Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Institute for Dynamic Systems and Control
IDSC
Institut für Dynamische Systeme und Regelungstechnik

Final Exam **January 26th, 2012**

Dynamic Programming & Optimal Control (151-0563-01)

Prof. R. D'Andrea

Solutions

Exam Duration: **150 minutes**

Number of Problems: **4**

Permitted aids: One A4 sheet of paper.

Use only the provided sheets for your solutions.

Problem 1 **25%**

Consider the following simplified map of Romania that is represented as a directed graph[1].

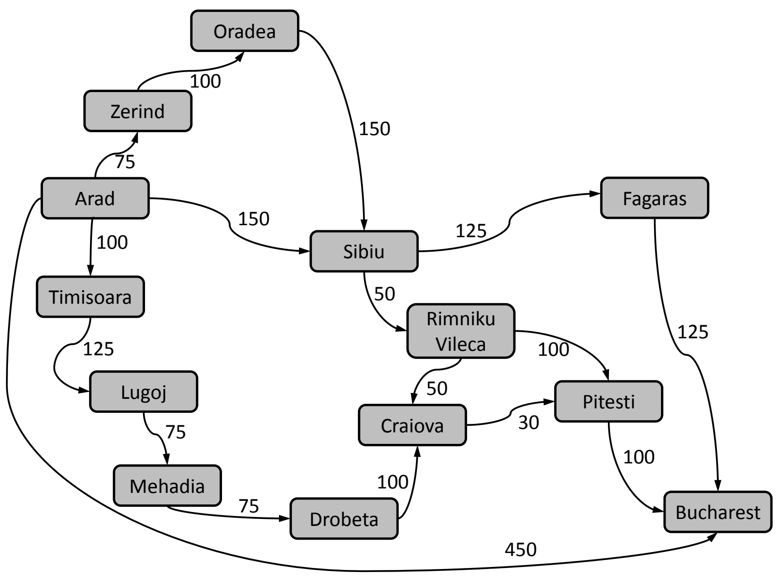

Figure 1: Simplified map of Romania

City	Straight-line dist.
Arad	350
Bucharest	0
Craiova	100
Dobreta	150
Fagaras	100
Lugoj	200
Mehadia	175
Oradea	400
Pitesti	50
Rim. Vil.	150
Sibiu	200
Timisoara	325
Zerind	350

Figure 2: Straight-line distances to Bucharest

Find the shortest path from Arad to Bucharest for the graph given in Figure 1 by applying the A*-Algorithm. Use the straight-line distances from Figure 2 as heuristics and the best-first method to determine, at each iteration, which node to remove from the `OPEN` bin; that is, remove node i with

$$d_i = \min_{j \text{ in } \mathtt{OPEN}} d_j$$

where the variable d_i denotes the length of the shortest path from Arad to node i that has been found so far.

Solve the problem by populating a table of the following form[2]:

Iteration	Node exiting `OPEN`	`OPEN`	d_{Arad}	d_{Zerind}	d_{Oradea}	$d_{Timisoara}$	...	$d_{Bucharest}$
0	-	...						
1	Arad	...						
...								

State the resulting shortest path and its associated cost.

[1]Though not a realistic model, a directed graph was chosen to reduce the computational burden.
[2]Please use the paper in landscape orientation for the table.

Solution 1

Iter.	Node exiting OPEN	OPEN	d_{Arad}	$d_{Zer.}$	$d_{Orad.}$	$d_{Timi.}$	d_{Lugoj}	$d_{Meh.}$	$d_{Drob.}$	$d_{Cra.}$	d_{Sibiu}	$d_{Rim.Vil.}$	$d_{Fag.}$	$d_{Pit.}$	$d_{Buch.}$
0	-	Arad	0	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
1	Arad	Zerind, Timis., Sibiu	0	75	∞	100	∞	∞	∞	∞	150	∞	∞	∞	450
2*	Zerind	Timis., Sibiu	0	75	∞	100	∞	∞	∞	∞	150	∞	∞	∞	450
3	Timis.	Sibiu, Lugoj	0	75	∞	100	225	∞	∞	∞	150	∞	∞	∞	450
4	Sibiu	Lugoj, Fagaras, Rim.Vil.	0	75	∞	100	225	∞	∞	∞	150	200	275	∞	450
5	Rim.Vil.	Lugoj, Fagaras, Craiova, Pitesti	0	75	∞	100	225	∞	∞	250	150	200	275	300	450
6*	Lugoj	Fagaras, Craiova, Pitesti	0	75	∞	100	225	∞	∞	250	150	200	275	300	450
7	Craiova	Fagaras, Pitesti	0	75	∞	100	225	∞	∞	250	150	200	275	280	450
8	Fagaras	Pitesti	0	75	∞	100	225	∞	∞	250	150	200	275	280	400
9	Pitesti	-	0	75	∞	100	225	∞	∞	250	150	200	275	280	380

