Faculty of Science and Bio-Engineering Sciences
Department of Computer Science
Artificial Intelligence Laboratory

Thinking in Trade-Offs

Training Agents to Balance Conflicting Objectives under Uncertainty

Dissertation submitted in fulfilment of the requirements for the degree of Doctor of Science: Computer science

Willem Röpke

Brussels, February 2026

Promotors: Prof. Dr. Ann Nowé (Vrije Universiteit Brussel)
Prof. Dr. Roxana Rădulescu (Utrecht University)
Dr. Diederik M. Roijers (Vrije Universiteit Brussel, City of Amsterdam)

Printed by
Crazy Copy Center Productions
VUB Pleinlaan 2, 1050 Brussel
Tel : +32 2 629 33 44
crazycopy@vub.be
www.crazycopy.be

ISBN : 9789493461352
NUR CODE : 984
THEMA : UYQM

Members of the Jury

Prof. Dr. Beat Signer	Vrije Universiteit Brussel, BE (Chair)
Prof. Dr. Bart Bogaerts	Vrije Universiteit Brussel, BE Katholieke Universiteit Leuven, BE (Secretary)
Prof. Dr. Ann Nowé	Vrije Universiteit Brussel, BE (Promotor)
Prof. Dr. Roxana Rădulescu	Utrecht University, NL (Promotor)
Dr. Diederik M. Roijers	Vrije Universiteit Brussel, BE City of Amsterdam, NL (Promotor)
Prof. Dr. Marie-Anne Guerry	Vrije Universiteit Brussel, BE
Prof. Dr. Guillermo A. Pérez	University of Antwerp, BE
Prof. Dr. Giorgia Ramponi	University of Zurich, CH

Summary

Every day, people face decisions that involve multiple, often conflicting objectives: balancing cost against quality, safety against speed, or personal benefit against social responsibility. These trade-offs rarely admit a single "right" answer, and they become even more challenging when other decision-makers are involved. Reinforcement learning offers a powerful framework for constructing artificial agents that act autonomously in complex, uncertain environments, learning through trial and error how to make effective decisions. Yet, most existing approaches focus on a single objective, assuming away the trade-offs that are intrinsic to real decision-making. This thesis addresses that gap directly. Its central theme is how to design agents that can *think in trade-offs*, that is, agents that can reason about multiple objectives and act optimally under uncertainty.

The contributions unfold in three parts that build on one another. First, we bridge single- and multi-objective reinforcement learning by showing how decomposition techniques allow well-established single-objective methods to be extended to learning the Pareto front, a classical solution set that captures efficient trade-offs for certain decision-makers. Building on this foundation, we then introduce and analyse alternative solution concepts that more directly reflect decision-makers' preferences, developing rigorous theoretical guarantees and demonstrating their practical relevance. Finally, we turn to multi-agent systems and establish a novel reduction from multi-objective to single-objective games, which not only provides new theoretical insights but also enables the transfer of powerful algorithms across domains.

Taken together, these results provide both theoretical and practical advances. Theoretically, the thesis deepens our understanding of what it means to act optimally under multiple objectives. Practically, it demonstrates how learning agents can be equipped to handle genuine trade-offs. By establishing strong connections between multi-objective and single-objective paradigms, the thesis lays the groundwork for future progress to be accelerated, enabling advances in one field to be immediately translated into the other. These advances bring us closer to systems that can adapt their behaviour to different stakeholders, balance conflicting objectives transparently, and operate responsibly in safety-critical real-world environments.

Samenvatting

Elke dag staan mensen voor keuzes die meerdere, vaak tegenstrijdige doelstellingen omvatten: kosten tegenover kwaliteit, veiligheid tegenover snelheid, of persoonlijk voordeel tegenover maatschappelijk belang. Zulke trade-offs leveren zelden één "juiste" oplossing op, en worden nog uitdagender wanneer meerdere beslissers betrokken zijn. Reinforcement learning biedt een krachtig raamwerk voor het construeren van kunstmatige agenten die autonoom handelen in complexe, onzekere omgevingen, en via vallen en opstaan leren hoe zij doeltreffende beslissingen kunnen nemen. Toch richt het merendeel van de bestaande benaderingen zich op één enkele doelstelling, en negeert daarmee de trade-offs die eigen zijn aan echte besluitvorming. Dit proefschrift pakt die leemte rechtstreeks aan. Het centrale thema is hoe we agenten kunnen ontwerpen die *in trade-offs kunnen denken*: agenten die over meerdere doelstellingen kunnen redeneren en optimaal kunnen handelen onder onzekerheid.

