ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

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

151-0563-01 **Dynamic Programming and Optimal Control** (Fall 2012)

Programming Exercise #2 Topic: Infinite Horizon Problems

Issued: Nov 14, 2012 Due: Nov 28, 2012

Mohanarajah Gajamohan (gajan@ethz.ch) , 15. November 2012

Policy Iteration, Value Iteration, and Linear Programming

Tobi manages two locations of a nationwide car rental company. Each day, between 10:00 and 18:00, some number of customers arrive at each location to rent cars. If Tobi has a car available, he rents it out and is credited g_r CHF, $g_r > 0$, by the national company. If he is out of cars at that location, then he cannot rent a car out and the business is lost. All car returns happen early in the morning, between 07:00 and 10:00, before any renting begins. To help ensure that cars are available where they are needed, Tobi can move them between the two locations overnight, at a cost of g_m CHF, $g_m > 0$, per car moved.
Due to the limited availability of parking space, there can be no more than $n_{\max}$ cars at each location (any additional cars are returned to the nationwide company) and a maximum of $u_{\max}$ cars can be moved from one location to the other in one night.
We assume that the number of cars requested and returned at each location are Poisson random variables, meaning that the probability of the number being $n \in \mathbb{N}_0$ is given by $e^{-\lambda}\lambda^n/n!$, where $\lambda > 0$ is the Poisson average. Let $\lambda_{1,r}$ and $\lambda_{2,r}$ denote the Poisson averages for rental requests at the first and second location and $\lambda_{1,d}$ and $\lambda_{2,d}$ for drop-offs.
Your task is to help Tobi come up with the optimal policy to move cars between locations in order to maximize his gains. Note that Tobi, influenced by his banking buddies, discounts his future gain: the future gain in k days is discounted by α^k, where $\alpha < 1$ is the discount factor. Coming from a country that is known for its engineering precision and accuracy, Tobi wants to calculate the optimal policy and the corresponding optimal cost using *all* the following methods: just to be sure.

(i) Policy Iteration

(ii) Value Iteration

(iii) Linear Programming

Hints

- You may use the following state representation: $s := (n_1, n_2)$, where n_1 and n_2 are the number of cars *at the end of the day* (18:00) in locations one and two.
- The suppressed stage cost $g(i, u)$ can be calculated using (Bertsekas, pg. 405)

  $$g(i, u) = \sum_j p_{ij}(u)\tilde{g}(i, u, j),$$

  where $i, j \in \{(n_1, n_2) : n_1, n_2 \in \{0, 1, \ldots, n_{\max}\}\}$.

Deliverables

Please hand in by e-mail

- Your MATLAB implementation of the following function[1]:

  `[policy, cost] = progEx2Solver(n_max, lambda, g, u_max, alpha, opt)`

  Inputs

  - `n_max`: maximum number of cars each location can hold
  - `lambda`: Poisson averages for rental and drop-offs for both locations, $[\lambda_{1,r}, \lambda_{2,r}, \lambda_{1,d}, \lambda_{2,d}] \in \mathbb{R}_+^{1\times 4}$
  - `g`: associated costs $[g_r, g_m] \in \mathbb{R}^{1\times 2}$
  - `u_max`: maximum number of cars Tobi can move between the two locations
  - `alpha`: discount factor
  - `opt`: algorithm option `P`, `V`, `L` of type char; `P`-Policy Iteration, `V`-Value Iteration, and `L`-Linear Programming[2]

  You may use the following input values during implementation: `n_max=8`, `lambda=[3,4,3,2]`, `g=[10,2]`, `alpha=0.9`, and `u_max=3`. Note that, during evaluation, we test your function for different input values.

  Outputs

  - `policy`: $(n_{\max} + 1)$ sized square matrix where rows, indexed from $n_1 = 0$ to $n_{\max}$, represent the number of cars in the first location and similarly columns represents the cars in the second location. The matrix elements represents the corresponding optimal policy $\mu^*([n_1, n_2]) \in [-u_{\max}, u_{\max}]$, which is the number of cars to move from location one to location two. $\mu^* < 0$ implies a move in the opposite direction.
  - `cost`: $(n_{\max} + 1)$ sized square matrix similar to `policy`, but the elements represents the optimal cost $J^*([n_1, n_2])$

- In a PDF file, show using plots or verbally, how the optimal policy and the cost will change during the following scenarios:

  - Tobi's good friend Michael, works late near the first location and lives near the second one. If Tobi wants to move cars from the first location to the second, Michael will take one car for free. This means Tobi will save g_m CHF.
  - Due to economic fluctuations, the interest rate goes down.

Please include all files into one `zip`-archive, named `DPOCEx2_Names.zip`, where `Names` is a list of the full names of all students who have worked on the solution.
(e.g `DPOCEx2_HuebelNico_OungRaymond_TrimpeSebastian.zip`)[3]
Send your file to `gajan@ethz.ch` until the due date indicated above. We will send a confirmation e-mail upon receiving your e-mail. You are ultimately responsible that we receive your solution in time.

Plagiarism

When handing in any piece of work, the student (or, in case of a group work, each individual student) listed as author confirms that the work is original, has been done by the author(s) independently and that s/he has read and understood the *ETH Citation etiquette* http://www.ethz.ch/students/exams/plagiarism_s_en.pdf. Each work submitted will be tested for plagiarism.

[1] Strictly follow the structure. Grading of this part is automated.
[2] Use the `linprog` function of MATLAB's Optimization Toolbox to solve the linear program.
[3] Up to three students are allowed to work together on the problem. They will all receive the same grade.