_x \{500K + 0.9(0.7(0) + 0.2(0) + 0.1(0))\}$

$= \min_x \{500K\} = 500K \Rightarrow x^*(good) = 0$, $V_{good}^{(1)} = 500K$

$V_{fair}^{(1)} = \min_{x \in \{0,1\}} \{1M + 0.9(0.6(0) + 0.4(0)), 2.5M + 0.9(0.7(0) + 0.2(0) + 0.1(0))\}$

$= \min_x \{1M, 2.5M\} = 1M \Rightarrow x^*(fair) = 0$, $V_{fair}^{(1)} = 1M$

$V_{poor}^{(1)} = \min_{x \in \{0,1\}} \{2M + 0.9(0), 3.5M + 0.9(0.7(0) + 0.2(0) + 0.1(0))\}$

$= \min_x \{2M, 3.5M\} = 2M \Rightarrow x^*(poor) = 0$, $V_{poor}^{(1)} = 2M$

Policy says never overhaul.

Iteration 2

use $V_i^{(1)}$ values in Bellman equations.

$V_{good}^{(2)} = \min_x \{500K + 0.9(0.7(500K) + 0.2(1M) + 0.1(2M))\}$

$x^*(good) = 0$, $V_{good}^{(2)} = 1.175M$

use $V_i^{(1)}$ values in Bellman equations.

$$V_{good}^{(2)} = \min_x \{500K + 0.9(0.7(500K) + 0.2(1M) + 0.1(2M)\}$$
$$= \min_x \{1.175M\} = 1.175M \Rightarrow x^*(good) = 0, \quad V_{good}^{(2)} = 1.175M$$

$$V_{fair}^{(2)} = \min_{x\in\{0,1\}} \{1M + 0.9(0.6(1M) + 0.4(2M)), 2.5M + 0.9(0.7(500K) + 0.2(1M) + 0.1(2M)\}$$
$$= \min_x \{2.26M, 3.175M\} = 3.175M \Rightarrow x^*(fair) = 0, \quad V_{fair}^{(2)} = 2.26M$$

$$V_{poor}^{(2)} = \min_{x\in\{0,1\}} \{2M + 0.9(2M), 3.5M + 0.9(0.7(500K) + 0.2(1M) + 0.1(2M)\}$$
$$= \min_x \{3.8M, 4.175M\} = 3.8M \Rightarrow x^*(poor) = 0, \quad V_{poor}^{(2)} = 3.8M$$

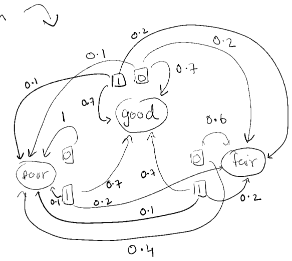
New policy : $x^*(i) = 0$, i.e. don't overhaul ever
$x = (0,0,0) \rightarrow$ hasn't converged yet

If we keep doing this, it will eventually converge to the policy iteration solution.

d) Average Cost Minimization

First, we draw state diagram

$\boxed{1}$ = overhaul, $\boxed{0}$ = keep running



General Avg MDP Formulation

$$\min \sum_s \sum_x r_{sx} \pi_{sx}$$

$$\sum_s \sum_x \pi_{sx} = 1, \quad \pi_{sx} \geq 0$$

$$\sum_x \pi_{sx} = \sum_x \sum_i \pi_{ix} P_{isx} \quad \forall s$$

↳ transition state probabilities : see diagram.

In our case: let π_{sx} be the state and choice

state $\in \{good, fair, poor\}$ (g) (f) (p); choice $\in \{r, 0\}$ (r = run, 0 = overhaul)

Decision variables : to understand which states form optimal policy.

r_{sx} = Fixed cost of running engine (even after overhaul cost)

$r_{gr} = 500K$, $r_{go} = 500K$, $r_{fr} = 1M$, $r_{fo} = 2.5M$
$r_{pr} = 2M$, $r_{po} = 3.5M$.

Objective Function

$$\min \; 500K\pi_{gr} + 500K\pi_{go} \; 1M\pi_{fr} + 2.5M\pi_{fo} + 2M\pi_{pr} + 3.5M\pi_{po}$$

Constraints

$$\pi_{gr} + \pi_{go} + \pi_{fr} + \pi_{fo} + \pi_{pr} + \pi_{po} = 1$$

$$\pi_{fr} + \pi_{fo} = 0.2\pi_{gr} + 0.6\pi_{fr} + 0.2\pi_{fo} + 0.2\pi_{go} + 0.2\pi_{po}$$
$$\pi_{pr} + \pi_{po} = 0.1\pi_{gr} + 0.4\pi_{fr} + \pi_{pr} + 0.1\pi_{po} + 0.1\pi_{fo} + 0.1\pi_{go}$$

Prob of being in state = prob of going to state

$$\pi_{gr}, \pi_{go}, \pi_{fr}, \pi_{fo}, \pi_{pr}, \pi_{po} \geq 0$$

$$\pi_{gr} + \pi_{go} = 0.7\pi_{gr} + 0.7\pi_{go} + 0.7\pi_{fo} + 0.7\pi_{po}$$ Not needed extra constraint.

