Introduction to Numerical Optimization – Project: Optimal Control using Linear Programming

Iver Småge, Alessandro Pase

November 2023

1 Introduction

This report aims to present the solution of an Optimal Control problem using Linear Programming (LP). Two problems will be presented in this report: the first represents the movement of a taxi to serve customers, while the second is a simplified version of the card game blackjack. The problems will be solved using mathematical programming, calculating the optimal actions for the taxi driver to take a customer to his destination and the optimal cards to play to become a blackjack millionaire. Both environments are modelled with an infinite-time discrete Markov Decision Process (MDP), as expressed in the statement.

2 Linear Models

1) Formalization of the primal problem with Linear Programming

One can start from the given Eq. 4 in the statement and expand the expected value:

$$\min(1-\gamma) \mathop{\mathbb{E}}_{s\sim P_0(\cdot))}[V(s)] = \min(1-\gamma) \sum_{s\in S} P_0(s)V(s)$$

where:

- $P_0(s)$ is the probability of starting from the state s, which in our case is uniform;
- $V(s)$ is the value function of a state s.

To analyze the constraint we can proceed in the same way. Knowing that the reward does not depend on s':

$$V(s) \geqslant \underset{s' \sim P(\cdot|s,a))}{\mathbb{E}} \left[R(s,a) + \gamma V\left(s'\right)\right] = R(s,a) + \sum_{s' \in S} P^a_{ss'} \gamma V\left(s'\right) \quad \forall s, a$$

Therefore the problem can be expressed as:

$$\begin{aligned} &\min (1-\gamma) \sum_{s \in S} P_0(s) V(s) \\ &\text{s.t. } V(s) \geqslant R(s,a) + \sum_{s' \in S} P^a_{ss'} \gamma V\left(s'\right) \quad \forall s, a \end{aligned}$$

where $P^a_{ss'}$ is the probability given action a and start state s of moving into the next state s'.
Using the fact that a discrete function can be expressed as a vector (or a matrix), we get the matrix form for the constraints:

$$\mathbf{V} \geqslant \mathbf{R}^a + \gamma \mathbf{P}^a \mathbf{V} \quad \forall a$$

The matrix $\mathbf{P}^a$ multiplied by the unit vector representing a given state $\mathbf{e_s}$ gives the probability vector of moving into all other states given the action a.
Therefore if we define the vector $\mathbf{x} = \mathbf{V}$ we get:

$$\mathbf{x} \geqslant \mathbf{R}^a + \gamma \mathbf{P}^a \mathbf{x} \quad \forall a$$

Which lead to:

$$\left(\mathbb{I} - \gamma \mathbf{P}^a\right) \mathbf{x} \geqslant \mathbf{R}^a \quad \forall a$$

Therefore, the problem in standard form can be expressed as:

$$\begin{aligned} &\min \mathbf{c}^T \mathbf{x} \\ &\text{s.t } \mathbf{A}\mathbf{x} \geqslant \mathbf{b} \\ &\text{with } \mathbf{x} \text{ free} \end{aligned}$$

where:

- $\mathbf{c}^T = (1-\gamma)\left[P_0\left(s_1\right), \ldots, P_0\left(s_n\right)\right]$ with $s_1, \ldots, s_n$ every state
- $\mathbf{x} = \left[V\left(s_1\right), \ldots, V\left(s_n\right)\right]^T$ value function on every state
- $\mathbf{A} = \left(\mathbb{I} - \gamma \mathbf{P}^a\right) \quad \forall a$ with $\mathbf{P}^a =:$ transition matrix given an action a
- $\mathbf{b} = \mathbf{R}^a = \left[R^a\left(s_1\right), \ldots, R^a\left(s_n\right)\right]^T \quad \forall a$

This formulation is expressed with a fixed action a, if we want to generalize the matrix $\mathbf{A}$ expressed as it is in the Gurobi model is:

$$a_{ij} = \delta_{i \bmod \dim(\mathcal{S}),\, j} - \gamma P^{\lfloor i/\dim(\mathcal{S}) \rfloor}_{i \bmod \dim(\mathcal{S}),\, j}$$

where:

- a_{ij} is the element i, j of the matrix $\mathbf{A}$;
- $P^a_{l,m}$ is the probability of going from state l to state m, given the action a

From the matrix all columns corresponding to infeasible states and all rows corresponding to infeasible actions are set to zero. An action is infeasible if it moves from or to an infeasible state or moves outside the map.