De bijdragen ontvouwen zich in drie samenhangende delen. Eerst overbruggen we single- en multi-objective reinforcement learning door te laten zien hoe decompositietechnieken het mogelijk maken om gevestigde single-objective methoden uit te breiden naar het leren van het Pareto front, een klassieke oplossingsverzameling die efficiënte trade-offs vastlegt voor specifieke beslissers. Op deze basis introduceren en analyseren we vervolgens alternatieve oplossingsconcepten die de voorkeuren van beslissers directer weerspiegelen, waarbij we formele theoretische garanties ontwikkelen en hun praktische relevantie aantonen. Ten slotte richten we ons op multi-agent systemen en formuleren we een nieuwe reductie van multi-objective naar single-objective games, die niet alleen nieuwe theoretische inzichten oplevert maar ook de overdracht van krachtige algoritmen tussen domeinen mogelijk maakt.

Gezamenlijk leveren deze resultaten zowel theoretische als praktische vooruitgang op. Theoretisch verdiept het proefschrift ons begrip van wat het betekent om optimaal te handelen onder meerdere doelstellingen. Praktisch toont het aan hoe lerende agenten kunnen worden uitgerust om met echte trade-offs om te gaan. Door sterke verbindingen te leggen tussen multi-objective en single-objective paradigma's, legt het proefschrift een fundament om toekomstige vooruitgang te versnellen, zodat ontwikkelingen in het ene veld direct kunnen worden vertaald naar het andere. Deze vooruitgang brengt ons dichter bij systemen die hun gedrag kunnen aanpassen aan verschillende belanghebbenden, conflicterende doelstellingen transparant kunnen afwegen, en verantwoordelijk kunnen opereren in veiligheidskritische omgevingen in de echte wereld.

Acknowledgments

It is remarkable how four years can feel both impossibly long and strangely short. My PhD journey was challenging, exciting, occasionally frustrating, and ultimately rewarding, and it would certainly not have been possible without the support of many people.

First of all, I would like to express my gratitude to my amazing team of supervisors: Ann Nowé, Roxana Rădulescu, and Diederik M. Roijers. Ann, thank you for providing me with so many opportunities and for supporting me at every stage. Roxana, thank you for sparking my love for research and for always motivating me towards the frontier. Diederik, thank you for introducing and reinforcing my interest in multi-objective reinforcement learning. The three of you have been incredible mentors, and I have learned so much from each of you.

I would also like to thank all the members of my jury, Beat Signer, Bart Bogaerts, Marie-Anne Guerry, Guillermo A. Pérez, and Giorgia Ramponi, for taking the time to read my thesis so carefully, for providing helpful feedback, and for asking interesting questions during my defence. It was a pleasure to discuss four years of research with such an interested and knowledgeable audience.

Doing a PhD is not a solo endeavour, and I have been fortunate to be surrounded by many great people within my lab. Andrea, thank you for your knowledge of F1, which helps the days go by faster. Hicham, your helpfulness and enthusiasm are unmatched, and I always knew I could count on you when I needed some help. Jérôme, thank you for being my original office buddy. Lucas, your year at the lab was memorable, and your work ethic and passion were inspiring. Mathieu, thank you for helping me get IPRO across the finish line. I would also like to make a special mention of Raphael and Florent, my "partners in crime". Together, we have shared many workshops, conferences, summer schools, and countless hours of discussion. I would not have enjoyed my PhD as much without you two. I would also like to thank all of the supporting staff, especially Brigitte, Anca, Frederik, and Youri. We are only able to do our research (relatively) uninterrupted thanks to your hard work. The AI Lab has always been a welcoming and energetic place to work, and I feel lucky to have spent these years there.

I was fortunate enough to collaborate with many great researchers during my PhD. I would like to thank all of them for their contributions, insights, and interesting discussions. In particular, I would like to thank Patrick Mannion, Enda Howley, and Jakob Foerster, who have all graciously welcomed me into their respective groups and without whom much of the research throughout my PhD would not have been possible.

I would like to thank my friends for listening to my rants about AI and my overly complicated explanations of my research. In no particular order, thank you Hendrik, Roel, Joris, Nils, Arno, Tibeaux, and Thomas. Your friendship has meant a great deal to me over the years. Thank you also to Frederik. While you are not here anymore to share this moment, I hope you are at peace. I am grateful for the time we had together.

I would also like to thank my family, in particular my parents and my brothers. Thank you for always supporting me, for trying to understand what it is that I actually do, and for encouraging me to follow my dreams. Toon and Korneel, I am grateful for the bond the three of us share. I could not ask for better brothers. Mama and papa, I only got here because of the wonderful start you gave me.

