INFO8003-1: Reinforcement Learning in a discrete domain

†**Guillaume Delporte**[1] and †**Andréas Coco**[2]

[1] *guillaume.delporte@student.uliege.be (s191981)*
[2] *andreas.coco@student.uliege.be (s2302246)*

† These authors contributed equally to this work.

Course taught by [1]**Damien Ernst**

[1] *dernst@uliege.be*

CONTENTS

I. SECTION 1: IMPLEMENTATION OF THE DOMAIN

For the first part of the project, we had to implement the components of the domain that was provided in the instructions. This domain then had to be used to simulate a policy we designed via a trajectory of 10 steps. We decided to simulate a stationary policy that chooses at random between the four actions, no matter the state. That is, the policy is given by Equation 1.

$$\mu(s) = \begin{cases} (0,1) & \text{with probability } \frac{1}{4} \\ (0,-1) & \text{with probability } \frac{1}{4} \\ (1,0) & \text{with probability } \frac{1}{4} \\ (-1,0) & \text{with probability } \frac{1}{4} \end{cases} \tag{1}$$

The resulting simulated trajectories for the deterministic and the stochastic domain are respectively displayed in Table I and Table II. Note that throughout this entire report, all numbers are rounded to the third decimal.

t	s_t	a_t	r_t	s_{t+1}
0	(3, 0)	(1, 0)	-20	(4, 0)
1	(4, 0)	(0, 1)	-17	(4, 1)
2	(4, 1)	(0, 1)	-4	(4, 2)
3	(4, 2)	(0, -1)	-17	(4, 1)
4	(4, 1)	(1, 0)	-17	(4, 1)
5	(4, 1)	(1, 0)	-17	(4, 1)
6	(4, 1)	(0, 1)	-4	(4, 2)
7	(4, 2)	(-1, 0)	4	(3, 2)
8	(3, 2)	(0, 1)	19	(3, 3)
9	(3, 3)	(0, 1)	-5	(3, 4)
10	(3, 4)	(0, 1)	-5	(3, 4)

TABLE I: Simulated trajectory in the deterministic domain

t	s_t	a_t	r_t	s_{t+1}
0	(3, 0)	(0, 1)	-9	(3, 1)
1	(3, 1)	(0, -1)	-3	(0, 0)
2	(0, 0)	(0, 1)	1	(0, 1)
3	(0, 1)	(-1, 0)	1	(0, 1)
4	(0, 1)	(-1, 0)	-3	(0, 0)
5	(0, 0)	(0, -1)	-3	(0, 0)
6	(0, 0)	(1, 0)	-3	(0, 0)
7	(0, 0)	(-1, 0)	-3	(0, 0)
8	(0, 0)	(1, 0)	6	(1, 0)
9	(1, 0)	(0, 1)	3	(1, 1)
10	(1, 1)	(0, -1)	6	(1, 0)

TABLE II: Simulated trajectory in the stochastic domain

II. SECTION 2: EXPECTED RETURN OF A POLICY

As a next step, we had to design a method to estimate J^μ and test it for the policy introduced in the previous section in the given domain. As Equation 2 provides us with a bound for the discrepancy between J^μ and J^μ_N, we decided to use $N = 10,000$ to approximate J^μ. Given the discount factor $\gamma = 0.99$ and that the rewards of the domain being considered are bounded above by $B_r = 19$,

Equation 2 ensures the accuracy of the estimation to be bounded by:
$\|J^{\mu} - J^{\mu}_{10000}\|_{\infty} \leq \frac{0.99^{10000}}{0.01} 19 \approx 10^{-40}$

$$\|J^{\mu} - J^{\mu}_{N}\|_{\infty} \leq \frac{\gamma^{N}}{1-\gamma} B_r \tag{2}$$

The estimations obtained for all states for the both domains are given in Table III

