1
Introduction and Overview

1.1 What Is Reinforcement Learning?

Concisely defined, reinforcement learning, abbreviated as RL, is the discipline of learning and acting in environments where sequential decisions are made. That is, the decision made at a given time will be followed by other decisions, and therefore the decision maker has to consider the implications of their decision on subsequent decisions.

In the early days of the field, there was an analogy drawn between human learning and computer learning. While the two are certainly tightly connected, this is merely an analogy that serves to motivate and inspire. Other terms that have been used are approximate dynamic programming (ADP) and neuro-dynamic programming (NDP), which to us mean the same thing but focus on a specific collection of techniques that grew into the discipline known as "RL."

Origins of reinforcement learning. Reinforcement learning has roots in quite a few disciplines. Naturally, by our own indoctrination, we are going to look through the lens of computer science and machine learning. From an engineering perspective, optimal control is the "mother" of RL, and many of the concepts that are used in RL naturally come from optimal control. Other notable origins are in operations research, where the initial mathematical frameworks had originated. Additional disciplines include neuroscience, psychology, statistics, and economics.

The origins of the term "reinforcement learning" are in psychology, where it refers to learning by trial and error. While this inspired much work in the early days of the field, current approaches are mostly based on machine learning and optimal control. We refer the reader to section 1.6 in [146] for a detailed history of RL as a field.

1.2 Motivation for RL

In recent years there has been a renewed interest in RL. The new interest is grounded in emerging applications of RL, and also in the progress of deep learning, which has been impressively applied for solving challenging RL tasks. But for us, the interest comes from the *promise* of RL and its potential to be an effective tool for control and behavior in dynamic environments.

Over the years, RL has proven to be highly successful for playing board games that require long-horizon planning. Early in 1962, Arthur Samuel [125] developed

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 16 Nov 2026 at 10:18:26, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/9781009711067.001

a checkers game, which was at the level of the best human. His original framework included many of the ingredients that later contributed to RL, as well as search heuristics for large domains. In 1992, Gerald Tesauro developed the TD-gammon program [154], which used a two-layer neural network to achieve a high-performance agent for playing the game of backgammon. The network was trained from scratch, by playing against itself in simulation and using a temporal differences learning rule. One of the amazing features of TD-gammon was that even in the *first* move, it played a different game move than the typical opening that backgammon grandmasters use. Indeed, this move was later adopted in the backgammon community [155]. More recently, DeepMind has developed AlphaGo – a deep neural network-based agent for playing Go, which was able to beat the best players in the world, solving a long-standing challenge for artificial intelligence [136].

To complete the picture of computer board games, we should mention Deep Blue, from 1996, which was able to beat the then world champion, Kasparov [26]. This program mainly built on heuristic search, and new hardware was developed to support it. Recently, DeepMind's AlphaZero matched the best chess programs (which are already much better than any human player), using an RL-based approach [138].

Another domain, popularized by DeepMind, is playing Atari video games [108], which were popular in the 1980s. DeepMind was able to show that deep neural networks can achieve human-level performance, using only the raw video image and the game score as input (and having no additional information about the goal of the game). Importantly, this result reignited interest in RL in the robotics community, where acting based on raw sensor measurements (a.k.a. "end-to-end") is a promising alternative to the conventional practice of separating decision making into perception, planning, and control components [95].

More recently, interest in RL sparked yet again, as it proved to be an important component in fine tuning large language models to match user preferences, or to accomplish certain tasks [115, 172]. One can think of the sequence of words in a conversation as individual decisions made with some higher level goal in mind, and RL fits naturally with this view of language generation.

While the RL implementations in each of the different applications mentioned above were very different, the fundamental models and algorithmic ideas were surprisingly similar. These foundations are the topic of this book.

1.3 The Need for This Book

There are already several books on RL. However, while teaching RL in class, we felt that there is a gap between advanced textbooks that focus on one aspect or another of the art and more general books that opt for readability rather than rigor. Coming from computer science and electrical engineering backgrounds, we like to teach RL in a rigorous, self-contained manner. This book serves this purpose and is based on our lecture notes for an advanced undergraduate course that we have taught for over a decade at Tel Aviv University and at the Technion.