Finally, thank you Alis, for being a wonderful person. You have had to endure many unexpected changes of plans, long nights, and ill-timed deadlines. You are the most amazing partner I could have wished for. I will never be able to express my gratitude for your love and patience throughout this journey.

Contents

1 Introduction

1 The State of AI Research

The field of Artificial Intelligence (AI) is undergoing a period of profound transformation. Rapid progress in machine learning, fuelled by the availability of extensive datasets [Sun et al., 2017; Gao et al., 2021], powerful computational infrastructure [Jouppi et al., 2017; Bradbury et al., 2018], and increasingly sophisticated algorithms [Silver et al., 2016; Vaswani et al., 2017], has produced remarkable breakthroughs across multiple domains. From natural language processing [Anil et al., 2023; OpenAI et al., 2024] to computer vision [McKinney et al., 2020; Kirillov et al., 2023], AI systems are now capable of solving tasks that were previously regarded as uniquely human.

Among the central challenges in contemporary AI research is the creation of *agentic AI*[1]: artificial *agents*, whether virtual or physically embodied, that act (semi-)autonomously in complex and dynamic environments. Such agents must not only acquire knowledge through interaction and experience, but also reason about their actions in ways that are aligned with human values and societal norms. Agentic AI is already influencing sectors such as healthcare [Komorowski et al., 2018], transportation [Kendall et al., 2019; Xu et al., 2017], and energy systems [Ruelens et al., 2018; Luo et al., 2022]. As these systems are deployed in increasingly high-stakes domains, there is a

[1] Although the terms "agentic AI" and "agent" are now often associated with large language models (LLMs), we use them here in a broader sense that encompasses artificial agents of any form.

growing need for decision-making frameworks that offer formal guarantees, provide quantifiable confidence in their outputs, and clarify the trade-offs and assumptions that govern agent behaviour [Delgrange, 2024].

2 Research Context

The developments in AI bring to the forefront a fundamental challenge: real-world decisions are rarely about optimising a single objective in isolation. Instead, they involve reconciling multiple, often conflicting, objectives under conditions of uncertainty and limited information. Understanding how to design agents that can navigate such trade-offs is therefore central to the broader project of agentic AI, and it provides the context for the questions pursued in this thesis.

Consider the operation of a traffic signal at a busy intersection. Extending the green phase for one direction reduces waiting times for those vehicles, but it simultaneously increases delays for other lanes. Allowing long queues to form raises emissions from idling cars, whereas switching too frequently may compromise safety and reduce overall throughput. Beyond a single intersection, neighbouring traffic lights interact with each other, creating complex dynamics that no signal can optimise in isolation.

This scenario exemplifies the core question of this thesis: how can autonomous agents make high-quality decisions in the presence of multiple objectives, uncertainty, and interaction with other agents? To address this, we integrate three foundational perspectives, *reinforcement learning*, *multi-objective decision-making*, and *multi-agent systems*, into a unified analytical framework and illustrate this in Fig. 1.1. This integration motivates a range of theoretical challenges, which we approach through principled algorithm design and mathematical analysis, with particular emphasis on solution concepts, existence results, and structural equivalences.

Reinforcement Learning

Suppose a traffic signal observes that congestion is building up in one direction and decides to extend the green phase. During this time, it notices that pedestrians are waiting to cross and keeps that information in mind for future decisions. This scenario consists of several essential components: an *agent* (the traffic signal) operating in an *environment* (the road network), which includes everything the agent can perceive and affect, and which evolves with a degree of *uncertainty* (such as varying traffic conditions). The agent has a set of *actions* (phase changes of the lights) and follows a *policy* that governs action selection given the current state of the environment. Finally, the agent *learns* from its experiences, adapting its policy over time to improve decision-making.

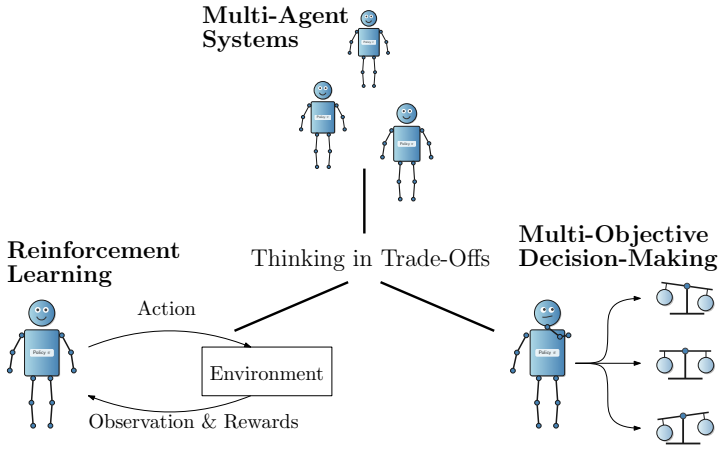


Figure 1.1: The three pillars of this thesis: reinforcement learning, multi-objective decision-making, and multi-agent systems. Each area contributes to a comprehensive framework for decision-making under uncertainty, with competing objectives and in the presence of other agents. The thesis develops principled formulations and analyses at the intersection of these domains.