	$J^{\mu}_{N}(s)$	
s	Deterministic	Stochastic
(0,0)	87.400	-125.512
(0,1)	95.245	-126.047
(0,2)	116.144	-124.283
(0,3)	144.607	-120.400
(0,4)	167.538	-116.020
(1,0)	82.076	-124.008
(1,1)	90.082	-124.511
(1,2)	109.231	-123.447
(1,3)	132.750	-121.292
(1,4)	148.753	-119.414
(2,0)	60.950	-124.996
(2,1)	70.344	-125.549
(2,2)	91.249	-124.410
(2,3)	114.580	-121.900
(2,4)	121.679	-124.368
(3,0)	23.803	-130.439
(3,1)	38.907	-129.480
(3,2)	69.478	-125.323
(3,3)	93.031	-125.058
(3,4)	108.640	-123.124
(4,0)	-9.306	-136.790
(4,1)	8.726	-135.668
(4,2)	47.433	-129.700
(4,3)	86.213	-123.021
(4,4)	100.448	-123.374

TABLE III: States expected return approximations, based on random policy

III. SECTION 3: OPTIMAL POLICY

In this section, we had to implement routines that derive the Markov Decision Process (MDP) corresponding to the domain that we defined. We then had to use that MDP to compute the sequence of Q_N functions obtained using Equation 3 and find the lowest N such that the policy inferred from the Q-functions after it would always remain the same. We found this value of N to be 7, both for the deterministic and for the stochastic domain. To discover this, we computed Q_N for values of N up to 1,000 and kept record of when the policy changed. The optimal policies derived are shown in Figures 1 and 2. The raw tables describing these policies can be found in Table Va in the Appendix.

$$Q_N(s,a) = r(s,a) + \gamma \sum_{s' \in S} p(s'|s,a) \max_{a' \in A} Q_{N-1}(s',a'), \quad \forall N \geq 1 \tag{3}$$

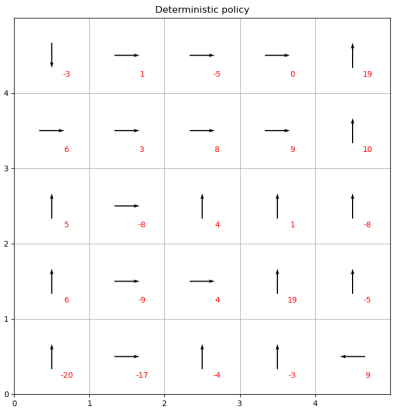

FIG. 1: Optimal policy derived based on the Q-function computed using dynamic programming in the deterministic setting

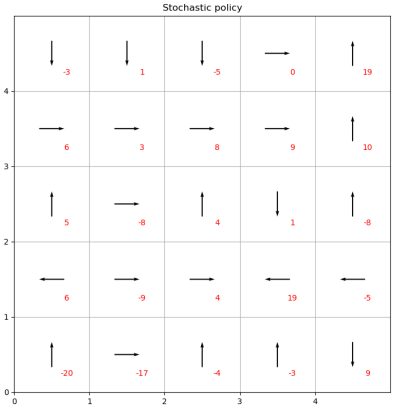

FIG. 2: Optimal policy derived based on the Q-function computed using dynamic programming in the stochastic setting

Using the optimal policy just derived, we can approximate J^{μ^*} using $J^{\mu^*}_{10,000}$, similarly to what we did in Section 2. The results obtained are exhibited in Table IV. By comparing these numbers with those obtained using our random policy in Section 2, it can be observed that, un-

surprisingly, the optimal policy yields much higher state expected returns.

	$J_N^\mu(s)$	
s	Deterministic	Stochastic
(0,0)	1842.031	159.446
(0,1)	1857.190	159.637
(0,2)	1881.000	163.052
(0,3)	1900.000	172.130
(0,4)	1900.000	172.130
(1,0)	1854.576	159.637
(1,1)	1870.279	163.052
(1,2)	1881.090	164.903
(1,3)	1891.000	167.630
(1,4)	1900.000	172.130
(2,0)	1842.031	159.446
(2,1)	1855.576	160.137
(2,2)	1870.279	163.052
(2,3)	1881.090	167.213
(2,4)	1891.000	167.630
(3,0)	1828.610	159.259
(3,1)	1849.010	162.196
(3,2)	1863.646	167.213
(3,3)	1863.279	162.196
(3,4)	1864.090	167.213
(4,0)	1816.324	159.259
(4,1)	1826.520	155.713
(4,2)	1849.010	162.196
(4,3)	1863.646	167.213
(4,4)	1842.010	162.229

TABLE IV: States expected return approximations, based on optimal policy

IV. SECTION 4: SYSTEM IDENTIFICATION

A. Deterministic Domain

In the deterministic domain, things happen in a predictable way. If we do the same action in the same state several times, the outcome will always be the same. When our trajectory gets longer and we observe more system transitions, our guesses about the system's rules get better and closer to the real rules. The plots show that our guesses are getting more accurate as we look at more examples. The tables with numbers show us that our guesses for the rewards and transition probabilities are very close to the real values, especially for long trajectories. This means that our method works well when the system dynamics and rewards are predictable. The policy we get from our guesses tells us the best actions to take, and it gets better as our system estimation improves. The policy we derived using the longest sequence we considered is shown in Figure 3.