Complementing this book is a booklet with exercises and exam questions to help students practice the material. These exercises were developed by us and our teaching assistants over the years.

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 16 Nov 2026 at 10:18:26, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/9781009711067.001

1.4 Mathematical Models

The main mathematical model we will use is the celebrated Markov decision process (MDP). The model tries to capture uncertainty in the dynamics of the environment, the actions, and our knowledge. The main focus will be on sequential decision making, namely, selecting actions one after the other. The evaluation will consider the long-term effect of the actions, trading off immediate rewards with long-term gains.

In contrast to standard machine learning problems, such as regression or classification, the RL model has a notion of a *state*, and the algorithm influences the state through its actions. The algorithm is faced with an inherent tradeoff between exploitation (getting the most reward given the current information) and exploration (gathering more information about the environment).

There are other models that are useful, such as partially observed MDPs (POMDPs), where the exact state is not fully known, and "bandits," where the current decision has no effect on subsequent decisions. In accordance with most of the literature, with the exception of Chapter 14, the book concerns the MDP model.

1.5 Book Organization

The book is thematically comprised of two main parts – *planning* and *learning*.

Planning: The planning theme develops the fundamentals of optimal decision making in the face of uncertainty, under the Markov decision process model. The basic assumption in planning is that the MDP model is *known* (yet, as the model is stochastic, uncertainty must still be accounted for in making decisions). In a preface to the planning section (Chapter 2), we motivate the MDP model and relate it to other models in the planning and control literature. In Chapter 3 we introduce the problem and basic algorithmic ideas in the deterministic setting. In Chapter 4 we review the topic of Markov chains, which the Markov decision process model is based on, and then in Chapter 5 we introduce the finite-horizon MDP model and a fundamental dynamic programming approach. Chapter 6 covers the infinite-horizon discounted setting, and Chapter 7 covers the episodic setting. Chapter 8 covers an alternative approach for solving MDPs using a linear programming formulation.

Learning: The learning theme covers decision making when the MDP model is *not known in advance*. In a preface to the learning section (Chapter 9), we motivate this learning problem and relate it to other learning problems in decision making. Chapter 10 introduces the *model-based* approach, where the agent explicitly learns an MDP model from its experience and uses it for planning decisions. Chapter 11 covers an alternative *model-free* approach, where decisions are learned without explicitly building a model. Chapters 12 and 13 address learning of approximately optimal solutions in *large* problems, that is, problems where the underlying MDP model is intractable. Chapter 12 approaches this topic using approximation of the value function, while Chapter 13 considers policy approximations. In Chapter 14 we study the online nature of decision making in more detail, under the framework of regret minimization. In this chapter we mostly focus on the special case of multi-arm bandits, which can be viewed as an MDP with a single state and unknown rewards.

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 16 Nov 2026 at 10:18:26, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/9781009711067.001