Reinforcement Learning (RL) [Sutton and Barto, 2018; Szepesvári, 2010] is a subfield of artificial intelligence that formalises the idea that agents can acquire intelligent behaviour by interacting with their environment. Conceptually, RL descends from the operant conditioning tradition in behavioural psychology, where actions producing favourable outcomes are reinforced and therefore repeated [Thorndike, 1911; Skinner, 1953]. It is based on the observation that intelligent behaviour can emerge from trial and error: acting, observing outcomes, and adapting accordingly. RL formalises this process by providing algorithms through which agents learn a policy that maximises expected long-term cumulative reward. It has enabled notable successes in diverse domains, including game playing [Silver et al., 2016], robotic manipulation [Levine et al., 2016], and magnetic control in nuclear fusion [Degrave et al., 2022].

Classical RL frameworks typically assume that the agent's goal can be expressed through a well-defined, scalar, and Markovian reward signal [Silver et al., 2021; Bowling et al., 2023]. However, these assumptions have been widely criticised as overly restrictive for many real-world applications, where objectives are multifaceted, conflicting, and not easily reducible to a single number [Roijers et al., 2015; Abel et al., 2021; Vamplew et al., 2022; Skalse and Abate, 2023; Subramani et al., 2024].

Multi-Objective Decision-Making

In many real-world settings, agents must weigh multiple objectives simultaneously, such as cost, fairness, safety, efficiency, and sustainability [Roijers and Whiteson, 2017]. These trade-offs are increasingly mandated by ethical and legal frameworks, such as the European AI Act [European Parliament and Council, 2024]. Multi-objective decision-making extends classical optimisation [Miettinen, 1998] and decision theory [Greco et al., 2016] to explicitly model such considerations and has enabled progress in applications ranging from drug discovery [Zhou et al., 2019] to intelligent traffic control [Zintgraf et al., 2018] and epidemic mitigation [Reymond et al., 2024].

A frequent misconception is that multi-objective problems can simply be reduced to single-objective ones by assigning fixed weights to each criterion [Hayes et al., 2022a]. This reduction ignores the epistemic uncertainty and context-dependence of real-world preferences. A growing literature argues that complex value systems, such as those underpinning human and societal decision-making, cannot be faithfully captured by such linear scalarisation [Kahneman and Tversky, 1979; Ethayarajh et al., 2024].

In the case of traffic signal control, the agent cannot rely on fixed trade-off weights between objectives such as throughput, fairness across lanes, safety, and emissions. For instance, during peak hours it may prioritise clearing long queues to prevent gridlock, whereas at night it may favour energy efficiency and reduced emissions. *Multi-objective reinforcement learning* (MORL) [Gábor et al., 1998; Van Moffaert and Nowé, 2014; Hayes et al., 2022a] extends classical RL to this setting by employing vector-valued rewards. When preferences are not specified in advance, MORL algorithms aim to learn a *solution set* of policies that captures the space of relevant trade-offs [Alegre et al., 2023; Liu et al., 2025], thereby empowering the decision-maker to select a policy aligned with their individual priorities.

Multi-Agent Systems

Agents do not operate in isolation but rather in environments populated by other agents. These may be collaborators, competitors, or both, and their actions can significantly influence the outcomes of an agent's decisions. Multi-agent systems introduce a strategic layer to decision-making, where each agent's optimal behaviour depends on the actions of others [Shoham and Leyton-Brown, 2008]. Analysing such interactions draws on tools from game theory [Leyton-Brown, 2008] and multi-agent RL [Albrecht et al., 2024], and poses unique challenges for learning and coordination. Nonetheless, these insights have been applied with success in areas such as coordinating airport and maritime security operations [Tambe, 2012] and real-time adaptive traffic signal control [Smith et al., 2013].