Problem Instance Solution (using Gurobi)

$\pi_{gr} = 0.7$
$\pi_{go} = 0$
$\pi_{fr} = 0$
$\pi_{fo} = 0.2$
$\pi_{pr} = 0$
$\pi_{po} = 0.1$

Decision variable values.

optimal policy must include states that take values

so $\pi_{gr}, \pi_{fo}, \pi_{po}$

Thus, $x^* = (0,1,1)$

Objective maximized: \$1.2M = \$1200000 ~ avg cost.

optimal policy = $\{0,1,1\}$

i.e overhaul if engine fair or poor and run if good.

Problem 2

a) Policy Iteration

Stages: time period (need to make decision at each stage)

States: tree size, $s \in \{0,1,2,3\}$

Decision: $x = \{0,1\}$ to maintain or harvest

Assumptions: after harvesting tree goes to size 0, start time at $t = 0^+$ (so already maintained)

General Bellman → 0.9 discount rate

$V_s = \max_x \{ r_{sx} + \beta \sum_j P_{sjx} V_j \} \ \forall s$ (maximize profit) ↳ transition probabilities given

Bellman Equations

$$V_0 = \max_{x\in\{0,1\}} \{-1 + 0.9(0.8V_0 + 0.2V_1), (20-5) + 0.9(0.8V_0 + 0.2V_1)\}$$

$$V_1 = \max_{x\in\{0,1\}} \{-1 + 0.9(0.9V_2 + 0.1V_3), (30-5) + 0.9(0.8V_0 + 0.2V_1)\}$$

$$V_2 = \max_{x\in\{0,1\}} \{-1 + 0.9(0.7V_2 + 0.3V_3), (45-5) + 0.9(0.8V_0 + 0.2V_1)\}$$

$$V_3 = \max_{x\in\{0,1\}} \{-1 + 0.9(V_3), (49-5) + 0.9(0.8V_0 + 0.2V_1)\}$$

Action

Initial policy guess: (1 1 1 1) : ie. always harvest, ie. $x^*(i) = 0$

$V_0 = 15 + 0.9(0.8V_0 + 0.2V_1)$

$V_1 = 25 + 0.9(0.8V_0 + 0.2V_1)$

$V_2 = 40 + 0.9(0.8V_0 + 0.2V_1)$

$V_3 = 44 + 0.9(0.8V_0 + 0.2V_1)$

Solving gives: (solved using Wolfram eqn solver)

$V_0 = 168$, $V_1 = 178$, $V_2 = 193$, $V_3 = 197$

Plug these into Bellman Equations:

$$V_0 = \max_{x\in\{0,1\}} \{-1 + 0.9(0.8(168) + 0.2(178)), (20-5) + 0.9(0.8(168) + 0.2(178))\}$$
$$= \max_{x\in\{0,1\}} \{152, 168\} = 168 \Rightarrow x^*(V_0) = 1$$

$$V_1 = \max_{x\in\{0,1\}} \{-1 + 0.9(0.9(193) + 0.1(197)), (30-5) + 0.9(0.8(168) + 0.2(178))\}$$
$$= \max_{x\in\{0,1\}} \{173.06, 178\} = 178 \Rightarrow x^*(V_1) = 1$$

$$V_2 = \max_{x\in\{0,1\}} \{-1 + 0.9(0.7(193) + 0.3(197)), (45-5) + 0.9(0.8(168) + 0.2(178))\}$$
$$= \max_{x\in\{0,1\}} \{173.78, 193\} = 193 \Rightarrow x^*(V_2) = 1$$

$$V_3 = \max_{x\in\{0,1\}} \{-1 + 0.9(197), (49-5) + 0.9(0.8(168) + 0.2(178))\}$$
$$= \max_{x\in\{0,1\}} \{176.3, 197\} = 197 \Rightarrow x^*(V_3) = 1$$

So policy = (1,1,1,1), and since the policy does not change from the initial guess, it is an optimal policy.