Optimal path: Arad $\mapsto$ Sibiu $\mapsto$ Rim.Vil. $\mapsto$ Craiova $\mapsto$ Pitesti $\mapsto$ Bucharest
Cost: 380
* Heuristics prevent a city from entering `OPEN`.

Problem 2 **25%**

Consider the dynamic system

$$\dot{x}(t) = x(t) + u(t)$$

with the initial state $x(0) = x_0$ and $t \in [0, T]$, $T \in \mathbb{R}^+$.

Use the Minimum Principle to find the optimal input $u^*(t)$ that minimizes the following cost function

$$\frac{1}{2}x^2(T) + \frac{1}{2}\int_0^T u^2(t)dt$$

and the corresponding optimal trajectory $x^*(t)$.

Solution 2

Applying the Minimum Principle:

The system equation is

$$\dot{x}(t) = x(t) + u(t). \tag{1}$$

The Hamiltonian is given by

$$\begin{aligned} H(x(t), u(t), p(t)) &= g(x(t), u(t)) + p(t) f(x(t), u(t)) \\ &= \frac{1}{2}u^2(t) + p(t)x(t) + p(t)u(t). \end{aligned}$$

The adjoint equation can be calculated as follows

$$\dot{p}(t) = -\frac{\partial H}{\partial x}(x(t), u(t), p(t)) = -p(t),$$

with the boundary condition

$$p(T) = \nabla h(x(T)) = x(T).$$

Solving this differential equation leads to

$$p(t) = \zeta e^{-t}, \quad \zeta = x(T)e^{T} \tag{2}$$

The optimal input $u^*(t)$ is obtained by minimizing the Hamiltonian along the optimal trajectory

$$\begin{aligned} & u^*(t) = \underset{u}{\operatorname{argmin}}\left(\frac{1}{2}u^2(t) + p(t)x(t) + p(t)u(t)\right) \\ & \Rightarrow u^*(t) = -p(t). \end{aligned} \tag{3}$$

Now, (1), (2), and (3) give

$$\dot{x}(t) = x(t) - \zeta e^{-t}, \quad x(0) = x_0.$$

The general solution of the above inhomogeneous ordinary differential equation is the sum of the homogeneous solution $x_h(t) = \lambda e^t$ and a particular solution $x_p(t) = \frac{1}{2}\zeta e^{-t}$ giving

$$x(t) = x_h(t) + x_p(t) = \lambda e^t + \frac{1}{2}\zeta e^{-t}.$$

The above general solution with the initial condition $x(0) = x_0$, and $\zeta = x(T)e^T$ gives

$$\lambda = \frac{x_0}{1 + e^{2T}} \quad \text{and} \quad \zeta = \frac{2x_0 e^{2T}}{1 + e^{2T}}.$$

