# Reinforcement Learning and Optimal Control, Dimitri Bertsekas

## Deterministic Dynamic Programming

All DP problems involve a discrete time dynamic system that generates a sequence of states under the influence of control. In finite horizon problems the system evolves over a finite number of time steps (also called stages). The state and control at time are denoted by and , respectively. In deterministic systems, is generated non-randomly, i.e., it is determined solely by and .

### Deterministic Problems

A deterministic DP problem involves a discrete-time dynamic system of the form:

(1)

where

is the time index

is the state of the system, an element of some space

is the control or decision variable, to be selected at time k from some given set that depends on

A diagram of a diagram

Description automatically generated

Figure 1: A deterministic N-stage optimal control problem. Starting from state , the next state under control is generated non-randomly according to

and a stage cost is incurred.

is a function of that describes the mechanism by which the state is updated from time to time .

is the horizon or number of times control is applied,

The set of all possible is called the ***state space*** at time . Similarly, the set of all possible is called the ***control space*** at time . Both the state space and the control space can depend on .

The problem also involves a cost function that is additive in the sense that the cost incurred at time , denoted by , accumulates over time. Formally, is a function of that takes real number values, and may depend on . For a given initial state , the total cost of a control sequence is

(2)

where is a terminal cost incurred at the end of the process. This cost is a well-defined number, since the control sequence together with determines exactly the state sequence via the system equation (1). We want to minimize the cost (2) over the sequences that satisfy the control constraints, thereby obtaining the optimal value

as a function of . Figure 1 illustrates the main elements of the problem.

We will next illustrate deterministic problems with some examples.

A diagram of a stage

Description automatically generated

Figure 2: Transition graph for a deterministic finite-state system. Nodes correspond to states . Arcs correspond to state-control pairs . An arc has start and end nodes and , respectively. We view the cost of the transition as the length of this arc. The problem is equivalent to finding a shortest path from initial node to terminal node .

#### Discrete Optimal Control Problems

There are many situations where the state and control are naturally discrete and take a finite number of values. Such problems are often conveniently specified in terms of an acyclic graph specifying for each state the possible transitions to next states . The nodes of the graph correspond to states and the arcs of the graph correspond to state-control pairs . Each arc with start node corresponds to a choice of a single control and has as end node the next state . The cost of an arc is defined as (see Figure 2). To handle the final stage, an artificial terminal node is added. Each state at stage is connected to the terminal node with an arc having cost .

Note that control sequences correspond to paths originating at the initial state (node at stage ) and terminating at one of the nodes corresponding to the final stage . If we view the cost of an arc as its length, we see that *a deterministic finite-state finite-horizon problem is equivalent to finding a minimum-length (or shortest) path from the initial node of the graph to the terminal node* . Here, by path we mean a sequence of arcs such that given two successive arcs in the sequence the end node of the first arc is the same as the start node of the second. By the length of a path we mean the sum of the lengths of its arcs.

Generally, combinatorial optimization problems can be formulated as deterministic finite-satte finite-horizon optimal control problems.

#### Example: Deterministic Scheduling

Suppose that to produce a certain product, four operations must be performed on a certain machine. The operations are denoted by , , , and . We assume that operation can be performed only after operation has been performed, and operation can be performed only after operation has been performed. Thus, the sequence is allowable but the sequence is not. The setup cost for passing from any operation to any other operation is given. There is also an initial startup cost or for starting with operation or , respectively (see Figure 3). The cost of a sequence is the sum of the setup costs associated with it; for example, the operation sequence has cost

We can view this problem as a sequence of three decisions, namely the choice of the first three operations to be performed (the last operation is determined from the preceding three). It is appropriate to consider as state the set of operations already performed, the initial state being an artificial state corresponding to the beginning of the decision process. The possible state transitions corresponding to the possible states and decisions for this problem are shown on Figure 3. Here the problem is deterministic i.e. at a given state each choice of control leads to a uniquely determined state. For example, at state the decision to perform operation D leads to state ACD with certainty and has a cost . Thus, the problem can be conveniently represented in terms of the transition graph of Figure 3. The optimal solution corresponds to the path that starts at the initial state and ends at some state at the terminal time and has minimum sum of arc costs plus the terminal cost.

A diagram of a diagram

Description automatically generated

Figure 3: The transition graph of the *Deterministic Scheduling* problem. Each arc of the graph corresponds to a decision leading from some state (the start node of the arc) to some other state (the end node of the arc). The corresponding cost is shown next to the arc. The cost of the last operation is shown as a terminal cost next to the terminal nodes of the graph.

#### Continuous-Spaces Optimal Control Problems

Many classical problems in control theory involve a state that belongs to a Euclidean space, i.e., the space of n-dimensional vectors of real variables, where is some positive integer. The following is representative of the class of linear-quadratic problems, where the system equation is linear, the cost function is quadratic, and there are no control constraints. In our example, the states and controls are one dimensional but there are multi-dimensional extensions which are widely applicable.

#### Example: A Linear-Quadratic Problem

A certain material is passed through a sequence of ovens (see Figure 4 below).

Denote

: initial temperature of the material

: temperature of the material at the exit of oven ,

: heat energy applied to the material in oven .

In practice there will be some constraints on , such as nonnegativity.

However, for analytical tractability one may consider the case where is unconstrained and check later if the solution satisfies some natural restrictions in the problem at hand.

We assume a system equation of the form

// TODO: finish the section on Continuous-Spaces Optimal Control Problems

## References

[1] [Reinforcement Learning and Optimal Control, Dimitri Bertsekas, 2019 (Draft)](https://github.com/dimitarpg13/reinforcement_learning_and_game_theory/blob/main/books/Reinforcement_Learning_and_Optimal_Control_Bertsekas.pdf)