In the case of traffic management, one intersection's decision to keep its light green may cause congestion to spill back into neighbouring intersections, reducing

the overall efficiency of the network. Agents must therefore reason not only about their own objectives and trade-offs, but also about the strategies of other agents with whom they share the environment. Game-theoretic solution concepts such as a *Nash equilibrium* [Nash, 1951] and a *correlated equilibrium* [Aumann, 1974] formalise the notion of stability in such settings: no agent has an incentive to deviate unilaterally from their strategy. These concepts form the foundation for modelling strategic interaction in both single- and multi-objective settings. When agents pursue multiple objectives or have private preferences over them, classical equilibrium notions must be extended accordingly, giving rise to the framework of multi-objective games [Shapley and Rigby, 1959; Rădulescu et al., 2020a].

3 Motivation

As we have just discussed, the settings in which we wish to deploy artificial agents rarely reduce to single-objective, single-agent problems. Traffic networks, energy markets, and clinical workflows are uncertain, involve competing objectives, and are shaped by the simultaneous actions of many decision-makers. If AI is to operate reliably in these contexts, agents must reason simultaneously about trade-offs and about the presence of other decision-makers.

We are now at an inflection point. The deployment of AI in high-stakes societal systems is no longer speculative but ongoing, and legislation, public opinion, and economic pressures are accelerating the demand for methods that are not only effective but also principled and trustworthy. Without rigorous approaches, agents risk producing behaviour that is unsafe, unfair, or inefficient in precisely the domains where errors are least tolerable [Amodei et al., 2016; Obermeyer et al., 2019; Knox et al., 2023].

At the same time, there is a tremendous opportunity. Research communities in reinforcement learning, game theory, decision theory, and multi-objective optimisation have each developed rich theoretical foundations and powerful algorithmic tools. Yet these traditions have largely evolved in isolation. A unified framework would allow us to leverage their respective strengths, transfer insights across settings, and scale methods to real-world problems.

This thesis is motivated by precisely this need. Its goal is twofold: first, to establish principled foundations through reductions and structural equivalences that connect multi-objective and multi-agent models to well-understood single-objective counterparts; and second, to develop algorithms that operationalise these insights, yielding methods that scale while preserving guarantees. Taken together, these contributions aim to enable agents that act responsibly and effectively in the environments where they are most needed.

4 Contributions

This thesis is structured around a central guiding question:

How can we develop principled methods for multi-objective decision-making and compute solutions that balance trade-offs in line with decision-makers' preferences by leveraging advances from single-objective problems?

Answering this question requires both a precise definition of what constitutes a good decision and algorithms capable of identifying such decisions in complex, potentially multi-agent environments. We take a bottom-up approach, starting from established solution concepts and building towards a unified framework for studying multi-objective and multi-agent reinforcement learning in artificial agents.

Structure. This thesis addresses three core research challenges, each combining theoretical analysis with algorithmic development. We begin by integrating reinforcement learning with the central concept from multi-objective optimisation: the Pareto front. While the Pareto front characterises efficient trade-offs for broad classes of decision-makers, it does not capture the needs of those whose choices are guided by expected utility. To fill this gap, we develop distributional solution sets that extend beyond Pareto efficiency and provide richer support for decision-making under uncertainty. Finally, we extend these ideas to settings with multiple interacting agents making decisions concurrently.

Integrating Multi- and Single-Objective RL

> **Challenge 1:** *How can recent advances in single-objective RL be leveraged to efficiently learn a Pareto front?*
>
> **Example:** The traffic signal's preferences over throughput, fairness, safety, and emissions are unknown. Instead of requiring a full utility specification, an artificial agent reuses single-objective RL with different scalarisations, assembling a set of policies that reflects the available trade-offs.

Chapter 3 introduces *Iterated Pareto Referent Optimisation* (IPRO), an anytime algorithm that decomposes a multi-objective problem into a sequence of constrained single-objective sub-tasks. Each sub-task can be solved using standard reinforcement learning algorithms, linking progress in single-objective RL directly to multi-objective settings. We show that IPRO converges to an approximate Pareto front in finitely many steps and to the true Pareto front in the limit, with explicit quality bounds at each iteration. We further analyse its complexity in terms of the number of sub-problems required to reach a given approximation accuracy. Empirically, IPRO outperforms prior methods while requiring less domain knowledge and offering broader applicability. By leveraging

specialised single-objective solvers, it also extends naturally to domains such as planning and multi-objective pathfinding.

This first step establishes how single-objective techniques can be systematically reused to construct multi-objective solution sets. Yet, the Pareto front is not always sufficient to support realistic decision-makers, motivating the next challenge.

Distributional Multi-Objective Decision-Making

> **Challenge 2:** *If decision-makers maximise expected utility, what solution concept captures optimality in multi-objective settings and how can it be computed?*
>
> **Example:** A traffic signal controller is not only concerned with the average waiting time but also with its variability: occasional long queues, sudden congestion, or irregular traffic surges. To make better decisions, an artificial agent must reason about the full distribution of outcomes, enabling it to select policies that are robust to such fluctuations and better reflect the preferences of the system.