This implies

$$u^*(t) = -\frac{2x_0 e^{2T}}{1 + e^{2T}} e^{-t}$$

and

$$x^*(t) = \frac{x_0}{1 + e^{2T}} e^t + \frac{x_0 e^{2T}}{1 + e^{2T}} e^{-t}.$$

Problem 3 **25%**

Consider the stochastic shortest path problem shown in Figure 3

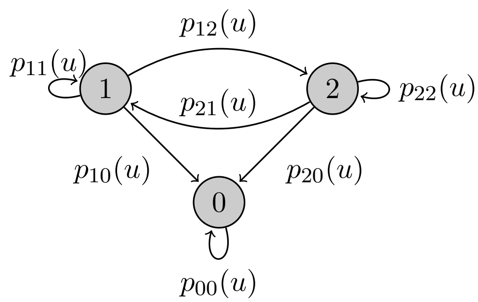

Figure 3: Transition graph of the stochastic shortest path problem

with the control sets

$$U(0) = \{a_0\}$$
$$U(1) = \{a_1, b_1, c_1\}$$
$$U(2) = \{a_2, b_2\},$$

the transition probabilities

$p_{10}(a_1) = 1/3$	$p_{20}(a_2) = 0$	$p_{00}(a_0) = 1$
$p_{11}(a_1) = 1/3$	$p_{21}(a_2) = 2/3$	
$p_{12}(a_1) = 1/3$	$p_{22}(a_2) = 1/3$	
$p_{10}(b_1) = 1/3$	$p_{20}(b_2) = 1/3$	
$p_{11}(b_1) = 2/3$	$p_{21}(b_2) = 1/3$	
$p_{12}(b_1) = 0$	$p_{22}(b_2) = 1/3$	
$p_{10}(c_1) = 0$		
$p_{11}(c_1) = 2/3$		
$p_{12}(c_1) = 1/3$		

and the cost function

$$g(i, \mu(i)) = 1 \quad i = 1, 2 \text{ and } \mu(i) \in U(i)$$
$$g(0, a_0) = 0.$$

a) Using policy iteration, perform 2 iterations for the given problem. Start with evaluating the initial policies $\mu^0(1) = a_1$ and $\mu^0(2) = a_2$.

b) Has the policy iteration converged after 2 iterations? If so, please explain why. If it has not converged, please explain the criterion that it would need to fulfill for convergence. Also, if it has not converged, can you comment on the possible outcome of iteration 3?

Solution 3

a) From the given information we can see that node 0 is the cost free terminal node. Therefore we do not need to consider it.

Iteration 1:
Policy evaluation:

$$\begin{aligned}
J^1(1) &= 1 + p_{11}(a_1)J^1(1) + p_{12}(a_1)J^1(2)\\
&= 1 + \frac{1}{3}J^1(1) + \frac{1}{3}J^1(2)\\
&\Rightarrow J^1(1) = \frac{3}{2} + \frac{1}{2}J^1(2) \qquad (4)\\
J^1(2) &= 1 + p_{21}(a_2)J^1(1) + p_{22}(a_2)J^1(2)\\
&= 1 + \frac{2}{3}J^1(1) + \frac{1}{3}J^1(2)\\
&\overset{\text{using (4)}}{=} 1 + \frac{2}{3}\left(\frac{3}{2} + \frac{1}{2}J^1(2)\right) + \frac{1}{3}J^1(2)\\
&= 2 + \frac{2}{3}J^1(2)\\
&\Rightarrow J^1(2) = 6,\ J^1(1) = 9/2
\end{aligned}$$

Policy improvement:

$$\begin{aligned}
\mu^1(1) &= \underset{u\in U(1)}{\operatorname{argmin}} \big[1 + p_{11}(a_1)J^1(1) + p_{12}(a_1)J^1(2),\\
&\qquad 1 + p_{11}(b_1)J^1(1) + p_{12}(b_1)J^1(2),\\
&\qquad 1 + p_{11}(c_1)J^1(1) + p_{12}(c_1)J^1(2)\big]\\
&= \underset{u\in U(1)}{\operatorname{argmin}} [1 + 1/3\cdot 9/2 + 1/3\cdot 6, 1 + 2/3\cdot 9/2 + 0\cdot 6, 1 + 2/3\cdot 9/2 + 1/3\cdot 6]\\
&= \underset{u\in U(1)}{\operatorname{argmin}} [9/2, 4, 6]\\
\mu^1(1) &= b_1\\
\mu^1(2) &= \underset{u\in U(2)}{\operatorname{argmin}} \big[1 + p_{21}(a_2)J^1(1) + p_{22}(a_2)J^1(2),\\
&\qquad 1 + p_{21}(b_2)J^1(1) + p_{22}(b_2)J^1(2)\big]\\
&= \underset{u\in U(2)}{\operatorname{argmin}} [1 + 2/3\cdot 9/2 + 1/3\cdot 6, 1 + 1/3\cdot 9/2 + 1/3\cdot 6]\\
&= \underset{u\in U(2)}{\operatorname{argmin}} [6, 9/2]\\
\mu^1(2) &= b_2
\end{aligned}$$