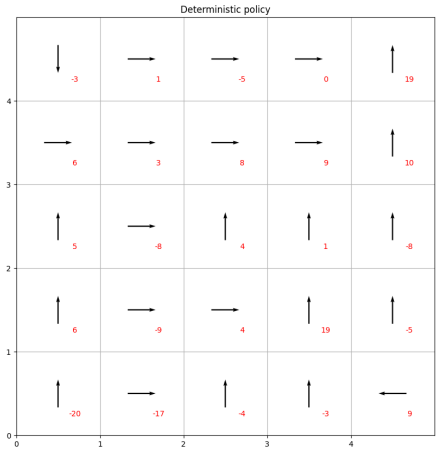

FIG. 3: Optimal policy derived based on the estimated Q-function computed using dynamic programming in the deterministic setting

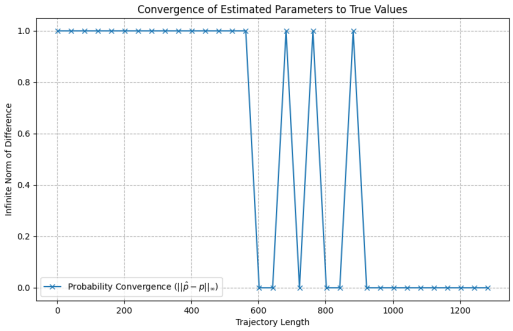

FIG. 4: Illustration of the convergence process for estimated transition probabilities ($\hat{p}$) towards the true probabilities in the deterministic domain, as measured by the infinite norm.

B. Stochastic Domain

In the stochastic domain, system transitions are unpredictable. Repeating the same action in a given state multiple times can lead to different outcomes. As a result, our method requires to visit the same state-action pair more times to correctly estimate the system's rules. However, since our trajectory starts in position (3, 0), with a 0.5 chance of going to (0, 0) at every step and the trajectory used to generate the trajectory is random, some state-action pairs (almost) never get

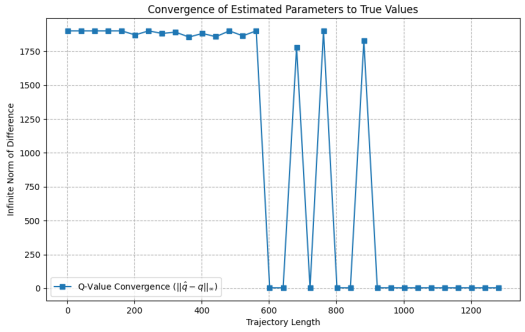

FIG. 5: Graph showing how the estimated Q-values ($\hat{Q}$) approach the true Q-values in the deterministic domain, with convergence quantified using the infinite norm.

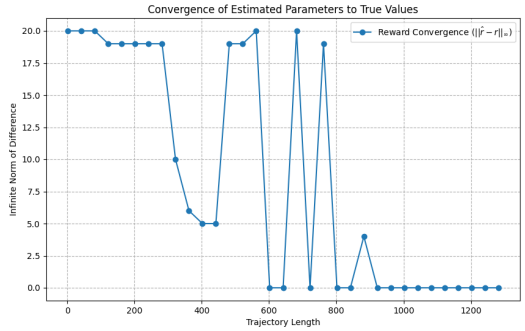

FIG. 6: Depiction of the convergence trajectory of the estimated rewards ($\hat{r}$) to the actual rewards in the deterministic domain, as captured by the infinite norm.

visited. For example, state (0, 4) is very unlikely to get visited with a random policy in the stochastic domain. As a consequence, even when we observe many state transitions, our guesses don't match reality. The plots and tables show that there's a difference between our guesses and the real values, and it doesn't get much better even with more examples.