This formulation is consistent with the matrix $\mathbf{A}$ in our Gurobi model. In Fig. 1 is presented the matrix $\mathbf{A}$ for the taxi-problem, which has 36 columns which are the number of possible states and 148 rows which are the number of possible actions.

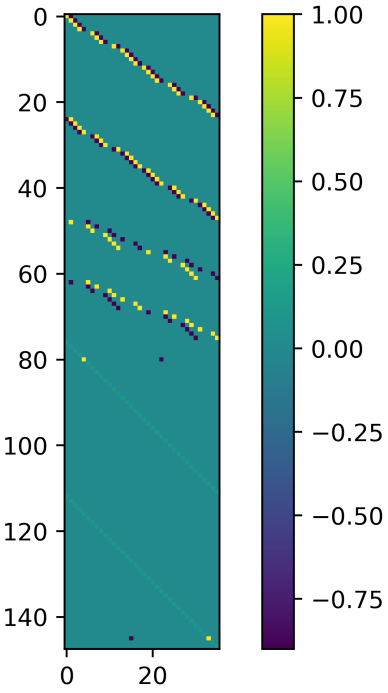

Figure 1: Representation of matrix $\mathbf{A}$ for the taxi problem

2) The dual problem formulation and interpretation

Starting from the primal problem, formalized in Section 2, one can derive the dual using the duality presented in class. Defining as $\mathbf{y}$ the dual variable the dual problem can be written as:

$$\begin{aligned} &\max \mathbf{b}^T\mathbf{y} \\ &\text{s.t } \mathbf{A}^T\mathbf{y} = \mathbf{c} \\ &\mathbf{y} \in \mathbb{R}^+ \end{aligned}$$

The dual variables represent the rates of change of the objective function with respect to changes in the right-hand side values of the constraints. These variables are crucial for sensitivity analysis, where you assess how changes in problem coefficients (e.g., constraint coefficients) affect the optimal solution. Dual variables help in understanding how the optimal objective value responds to these coefficient adjustments. This peculiarity will be pointe out in Chapter 8.
Dual variables are also used in optimality conditions such as complementary slackness. These conditions state that if a constraint is not binding (i.e., has slack), its dual variable must be zero. Conversely, if a constraint is binding (active), its dual variable is non-zero.

3) Q - the state-action value function

One can start with the more readable formulation presented in Chapter 2 and introduce the intermediate variable Q. The state-action value function $Q : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ maps to a state s and an action a, the expected discounted sum of future rewards when applying this action a in the environment, followed by an optimal policy, starting from the state s.

$$\begin{aligned} &\min(1-\gamma)\sum_{s\in S} P_0(s)V(s) \\ &\text{s.t. } Q(s,a) = R(s,a) + \sum_{s'\in S} P^a_{ss'}\gamma V\left(s'\right) \quad \forall s,a \\ &\quad V(s) \geqslant Q(s,a) \qquad\qquad\qquad\qquad\quad \forall s,a \end{aligned}$$

4) The optimal policy

In order to get the optimal policy of a given state we can take the argmax of Q over all possible actions in a given state.

$$\mu^*(s) \in \underset{a \in \mathcal{A}}{\operatorname{argmax}} Q(s, a)$$

Therefore, after computing the value of Q for every state and for every action is easy to obtain the optimal policy.
The computation complexity of computing the optimal policy for a given state is proportional to the possible number of actions for that given state. For all states this adds up to be the number of states times the number of actions per state: $O(dim(\mathcal{A}) \cdot dim(\mathcal{S}))$.

3 Solving Control problems

5) Optimal policy computation

The problem presented in Section 3 was solved for both environments using Gurobi. All the provided results are obtained with a value of $\gamma = 0.9$.

5.1) Solution for the taxi problem

The solution to the taxi problem returns the optimal path the agent must take to minimise the expected cost. The optimal choices it makes are presented in Fig. 2. The state values are presented in Tab. 1 and 2.