Iteration 2:
Policy evaluation:

$$\begin{aligned} J^2(1) &= 1 + \frac{2}{3}J^2(1) + 0 * J^2(2) \\ &\Rightarrow J^2(1) = 3 \\ J^2(2) &= 1 + \frac{1}{3}J^2(1) + \frac{1}{3}J^2(2) = 2 + \frac{1}{3}J^2(2) \\ &\Rightarrow J^2(2) = 3 \end{aligned}$$

Policy improvement:

$$\begin{aligned} \mu^2(1) &= \underset{u \in U(1)}{\operatorname{argmin}} \left[1 + 1/3 \cdot 3 + 1/3 \cdot 3, 1 + 2/3 \cdot 3, 1 + 2/3 \cdot 3 + 1/3 \cdot 3\right] \\ &= \underset{u \in U(1)}{\operatorname{argmin}} \left[3, 3, 4\right] \\ \mu^2(1) &= a_1 \text{ or } b_1 \\ \mu^2(2) &= \underset{u \in U(2)}{\operatorname{argmin}} \left[1 + 2/3 \cdot 3 + 1/3 \cdot 3, 1 + 1/3 \cdot 3 + 1/3 \cdot 3\right] \\ &= \underset{u \in U(2)}{\operatorname{argmin}} \left[4, 3\right] \\ \mu^2(2) &= b_2 \end{aligned}$$

b) Policy iteration has converged, if $J^k(i) = J^{k-1}(i)$ holds for all nodes i at iteration k. This is clearly not the case for $k = 2$. Therefore, policy iteration has not converged yet.

Even though policy iteration has not converged yet, we can still comment on the policy that the iteration will converge to.

Let's first pick $\mu^2(1) = b_1$ for iteration 3. In this case we apply the same combination of inputs as in iteration 2 and the transition probabilities in the policy evaluation step of iteration 3 will be the same. Therefore it will yield the same costs as in iteration 2, i.e., $J^2(i) = J^3(i)$ for all nodes i. Hence, policy iteration has converged to the policy $\mu(1) = b_1$ and $\mu(2) = b_2$.

Now, if we pick $\mu^2(1) = a_1$ for iteration 3 the only thing that we can tell about the convergence is that this choice will also eventually lead to convergence. But we cannot say after how many iterations or to which policy.

Additional information:
In fact, if you do iteration 3 with $\mu^2(1) = a_1$ you would also see that $J^2(i) = J^3(i)$ holds for all nodes i. Therefore, both choices for $\mu^2(1)$ are feasible outcomes of the policy iteration.

Problem 4 **25%**

A burglar broke into a house and found $N \in \mathbb{Z}^+$ items[3]. Let $v_i > 0$ denote the value and $w_i \in \mathbb{Z}^+$ the weight of the i^{th} item. There is a limit, $W \in \mathbb{Z}^+$, on the total weight the burglar can carry (the total weight of all stolen items must be less than or equal to W) and obviously he wants to maximize the total value of the items that he can take.

a) Case A: A burglar who did not take the Dynamic Programming and Optimal Control class

How many different combinations of items does the burglar have to consider to find the combination with the highest value that he can take?

b) Case B: A burglar who did take the Dynamic Programming and Optimal Control class

b.1) Formulate the burglar's problem by defining the state space, control space, system dynamics, stage cost and terminal cost.

b.2) State the Dynamic Programming Algorithm that is required to solve the burglar's problem.

b.3) Using the Dynamic Programming Algorithm from b.2, solve the burglar's problem where $N = 3$, $W = 5$, $v_1 = 6$, $v_2 = 10$, $v_3 = 12$, $w_1 = 1$, $w_2 = 2$ and $w_3 = 3$.

c) Briefly compare Case A and Case B in terms of computational cost.

[3] $\mathbb{Z}^+$ denotes the set of positive integers

Solution 4

a) 2^N (Cardinality of the power set)

b) b.1) Let x_k, $k > 1$, be the total weight of the items the burglar has decided to take from the set of $\{1, .., k-1\}$ items and $x_1 = 0$. Furthermore, let $u_k \in \{0, 1\}$ be the binary decision variable that controls if item k goes into the burglar's bag or not.