Optimal policy: always harvest the tree, regardless of stage.

b) LP Formulation

Decision Variables: V_0, V_1, V_2, V_3

Objective Function — to ensure atleast one of the V's hit the constraints to be active

$\text{Min} \sum_i V_i$, $i \in \{0,1,2,3\}$

Constraints

$V_0 \geq -1 + 0.9(0.8V_0 + 0.2V_1)$ (y_1) → shadow prices — writing each of the possibilities

Constraints

shadow prices

$V_0 \geq -1 + 0.9(0.8V_0 + 0.2V_1)$ (y_1)

$V_0 \geq (20-5) + 0.9(0.8V_0 + 0.2V_1)$ (y_2)

$V_1 \geq -1 + 0.9(0.9V_2 + 0.1V_3)$ (y_3)

$V_1 \geq (30-5) + 0.9(0.8V_0 + 0.2V_1)$ (y_4)

$V_2 \geq -1 + 0.9(0.7V_2 + 0.3V_3)$ (y_5)

$V_2 \geq (45-5) + 0.9(0.8V_0 + 0.2V_1)$ (y_6)

$V_3 \geq -1 + 0.9(V_3)$ (y_7)

$V_3 \geq (49-5) + 0.9(0.8V_0 + 0.2V_1)$ (y_8)

$V_i \geq 0, \; i \in \{0,1,2,3\}$

writing each of the possibilities in Bellman equations as a constraint.

Problem Instance Solution (using Gurobi)

$V_0 = \$168, \; V_1 = \$178, \; V_2 = \$193, \; V_3 = \$197$: decision variable values.

Minimized objective : $\$736$

Shadow prices :

$y_1 = 0, \; y_2 = 29.8, \; y_3 = 0, \; y_4 = 8.2, \; y_5 = 0, \; y_6 = 1, \; y_7 = 0, \; y_8 = 1$: only 2, 4, 6, 8 are tight

i.e. $x^*(V_0) = 1, \; x^*(V_1) = 1, \; x^*(V_2) = 1, \; x^*(V_3) = 1$

So again, optimal policy : $(1,1,1,1)$

c) Value Iteration

Let's guess: $V_i^{(0)} = 0, \quad i \in \{0,1,2,3\}$: i.e. Policy $= x^*(i) = 0$
$(0,0,0,0)$

Plugging into Bellman equations we get :

Iteration 1

$V_0^{(1)} = \max_{x \in \{0,1\}} \{-1 + 0.9(0.8(0) + 0.2(0)), (20-5) + 0.9(0.8(0) + 0.2(0))\}$

$= \max_{x \in \{0,1\}} \{-1, 15\} = 15 \Rightarrow x^*(V_0) = 1, \; V_0^{(1)} = 15$

$V_1^{(1)} = \max_{x \in \{0,1\}} \{-1 + 0.9(0.9(0) + 0.1(0)), (30-5) + 0.9(0.8(0) + 0.2(0))\}$

$= \max_{x \in \{0,1\}} \{-1, 25\} = 25 \Rightarrow x^*(V_1) = 1, \; V_1^{(1)} = 25$

$V_2^{(1)} = \max_{x \in \{0,1\}} \{-1 + 0.9(0.7(0) + 0.3(0)), (45-5) + 0.9(0.8(0) + 0.2(0))\}$

$= \max_{x \in \{0,1\}} \{-1, 40\} = 40 \Rightarrow x^*(V_2) = 1, \; V_2^{(1)} = 40$

$V_3^{(1)} = \max_{x \in \{0,1\}} \{-1 + 0.9(0), (49-5) + 0.9(0.8(0) + 0.2(0))\}$

$= \max_{x \in \{0,1\}} \{-1, 44\} = 44 \Rightarrow x^*(V_3) = 1, \; V_3^{(1)} = 44$

policy says always harvest

Iteration 2

use $V_i^{(1)}$ values in Bellman equations.

$V_0^{(2)} = \max_{x \in \{0,1\}} \{-1 + 0.9(0.8(15) + 0.2(25)), (20-5) + 0.9(0.8(15) + 0.2(25))\}$

$= \max_{x \in \{0,1\}} \{14.3, 30.3\} = 30.3 \Rightarrow x^*(V_0) = 1, \; V_0^{(2)} = 30.3$

$V_1^{(2)} = \max_{x \in \{0,1\}} \{-1 + 0.9(0.9(40) + 0.1(44)), (30-5) + 0.9(0.8(15) + 0.2(25))\}$

$= \max_{x \in \{0,1\}} \{35.36, 40.3\} = 40.3 \Rightarrow x^*(V_1) = 1, \; V_1^{(2)} = 40.3$

$V_2^{(2)} = \max_{x \in \{0,1\}} \{-1 + 0.9(0.7(40) + 0.3(44)), (45-5) + 0.9(0.8(15) + 0.2(25))\}$