- 5.43	B	-6.30	-6.67	-7.00
-4.92	-5.43	-5.89	B	-7.30
-4.36	B	-6.30	B	-7.57
-3.73	B	-6.67	-7.00	-7.30
-3.04	B	B	-7.30	-7.57

Table 1: State values - No passenger

- 0.45	B	04.56	06.17	07.97
00.61	01.79	03.10	B	09.97
-0.45	B	04.56	B	12.18
-1.41	B	06.17	07.97	09.97
-2.26	B	B	06.17	07.97

Table 2: State values - With passenger

As expected the optimal policy follows the shortest path to reach the passenger if the taxi is empty or to reach the arrival if the taxi is full.
It can be seen, moreover, that for both the path with the empty taxi and the path with the full taxi there are two equivalent optimal paths: without the passenger, if the agent is in position (3,5) on the board it is equivalent to move upwards and downwards; similarly the full taxi will achieve the same result by moving upwards or downwards if it is in position (2,3). This is

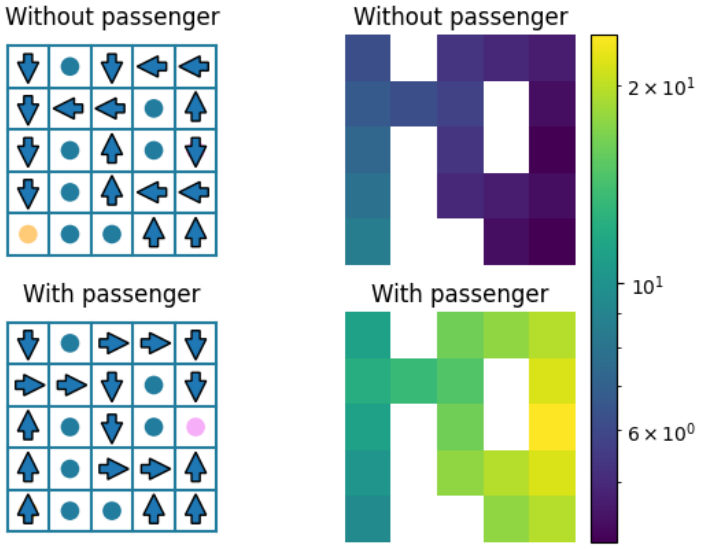

Figure 2: Optimal actions for the taxi problem along with the state value - Note that, for visualisation reasons, the values have been shifted to be positive and are displayed in a logarithmic scale.

evidenced by the equivalence of the state values along those two paths; the choice made is pseudo-random and depends on the implementation of the argmax.

5.2) Solution for the Blackjack problem

The solution to the blackjack problem returns the optimal choices the agent should take to maximize the chance of winning the games. The optimal choices are presented in Fig. 3.

	12	13	14	15	16	17	18	19	20	21
1	-0.3153	-0.3668	-0.3959	-0.3959	-0.3959	-0.2527	0.0270	0.2931	0.5444	0.8331
2	-0.2395	-0.2542	-0.2542	-0.2542	-0.2542	-0.1155	0.1556	0.4141	0.6588	0.8888
3	-0.2174	-0.2174	-0.2174	-0.2174	-0.2174	-0.0834	0.1790	0.4295	0.6674	0.8915
4	-0.1794	-0.1794	-0.1794	-0.1794	-0.1794	-0.0501	0.2033	0.4458	0.6764	0.8943
5	-0.1407	-0.1407	-0.1407	-0.1407	-0.1407	-0.0160	0.2284	0.4626	0.6859	0.8973
6	-0.1015	-0.1015	-0.1015	-0.1015	-0.1015	0.0185	0.2539	0.4799	0.6956	0.9004
7	-0.2220	-0.2796	-0.3334	-0.3838	-0.4294	-0.0488	0.4226	0.5994	0.7668	0.9240
8	-0.2652	-0.3200	-0.3712	-0.4191	-0.4639	-0.3310	0.1781	0.6280	0.7835	0.9295
9	-0.3181	-0.3695	-0.4175	-0.4624	-0.5044	-0.3788	-0.1203	0.3689	0.7989	0.9345
10	-0.3794	-0.4268	-0.4711	-0.5126	-0.5431	-0.4232	-0.1832	0.0568	0.5276	0.9392

Figure 3: Best actions for the Blackjack problem along with the state value - Green = Hit; Red = Stand

The figure shows the solution: the x-axis represents the player's score and the y-axis is the dealer's face up card. For a more realistic implementation,

the suggested assumption that the dealer's sum of cards is uniformly distributed over values greater than the face-up card and lower than or equal to 21 was not used. Instead the actual probabilities of the dealer getting each number and of busting are calculated recursively using the dealer's fixed policy.

6) Computing the dual variables without doing the dual problem

To calculate the dual variables of the problem without solving the dual problem, which will be the subject of the next chapter, we can use the output of Gurobi, which provides the value of the dual variables from solving the primary problem. The dual variables for the taxi problem are shown below in Tab. 3 and Tab. 4.