Our method tries to figure out the best actions to take, but because of the absence of evidence for some (s, a) values, it's not as successful as with the deterministic domain. The randomness makes it hard for our method to appropriately determine the system dynamics and rewards to then find the best choices to make in each state. The policy we derived in the stochastic case is given in Figure 7. It is clearly not optimal as this policy suggests going down in state (0,4), when the optimal choice is clearly to stay in that cell.

Our observations can be even more easy to see when takin g in consideration the table of estimated Q functions, where we can see that for some state action pairs, we do not have any observations and are prompted with a 0. VII

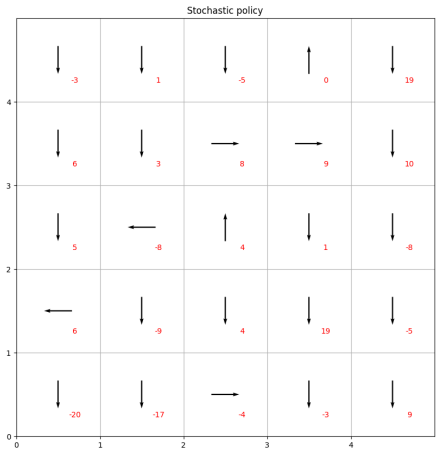

FIG. 7: Optimal policy derived based on the estimated Q-function computed using dynamic programming in the deterministic setting

The estimated expected return tables can be found in the Appendix. The graphs representing the convergence of $\hat{r}$, $\hat{Q}$ and $\hat{p}$ in infinite norm are respectively displayed in Figures 8, 9 and 10.

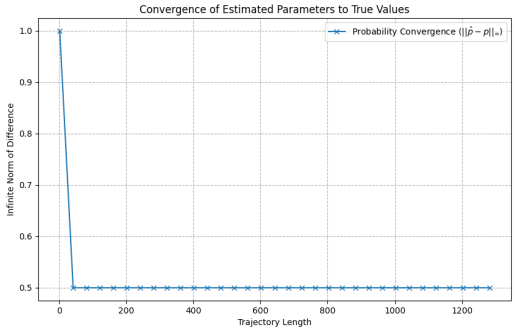

FIG. 8: Visualization of the estimated transition probabilities ($\hat{p}$) convergence towards the true probabilities within the stochastic domain, evaluated using the infinite norm.

V. SECTION 5: Q-LEARNING IN A BATCH SETTING

A. Offline Q-Learning

B. Online Q-Learning

C. Discount Factor

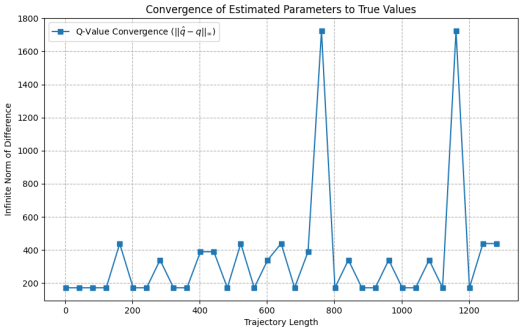

FIG. 9: Graphical representation of the convergence behavior of estimated Q-values ($\hat{Q}$) towards the actual Q-values in the stochastic domain, measured by the infinite norm.

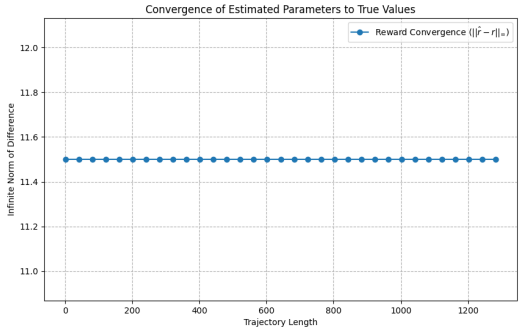

FIG. 10: Chart depicting the convergence process of estimated rewards ($\hat{r}$) to the true rewards in the stochastic domain, as quantified by the infinite norm.