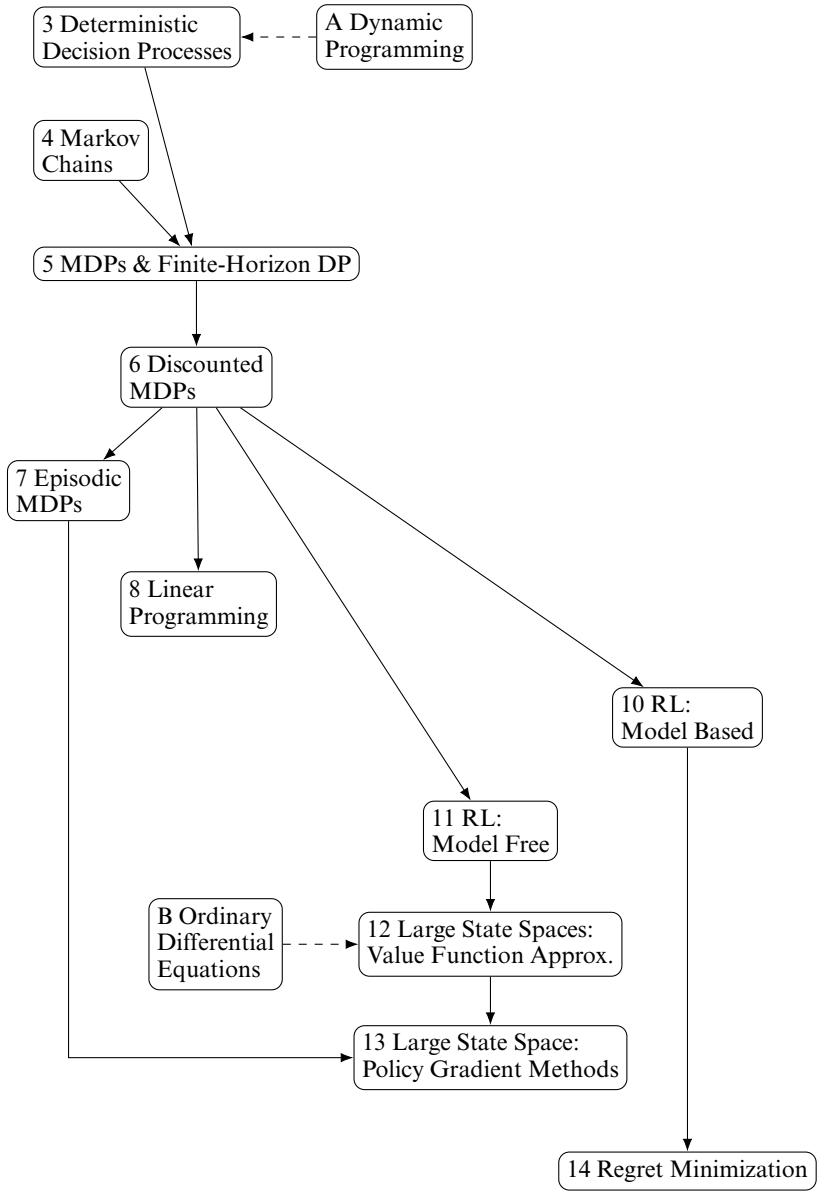


Figure 1.1 Chapter dependencies.

The chapters are organized to be taught in sequence. Figure 1.1 depicts the dependency structure between the different chapters and may be convenient for teachers who opt for a different course structure.

Acknowledgments. We are grateful to our colleagues and students who helped us improve this manuscript. In alphabetical order: Stav Belogolovsky, Esther Derman, Yonathan Efroni, Uri Gadot, Hector Geffner, Nan Jiang, Navdeep Kumar, Tal Lancewicki, Orin Levy, Nadav Merlis, Avinash Mohan, Mirco Mutti, Ofir Nabati, Benny Pereth, Uri Sherman, Satinder Singh, and Jonathan Somer. Special thanks are due to Elad Sharony for carefully sieving through this manuscript. Some parts

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 16 Nov 2026 at 10:18:26, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/9781009711067.001

of this book evolved from notes that were initially written by Nahum Shimkin. We thank him deeply for structuring the first versions of the course upon which this book is based.

1.6 Bibliography Notes

Markov decision processes have a long history. The first work directly addressed Markov decision processes and reinforcement learning are [13] and [58]. The book by Bellman [13], based on a sequence of works by him, introduced the notion of *dynamic programming* and the *principle of optimality*, and defined discrete-time MDPs. The book of Howard [58], building on his PhD thesis, introduced the policy iteration algorithm as well as a clear algorithmic definition of value iteration. A precursor work by Shapely [134] introduced a discounted MDP model for stochastic games.

There are a variety of books addressing Markov decision processes and reinforcement learning. Puterman's book [119] gives an extensive exposition of the mathematical properties of MDPs, including planning algorithms. Bertsekas and Tsitsiklis [18] and Bertsekas [17] give a stochastic processes approach for reinforcement learning and control.

Sutton and Barto [146] give a general exposition of modern reinforcement learning, which is focused more on implementation issues and less on mathematical issues. They also give an excellent historical account of the development of RL from its infancy. Szepesvári's monograph [149] gives an outline of basic reinforcement learning algorithms.

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 16 Nov 2026 at 10:18:26, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/9781009711067.001