0.003	B	0.010	0.008	0.005
0.048	0.048	0.050	B	0.003
0.046	B	0.043	B	0.045
0.044	B	0.044	0.046	0.045
0.043	B	B	0.003	0.003

Table 3: Dual variables - No passenger

0.003	B	0.003	0.005	0.008
0.040	0.039	0.038	B	0.010
0.039	B	0.037	B	0.046
0.040	B	0.036	0.037	0.039
0.041	B	B	0.003	0.003

Table 4: Dual variables - With passenger

The same can be done for the blackjack problem. The results are shown in Fig. 4.

	12	13	14	15	16	17	18	19	20	21
1	0.1000	0.1069	0.1143	0.1143	0.1143	0.1143	0.1143	0.1143	0.1143	0.1143
2	0.1000	0.1069	0.1069	0.1069	0.1069	0.1069	0.1069	0.1069	0.1069	0.1069
3	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000
4	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000
5	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000
6	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000
7	0.1000	0.1069	0.1143	0.1222	0.1307	0.1307	0.1307	0.1307	0.1307	0.1307
8	0.1000	0.1069	0.1143	0.1222	0.1307	0.1398	0.1398	0.1398	0.1398	0.1398
9	0.1000	0.1069	0.1143	0.1222	0.1307	0.1398	0.1398	0.1398	0.1398	0.1398
10	0.1000	0.1069	0.1143	0.1222	0.1307	0.1307	0.1307	0.1307	0.1307	0.1307

Figure 4: Dual variables values with the optimal action policy - Green = Hit; Red = Stand

7) Comparison between primal problem and dual problem

After implementing the dual problem, it can be verified that the calculated dual variables are the solution of the dual. Moreover, the objective function

has the same value for both the primal and dual problem solution (strong duality). Using the output provided by Gurobi, one can proceed to analyse the speed in terms of iterations required for convergence. The results are shown in Tab 5.

For both the taxi problem and the blackjack problem the solving proce-

	Taxi problem	Blackjack problem
Primal	43 it.	100 it.
Dual	46 it.	31 it.

Table 5: Comparison between the number of iterations required for solving the primal and the dual problem

dure was performed disabling the presolver implemented in Gurobi, since the number of iterations with the presolver enabled were zero for the blackjack problem. While for the taxi problem the presolver speed up the solution of the primal problem and slow down the solution of the dual.
Moreover only the number of iteration are taken into account to analyze the speed since the time needed was close to zero (10^{-2}) for every case.

As one can note from the notebook, both the *readable* and the matrix based formulation were implemented. For the taxi problem the *readable* implementation of the taxi problem the solver used slightly more iterations than the formal matrix-based implementation. For the blackjack problem they used exactly the same number of iterations.

To avoid solving the dual problem we can compute the dual optimization variables by using the dual simplex algorithm on the primal problem instead of applying the primal simplex on the dual problem. The Dual Simplex Algorithm starts from the primal problem but its main goal is to solve the dual problem. In other words, the Dual Simplex Algorithm focuses on finding a feasible dual solution while maintaining primal feasibility.

Solving the Dual Simplex Algorithm, for the taxi problem, takes 53 iterations while for the blackjack problem it takes 199 iteration. Therefore, in the considered case, it is slightly slower than Simplex Algorithm but does not require the implementation of the dual problem.

8) Interpretation of the dual variables

Given the fact that the dual variable can be seen as a probability measure representing how often the agent is in a state on expectation, the dual problem can be seen as maximizing the expected reward.

For the blackjack problem we can see that all the actions we don't take have their corresponding dual variable equal to zero. This means that even if their rewards change it still doesn't change the outcome of the game. From the taxi problem it is clear that the states along the optimal path have a higher optimal-action dual value than the other states, even if it is the optimal action. This indicates that these states are "more likely". An interesting observation is that for the two equally optimal paths on the taxi driver problem they have different dual variables. This indicates that we have the case where there are at least two (probably four) equivalently optimal solutions to the dual problem, distinguished by whether or not we go up or go down in the intersections. Changing the RHS of the constraint in such a way that the optimal path is not optimal anymore, the dual optimal basis changes: i.e. changing the cost of going down at the bifurcation infinitesimally results in a change of the optimal path.

For the blackjack problem we can see that the dual variables of standing are the same for any given dealer face card. This is because under the assumptions we have made they are all equally likely to happen. When the policy is hitting we can see that the dual variables increase towards the right. This is because there are more combinations that e.g. go through the hand of 13, than through 12. 13 can be achieved by getting 13 directly or through getting 12 and then drawing a 1. 12 can only be achieved by starting at 12. If we didn't assume uniform distribution of initial state, this would be different.