State space $S_k \in [0, W] \subset \{0, \mathbb{Z}^+\}$, $k = 1, \ldots, N+1$.
Control space U_k: if $x_k + w_k u_k > W$, $U_k = \{0\}$ else, $U_k = \{0, 1\}$, $\quad k = 1, \ldots, N$.
System dynamics: $x_{k+1} = x_k + w_k u_k, \quad x_1 = 0, \quad k = 1, \ldots, N$.
Stage cost: $g_k(x_k, u_k) = -v_k u_k, \quad k = 1, \ldots, N$.
Terminal cost: $g_{N+1}(x_{N+1}) = 0, \quad \forall x_{N+1} \in S_{N+1}$.

b.2) Now applying the Dynamic Programming Algorithm (DPA) gives

$$\begin{aligned} J_{N+1}(x_{N+1}) &= g_{N+1}(x_{N+1}) = 0, \quad \forall x_{N+1} \in S_{N+1} \\ J_k(x_k) &= \min_{u_k} \{-v_k u_k + J_{k+1}(x_{k+1})\}, \quad k = 1, \ldots, N. \end{aligned}$$

b.3) Applying the above DPA for $N = 3$, $W = 5$, $v_1 = 6$, $v_2 = 10$, $v_3 = 12$, $w_1 = 1$, $w_2 = 2$ and $w_3 = 3$, gives:
$J_4(x_4) = 0, \quad \forall x_4 \in \{0, 1, 2, 3, 4, 5\}$
$\underline{k = 3}$

$$\begin{aligned} J_3(0) &= \min\{-12 \cdot 0 + J_4(0 + 0 \cdot 3), -12 \cdot 1 + J_4(0 + 1 \cdot 3)\} = -12, \quad \mu_3(0) = 1 \\ J_3(1) &= \min\{-12 \cdot 0 + J_4(1 + 0 \cdot 3), -12 \cdot 1 + J_4(1 + 1 \cdot 3)\} = -12, \quad \mu_3(1) = 1 \\ J_3(2) &= \min\{-12 \cdot 0 + J_4(2 + 0 \cdot 3), -12 \cdot 1 + J_4(2 + 1 \cdot 3)\} = -12, \quad \underline{\mu_3(2) = 1} \\ J_3(3) &= \min\{-12 \cdot 0 + J_4(3 + 0 \cdot 3)\} = 0, \quad \mu_3(3) = 0 \\ J_3(4) &= \min\{-12 \cdot 0 + J_4(4 + 0 \cdot 3)\} = 0, \quad \mu_3(4) = 0 \\ J_3(5) &= \min\{-12 \cdot 0 + J_4(5 + 0 \cdot 3)\} = 0, \quad \mu_3(5) = 0 \end{aligned}$$

$\underline{k = 2}$

$$\begin{aligned} J_2(0) &= \min\{-10 \cdot 0 + J_3(0 + 0 \cdot 2), -10 \cdot 1 + J_3(0 + 1 \cdot 2)\} = -22, \quad \underline{\mu_2(0) = 1} \\ J_2(1) &= \min\{-10 \cdot 0 + J_3(1 + 0 \cdot 2), -10 \cdot 1 + J_3(1 + 1 \cdot 2)\} = -12, \quad \mu_2(1) = 0 \\ J_2(2) &= \min\{-10 \cdot 0 + J_3(2 + 0 \cdot 2), -10 \cdot 1 + J_3(2 + 1 \cdot 2)\} = -12, \quad \mu_2(2) = 0 \\ J_2(3) &= \min\{-10 \cdot 0 + J_3(3 + 0 \cdot 2) - 10 \cdot 1 + J_3(3 + 1 \cdot 2)\} = -10, \quad \mu_2(3) = 1 \\ J_2(4) &= \min\{-10 \cdot 0 + J_3(4 + 0 \cdot 2)\} = 0, \quad \mu_2(4) = 0 \\ J_2(5) &= \min\{-10 \cdot 0 + J_3(5 + 0 \cdot 2)\} = 0, \quad \mu_2(5) = 0 \end{aligned}$$

$\underline{k = 1}$

$$J_1(0) = \min\{-6 \cdot 0 + J_2(0 + 0 \cdot 1), -6 \cdot 1 + J_2(0 + 1 \cdot 1)\} = -22, \quad \underline{\mu_1(0) = 0}.$$

Therefore, optimal solution is to leave the first item and take the second and third items.

c) Case A has a time complexity $O(2^N)$ which is exponential in N and Case B has a time complexity $O(WN)$ which is linear in N. Therefore, for large N Case B (DPA) is computationally less expensive than Case A (Brute force approach).