Chapter 4 shows that optimal decisions often depend on the entire *distribution over outcomes*. Within the expected utility framework, where decision-makers seek to maximise expected utility, we demonstrate that the Pareto front may fail to capture relevant trade-offs. To address this, we introduce the *Distributional Undominated Set* (DUS) and its convex variant (CDUS) and prove that the CDUS consists precisely of those policies that maximise expected utility for multivariate risk-averse decision-makers. We further analyse the relationship between these sets and the Pareto front, develop pruning algorithms that reduce a given set of distributions to a (C)DUS, and extend an existing MORL algorithm to recover a DUS in practice.

The techniques in this chapter build on ideas from distributional RL and show how they can be applied to multi-objective decision-making. Building on this perspective, the final challenge extends the same principle of connection by relating multi-objective games to established single-objective game models.

Equilibria in Multi-Objective Games

> **Challenge 3:** *How can equilibrium solutions with multiple objectives be characterised, and how do they relate to classical game-theoretic models?*
>
> **Example:** Multiple traffic signals operate within the same road network, each seeking to balance efficiency, fairness, and safety at its intersection. Their decisions interact: keeping one light green for too long may create queues that spill into neighbouring junctions. An artificial agent must reason not only about local trade-offs, but also about how the choices of multiple agents combine,

seeking stable outcomes where no signal has an incentive to change its strategy unilaterally.

Chapter 5 examines multi-objective games from a game-theoretic perspective. Such games are known to exhibit surprising non-existence results [Rădulescu et al., 2020b]. We establish a formal equivalence between multi-objective games and a specific class of single-objective games, enabling the transfer of theoretical insights and practical algorithms across these models. Building on this connection, we derive strong existence guarantees for Nash equilibria and demonstrate how established algorithms can be adapted to compute them. While the equivalence is broadly applicable, the resulting equilibria can sometimes remain challenging to analyse and compute. To mitigate this, we identify conditions under which stronger guarantees can be obtained, yielding settings where analysis and computation become more tractable.

In this way, the thesis closes the circle: beginning from single-objective RL methods, we extend to multi-objective and distributional solution concepts, and finally connect these insights to multi-agent interactions through formal equivalences.

5 Reading Guide

This thesis offers a comprehensive account of my doctoral research on multi-objective decision-making. As the work spans several subfields, different readers may engage with different parts depending on their interests and background. The following guide is intended to help you navigate the material.

New to multi-objective decision-making? Begin with the Background chapter (Chapter 2), which introduces the core ideas through practical examples. This chapter provides the necessary foundations in structure, notation, and problem framing, and serves as a starting point even for readers with limited prior exposure to the field. For more technical readers, Chapter 2 may be skimmed and used only as a reference for notation and definitions.

Interested in sequential multi-objective decision-making? If your primary interest lies in sequential settings, start with Chapter 3, which presents methods for learning Pareto fronts of policies, followed by Chapter 4, where we explore how reasoning over full return distributions leads to more robust and principled decision-making. These chapters introduce novel solution concepts and formal tools for understanding optimal behaviour in uncertain, multi-objective environments.

Focused on multi-agent multi-objective systems? Readers interested in strategic interactions between agents may turn directly to Chapter 5. Building on the earlier chapters, this part examines how multiple agents with potentially conflicting objectives interact, how equilibrium solutions can be characterised, and how classical game-theoretic models relate to multi-objective settings.

A note on formalism. The thesis takes a formal approach with definitions, theorems, and proofs provided to ensure clarity and rigour. These details serve to ground the arguments and sharpen the insights, but the central ideas remain accessible through the intuition, examples, and discussion accompanying them. Readers are encouraged to engage with the formalism at their own pace; key messages are reinforced throughout.

A message from the author. Ultimately, this thesis is about understanding what it means to make good decisions in the presence of trade-offs and uncertainty. We all navigate such trade-offs, often without realising it. *Thinking in trade-offs* is the central perspective that runs through this work: there are no universally optimal choices, only solutions that reflect how competing objectives are balanced. This thesis aims to make that reasoning precise.

2

Background

This chapter lays the foundation for the remainder of the thesis, which addresses how to make principled decisions under uncertainty, particularly in the presence of multiple, often conflicting, objectives. Such multi-objective settings arise naturally in many real-world applications, where trade-offs are unavoidable and must be navigated with care [Zintgraf et al., 2018; Deng and Liu, 2018; Biswas et al., 2025; Osika et al., 2025].