APPENDIX

D. Tables

	$\mu^*(s)$	
s	Deterministic	Stochastic
(0, 0)	(1, 0)	(1, 0)
(0, 1)	(0, 1)	(1, 0)
(0, 2)	(0, 1)	(1, 0)
(0, 3)	(0, 1)	(0, 1)
(0, 4)	(-1, 0)	(-1, 0)
(1, 0)	(0,1)	(0, 1)
(1, 1)	(0,1)	(0, 1)
(1, 2)	(0,1)	(0, 1)
(1, 3)	(0,1)	(0, 1)
(1, 4)	(-1, 0)	(-1, 0)
(2, 0)	(-1, 0)	(-1, 0)
(2, 1)	(0,1)	(0, 1)
(2, 2)	(-1, 0)	(-1, 0)
(2, 3)	(-1, 0)	(1, 0)
(2, 4)	(-1, 0)	(-1, 0)
(3, 0)	(-1, 0)	(0, -1)
(3, 1)	(0,1)	(0, 1)
(3, 2)	(0,1)	(0, 1)
(3, 3)	(-1, 0)	(0, -1)
(3, 4)	(-1, 0)	(0, -1)
(4, 0)	(-1, 0)	(-1, 0)
(4, 1)	(0,1)	(0, 1)
(4, 2)	(-1, 0)	(-1, 0)
(4, 3)	(-1, 0)	(-1, 0)
(4, 4)	(0, -1)	(1, 0)

(a) Optimal policies based on Q-values derived using dynamic programming for the third section of the homework

	$\hat{J}^*(s)$	
s	**Deterministic**	**Stochastic**
(0,0)	1842.031	-46.392
(0,1)	1857.190	-48.656
(0,2)	1881.000	-44.756
(0,3)	1900.000	-51.413
(0,4)	1900.000	-48.443
(1,0)	1854.576	-45.485
(1,1)	1870.279	-48.873
(1,2)	1881.090	-47.686
(1,3)	1891.000	-49.943
(1,4)	1900.000	-48.443
(2,0)	1842.031	-44.753
(2,1)	1855.576	-44.116
(2,2)	1870.279	-44.068
(2,3)	1881.090	-48.443
(2,4)	1891.000	-48.443
(3,0)	1828.610	-42.503
(3,1)	1849.010	-48.443
(3,2)	1863.646	-51.186
(3,3)	1863.279	-48.443
(3,4)	1864.090	-48.443
(4,0)	1816.324	-48.443
(4,1)	1826.520	-48.443
(4,2)	1849.010	-49.943
(4,3)	1863.646	-48.443
(4,4)	1842.010	-48.443

(b) Expected returns for states in both deterministic and stochastic domains, based on the estimated optimal policy.

State (s)	**Action (a)**	$\hat{Q}$**(s, a) Deterministic**	$\hat{Q}$**(s, a) Stochastic**
(0, 0)	(1, 0)	1837.461	551.048
(0, 0)	(-1, 0)	1816.041	542.523
(0, 0)	(0, 1)	1835.049	540.940
(0, 0)	(0, -1)	1816.041	542.523
(0, 1)	(1, 0)	1850.007	544.185
(0, 1)	(-1, 0)	1835.049	541.325
(0, 1)	(0, 1)	1852.620	537.942
(0, 1)	(0, -1)	1816.041	542.523
(0, 2)	(1, 0)	1865.710	546.305
(0, 2)	(-1, 0)	1852.620	538.059
(0, 2)	(0, 1)	1876.430	541.435
(0, 2)	(0, -1)	1835.049	541.507
(0, 3)	(1, 0)	1876.520	0.000
(0, 3)	(-1, 0)	1876.430	542.523
(0, 3)	(0, 1)	1895.430	542.523
(0, 3)	(0, -1)	1852.620	0.000
(0, 4)	(1, 0)	1886.430	0.000
(0, 4)	(-1, 0)	1895.430	0.000
(0, 4)	(0, 1)	1895.430	0.000
(0, 4)	(0, -1)	1876.430	0.000
(1, 0)	(1, 0)	1824.041	560.482
(1, 0)	(-1, 0)	1816.041	542.523
(1, 0)	(0, 1)	1850.007	544.388
(1, 0)	(0, -1)	1837.461	554.193
(1, 1)	(1, 0)	1824.451	548.345
(1, 1)	(-1, 0)	1835.049	541.126
(1, 1)	(0, 1)	1865.710	544.924
(1, 1)	(0, -1)	1837.461	547.108
(1, 2)	(1, 0)	1851.007	543.943
(1, 2)	(-1, 0)	1852.620	538.059
(1, 2)	(0, 1)	1876.520	544.303
(1, 2)	(0, -1)	1850.007	542.523
(1, 3)	(1, 0)	1858.710	0.000
(1, 3)	(-1, 0)	1876.430	0.000
(1, 3)	(0, 1)	1886.430	542.523
(1, 3)	(0, -1)	1865.710	0.000
(1, 4)	(1, 0)	1859.520	0.000
(1, 4)	(-1, 0)	1895.430	0.000
(1, 4)	(0, 1)	1886.430	0.000
(1, 4)	(0, -1)	1876.520	0.000
(2, 0)	(1, 0)	1811.755	570.540
(2, 0)	(-1, 0)	1837.461	549.576
(2, 0)	(0, 1)	1824.451	549.315
(2, 0)	(0, -1)	1824.041	556.171
(2, 1)	(1, 0)	1816.950	542.523
(2, 1)	(-1, 0)	1850.007	545.847
(2, 1)	(0, 1)	1851.007	544.227
(2, 1)	(0, -1)	1824.041	569.820
(2, 2)	(1, 0)	1844.440	531.738
(2, 2)	(-1, 0)	1865.710	546.846

