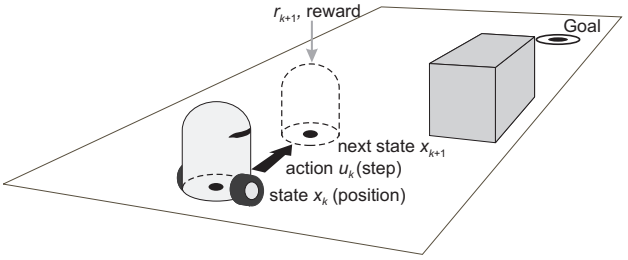
# Corrections to
# *Reinforcement learning and dynamic programming using function approximators*

Lucian Buşoniu, Robert Babuška, Bart De Schutter, and Damien Ernst
Taylor & Francis CRC Press 2010

Updated July 8, 2011

### Figure 1.3 (page 3)

The figure omits the goal and obstacle mentioned in the text (these elements are shown later, in Figure 1.5). The correct figure is given here:




### Example 2.3 (page 26) and Example 2.4 (page 34)

The results in Table 2.2 are not obtained with the version of Q-iteration from Algorithm 2.1, as stated in the text, but with an asynchronous version that employs the most recently updated Q-values at each step of the computation. This version replaces lines 3–5 of Algorithm 2.1 with the procedure:

---

$Q \leftarrow Q_\ell$
**for** every $(x,u)$ **do**
  $Q(x,u) \leftarrow \rho(x,u) + \gamma \max_{u'} Q(f(x,u),u')$
**end for**
$Q_{\ell+1} \leftarrow Q$

---

Similarly, Tables 2.3, 2.5, and 2.6 are produced using asynchronous versions of, respectively, Algorithms 2.2, 2.5, and 2.6. These versions are easily obtained by modifications similar to the one above, and are not given here. The computational cost considerations and comparisons in the examples remain valid, but apply to the asynchronous algorithm variants. (Since the synchronous variants given in the book are less efficient, they would run in a larger number of iterations.)

**Equation 3.50 (page 89):** $\widehat{\widehat{Q}^{\widehat{h}_\ell}}$ should be $\widehat{Q}^{\widehat{h}_\ell}$

**Section 4.5.4 (page 160): Car on the hill example**

The terminal states were incorrectly handled in this example. In particular, the usual fuzzy Q-iteration updates:

$$\theta_{\ell+1,[i,j]} \leftarrow \rho(x_i,u_j) + \gamma \max_{j'} \sum_{i'=1}^{N} \phi_{i'}(f(x_i,u_j))\theta_{\ell,[i',j']}$$

were applied even when the next state $f(x_i,u_j)$ was terminal, i.e., outside the domain $[-1,1]\times[-3,3]$. This, however, corresponds to – incorrectly – assigning non-zero rewards to the terminal states. These rewards then propagate through the updates and lead to overly large Q-values. The correct way to perform the updates is by explicitly enforcing a zero Q-value in any terminal state:

$$\theta_{\ell+1,[i,j]} \leftarrow \rho(x_i,u_j) + \begin{cases} \gamma \max_{j'} \sum_{i'=1}^{N} \phi_{i'}(f(x_i,u_j))\theta_{\ell,[i',j']} & \text{if } f(x_i,u_j) \text{ is non-terminal} \\ 0 & \text{if } f(x_i,u_j) \text{ is terminal} \end{cases}$$

The following results change due to this modification. Figure 4.14(b) changes to:

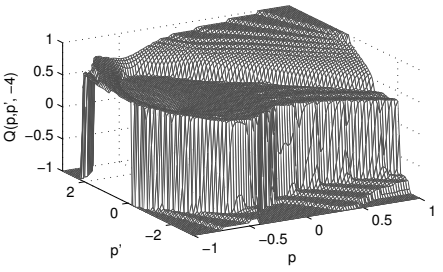
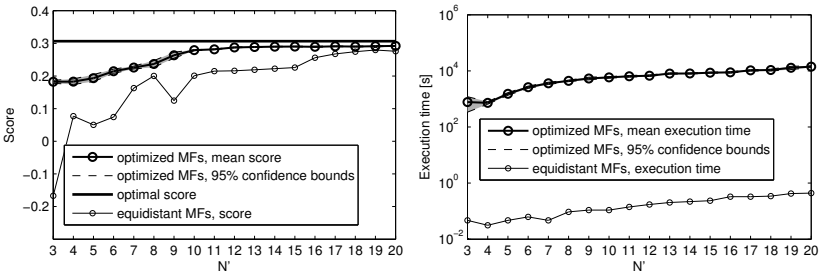
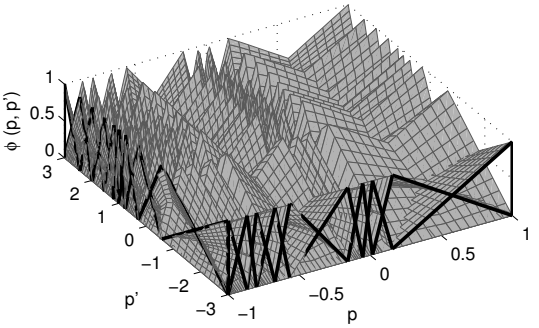



Figure 4.15 changes to:




(a) Performance. (b) Execution time.

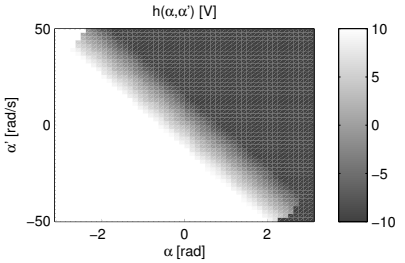
Figure 4.16 changes to:




The original discussion remains entirely valid, and these results are qualitatively similar to the original ones. This is because the policies computed by fuzzy Q-iteration remain almost unaffected by the change.

## Page 196, line 5: $\gamma = 0.95$ should be $\gamma = 0.98$

Note that the policy in Figure 5.13(b) is near-optimal for $\gamma = 0.95$. Nevertheless, the corresponding near-optimal policy for $\gamma = 0.98$:




has the same structure, and the considerations after Figure 5.13 remain valid.