Overview. We begin in Section 1 with an introduction to probability theory, which provides the mathematical machinery for representing and reasoning about uncertainty. Section 2 introduces the reinforcement learning (RL) framework, which formalises the process by which agents learn to act optimally through interaction with their environment. Since real-world decision problems rarely involve a single objective, this leads into Section 3, which examines how to formulate and solve decision problems involving multiple objectives. Finally, Section 4 considers environments with multiple interacting agents, extending the discussion to the strategic and game-theoretic dimensions of decision-making.

Motivating example. To ground these ideas, we make repeated use of a stylised example based on adaptive traffic signal control. In this scenario, an artificial agent manages signal phases at an intersection in response to stochastic and dynamic traffic conditions. The problem captures essential features of sequential decision-making under uncertainty and exemplifies the tensions that arise when objectives such as minimising delay and ensuring safety must be simultaneously addressed.

1 A Brief Introduction to Probability

To make principled decisions under uncertainty, we require a precise mathematical framework for describing and reasoning about randomness. Probability theory provides this foundation. It allows us to model uncertain outcomes, assign likelihoods to events, and derive systematic decision rules. In this section, we introduce the core mathematical constructs underpinning probabilistic reasoning. These structures form the basis for the reinforcement learning framework introduced in Section 2, and ultimately for the multi-objective and multi-agent decision problems studied in the thesis.

1.1 Measure Theory and Probability Spaces

Our starting point is measure theory, which formalises the intuitive notion of assigning sizes, or in the case of probability, likelihoods, to sets of outcomes [Klenke, 2020]. While abstract, this machinery enables us to define probabilities in a mathematically rigorous way, and is indispensable for working with general state spaces, continuous random variables, and stochastic processes.

We begin with the notion of a σ*-algebra*, which identifies the collection of subsets of a space that are considered *measurable* [Royden and Fitzpatrick, 1988]. This is a prerequisite for assigning probabilities consistently.

Definition 1: σ-algebra

A collection of subsets Σ of a set $\mathcal{X}$ is a σ*-algebra* if it contains the empty set, is closed under complementation, and is closed under countable unions.

Example 1: σ-algebra over traffic conditions

Let $\mathcal{X} = \{\text{low}, \text{moderate}, \text{high}\}$ represent the possible traffic conditions at a given intersection and consider the collection

$$\Sigma = \{\varnothing, \mathcal{X}, \{\text{low}\}, \{\text{moderate}, \text{high}\}\}.$$

This collection satisfies:

- **Contains $\varnothing$ and $\mathcal{X}$:** Both the empty set and the full set $\mathcal{X}$ are included.
- **Closed under complementation:** The complement of $\{\text{low}\}$ is $\{\text{moderate}, \text{high}\}$, and vice versa.
- **Closed under countable unions:** Any union of sets in Σ is also in Σ, for example, $\{\text{low}\} \cup \{\text{moderate}, \text{high}\} = \mathcal{X}$.

If $\mathcal{X}$ is a topological space, the *Borel σ-algebra* is the smallest σ-algebra containing all open sets of $\mathcal{X}$ and is denoted by $\mathcal{B}(\mathcal{X})$. In discrete countable spaces, the Borel σ-algebra coincides with the *power set* $2^{\mathcal{X}}$, which contains all subsets of $\mathcal{X}$ [Royden and Fitzpatrick, 1988]. Once a σ-algebra is specified, we can define a *measure*, which assigns a non-negative value to each measurable set. This value reflects the size or likelihood of the set, depending on the context.

Definition 2: Measure

Let $\mathcal{X}$ be a set and Σ a σ-algebra over $\mathcal{X}$. A *measure* $\mu : \Sigma \to [0, \infty]$ satisfies:

- **Non-negativity:** $\mu(A) \geq 0$ for all $A \in \Sigma$;
- **Null empty set:** $\mu(\varnothing) = 0$;
- **Countable additivity:** For any countable collection of disjoint sets $\{A_i\}_{i=1}^{\infty} \subseteq \Sigma$,

$$\mu\left(\bigcup_{i=1}^{\infty} A_i\right) = \sum_{i=1}^{\infty} \mu(A_i).$$

A measure is called a *probability measure* if $\mu(\mathcal{X}) = 1$.