TABLE VI: Q function values for deterministic and stochastic scenarios

State (s)	**Action (a)**	$\hat{Q}$**(s, a) Deterministic**	$\hat{Q}$**(s, a) Stochastic**
(2, 2)	(0, 1)	1858.710	542.523
(2, 2)	(0, -1)	1824.451	542.523
(2, 3)	(1, 0)	1859.077	0.000
(2, 3)	(-1, 0)	1876.520	0.000
(2, 3)	(0, 1)	1859.520	0.000
(2, 3)	(0, -1)	1851.007	0.000
(2, 4)	(1, 0)	1835.880	0.000
(2, 4)	(-1, 0)	1886.430	0.000
(2, 4)	(0, 1)	1859.520	0.000
(2, 4)	(0, -1)	1858.710	0.000
(3, 0)	(1, 0)	1773.591	542.523
(3, 0)	(-1, 0)	1824.041	569.820
(3, 0)	(0, 1)	1816.950	542.523
(3, 0)	(0, -1)	1811.755	598.557
(3, 1)	(1, 0)	1786.685	0.000
(3, 1)	(-1, 0)	1824.451	0.000
(3, 1)	(0, 1)	1844.440	0.000
(3, 1)	(0, -1)	1811.755	0.000
(3, 2)	(1, 0)	1821.950	533.083
(3, 2)	(-1, 0)	1851.007	0.000
(3, 2)	(0, 1)	1859.077	0.000
(3, 2)	(0, -1)	1816.950	0.000
(3, 3)	(1, 0)	1837.440	0.000
(3, 3)	(-1, 0)	1858.710	0.000
(3, 3)	(0, 1)	1835.880	0.000
(3, 3)	(0, -1)	1844.440	0.000
(3, 4)	(1, 0)	1828.020	0.000
(3, 4)	(-1, 0)	1859.520	0.000
(3, 4)	(0, 1)	1835.880	0.000
(3, 4)	(0, -1)	1859.077	0.000
(4, 0)	(1, 0)	1773.591	0.000
(4, 0)	(-1, 0)	1811.755	0.000
(4, 0)	(0, 1)	1786.685	0.000
(4, 0)	(0, -1)	1773.591	0.000
(4, 1)	(1, 0)	1786.685	0.000
(4, 1)	(-1, 0)	1816.950	0.000
(4, 1)	(0, 1)	1821.950	0.000
(4, 1)	(0, -1)	1773.591	0.000
(4, 2)	(1, 0)	1821.950	0.000
(4, 2)	(-1, 0)	1844.440	0.000
(4, 2)	(0, 1)	1837.440	542.523
(4, 2)	(0, -1)	1786.685	0.000
(4, 3)	(1, 0)	1837.440	0.000
(4, 3)	(-1, 0)	1859.077	0.000
(4, 3)	(0, 1)	1828.020	0.000
(4, 3)	(0, -1)	1821.950	0.000
(4, 4)	(1, 0)	1828.020	0.000
(4, 4)	(-1, 0)	1835.880	0.000
(4, 4)	(0, 1)	1828.020	0.000
(4, 4)	(0, -1)	1837.440	0.000

TABLE VII: Q function values for deterministic and stochastic scenarios cont´d)