$= \max_{x \in \{0,1\}} \{36.08, 55.3\} = 55.3 \Rightarrow x^*(V_2) = 1, \; V_2^{(2)} = 55.3$

$V_3^{(2)} = \max_{x \in \{0,1\}} \{-1 + 0.9(44), (49-5) + 0.9(0.8(15) + 0.2(25))\}$

$= \max_{x \in \{0,1\}} \{38.6, 59.3\} = 59.3 \Rightarrow x^*(V_3) = 1, \; V_3^{(2)} = 59.3$

New policy : $x^*(i) = 1$, i.e. always harvest

$x \in \{0,1,3\}$

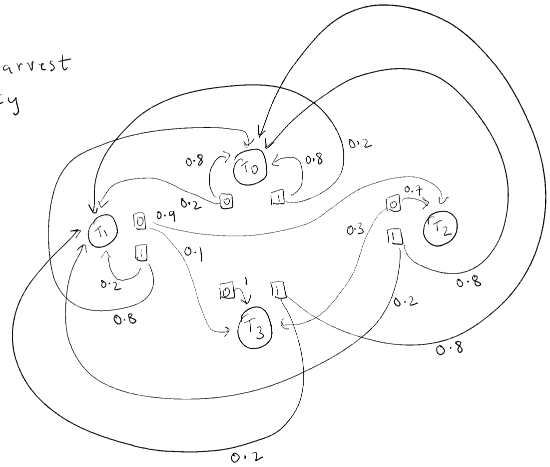
New policy : $x^*(i) = 1$, ie. always harvest

$x = (1,1,1,1) \rightarrow$ same policy

d) Average Cost Minimization

First, we draw state diagram

$\boxed{1}$ = harvest, $\boxed{0}$ = maintain.



General Avg MDP Formulation

$$\min \sum_s \sum_x r_{sx} \pi_{sx}$$

$$\sum_s \sum_x \pi_{sx} = 1 \quad , \pi_{sx} \geq 0$$

$$\sum_x \pi_{sx} = \sum_x \sum_i \pi_{ix} P_{isx} \quad \forall s$$

↳ transition state probabilities : see diagram

In our case : let π_{sx} be the state and choice

↳ $\in \{0,1,2,3\}$ ↳ $\in \{M, H\}$ (maintain, harvest)

Decision variables : to understand which states form optimal policy.

r_{sx} = Fixed cost of the sale (even after harvesting cost)

$r_{0M} = -1$, $r_{0H} = 15$, $r_{1M} = -1$, $r_{1H} = 25$

$r_{2M} = -1$, $r_{2H} = 40$, $r_{3M} = -1$, $r_{3H} = 44$

Objective Function

$$\max -\pi_{0M} + 15\pi_{0H} - \pi_{1M} + 25\pi_{1H} - \pi_{2M} + 40\pi_{2H} - \pi_{3M} + 44\pi_{3H}$$

Constraints

$$\pi_{0M} + \pi_{0H} + \pi_{1M} + \pi_{1H} + \pi_{2M} + \pi_{2H} + \pi_{3M} + \pi_{3H} = 1$$

$$\pi_{1M} + \pi_{1H} = 0.2\pi_{0M} + 0.2\pi_{2H} + 0.2\pi_{3H} + 0.2\pi_{0H} + 0.2\pi_{1M}$$

$$\pi_{2M} + \pi_{2H} = 0.9\pi_{1M} + 0.7\pi_{2M}$$

$$\pi_{3M} + \pi_{3H} = 0.1\pi_{1M} + 0.3\pi_{2M} + \pi_{3M}$$

Prob of being in state = prob of going to state

$$\pi_{0M}, \pi_{0H}, \pi_{1M}, \pi_{1H}, \pi_{2M}, \pi_{2H}, \pi_{3M}, \pi_{3H} \geq 0$$

$$\pi_{0M} + \pi_{0H} = 0.8\pi_{0M} + 0.8\pi_{0H} + 0.8\pi_{1H} + 0.8\pi_{2H} + 0.8\pi_{3H}$$

Not needed extra constraint.

Problem Instance Solution (using Gurobi)

$\pi_{0M} = 0$

$\pi_{0H} = 0.8$

$\pi_{1M} = 0$

$\pi_{1H} = 0.2$

$\pi_{2M} = 0$

$\pi_{2H} = 0$

$\pi_{3M} = 0$

$\pi_{3H} = 0$

Decision variable values.

optimal policy must include states that take values

so π_{0H}, π_{1H}

thus $x^* = (1,1)$

Objective maximized : \$17 ~ avgcost.

optimal policy = $\{1,1\}$

ie harvest the tree when it is in stage 0 or 1, we never get to stage 2 or 3, the tree gets cut before that.