Example 2: Probability measure over traffic conditions

From this point onward we take the entire power set, i.e. the set of all subsets, as our default σ-algebra for the remaining examples. Thus, for the set of traffic conditions

$$\mathcal{X} = \{\text{low}, \text{moderate}, \text{high}\}, \qquad \Sigma = 2^{\mathcal{X}}.$$

We can assign probabilities to the three atomic events

$$\mu\big(\{\text{low}\}\big) = 0.6, \qquad \mu\big(\{\text{moderate}\}\big) = 0.3, \qquad \mu\big(\{\text{high}\}\big) = 0.1,$$

and extend μ to every other subset of $\mathcal{X}$ by finite additivity. Because the probabilities of the singletons sum to 1, μ is a probability measure on $(\mathcal{X}, \Sigma)$.

The pair $(\mathcal{X}, \Sigma)$ is called a *measurable space*. A *probability space* augments this with a probability measure, fully specifying a random experiment by defining its possible outcomes, measurable events, and associated probabilities.

Definition 3: Probability space

A *probability space* is a triple (Ω, Σ, μ), where:

- Ω is the sample space of possible outcomes;
- Σ is a σ-algebra over Ω, representing measurable events;
- $\mu : \Sigma \to [0, 1]$ is a probability measure.

Example 3: Probability space for traffic conditions

With the full power set convention adopted in Example 2, set

$$\Omega = \{\text{low}, \text{moderate}, \text{high}\}, \qquad \Sigma = 2^{\Omega},$$

and let μ be the probability measure that assigns $\mu(\{\text{low}\}) = 0.6$, $\mu(\{\text{moderate}\}) = 0.3$, $\mu(\{\text{high}\}) = 0.1$ and extends to all other subsets by additivity. The triple (Ω, Σ, μ) is therefore a probability space representing the uncertainty over the current traffic condition at the intersection.

1.2 Random Variables and Expectation

Probability spaces provide a rigorous foundation for modelling uncertainty, but by themselves they remain abstract: they describe events and their likelihoods without yet giving us a way to quantify outcomes. To analyse decision-making problems, we require a means of mapping uncertain events to numerical values that capture the aspects we care about, such as rewards, costs, queue lengths, or delays. *Random variables* provide exactly this connection, translating uncertainty into measurable quantities from which we can derive expectations and other useful statistics [Billingsley, 1995].

To reason about probabilities of events defined through a random variable, we require the variable to be a *measurable function*. This ensures that statements such as the probability of its value lying in a given range correspond to valid measurable events in the underlying probability space. Formally, a function $f : \mathcal{X} \to \mathcal{Y}$ between measurable spaces $(\mathcal{X}, \Sigma)$ and $(\mathcal{Y}, \Sigma')$ is measurable if, for every $B \in \Sigma'$, the preimage $f^{-1}(B)$ lies in Σ. Thus a random variable is precisely such a measurable function from the sample space Ω to a target space, typically $\mathbb{R}^d$ for real-valued variables.

Definition 4: Random variable

Given a probability space (Ω, Σ, μ) and a measurable space $(\mathcal{Y}, \Sigma')$, a *random variable* is a measurable function $X : \Omega \to \mathcal{Y}$.

Example 4: Number of waiting vehicles as a random variable

We attach a simple queue length to each traffic label by defining

$$X : \Omega \to \mathbb{N}, \qquad X(\omega) = \begin{cases} 2, & \omega = \text{low}, \\ 5, & \omega = \text{moderate}, \\ 10, & \omega = \text{high}, \end{cases}$$

where $\mathbb{N} = \{0, 1, 2, \ldots\}$ is the set of natural numbers. Hence $X(\omega)$ records the typical number of vehicles waiting at the intersection when the traffic condition is $\omega \in \{\text{low}, \text{moderate}, \text{high}\}$.

Because the underlying σ-algebra is the full power set, $\Sigma = 2^{\Omega}$, every subset of Ω is measurable, so X is measurable automatically. For example,

$$X^{-1}(\{2, 10\}) = \{\text{low}, \text{high}\} \in \Sigma.$$

Once we have defined random variables, the next step is to extract meaningful numerical summaries from them. The most fundamental such summary is the *expectation*, which describes the average value the random variable takes under the probability measure. Since all random variables considered in this thesis are real- or vector-valued, we focus directly on random variables taking values in $\mathbb{R}^d$.

Definition 5: Expectation

Let (Ω, Σ, μ) be a probability space and let $X : \Omega \to \mathbb{R}^d$ be an integrable random variable. Its *push-forward (distribution)* is the measure

$$P_X(A) = \mu\big(X^{-1}(A)\big), \qquad A \in \mathcal{B}(\mathbb{R}^d),$$

where $\mathcal{B}(\mathbb{R}^d)$ denotes the Borel σ-algebra. The *expectation* of X is

$$\mathbb{E}[X] = \int_{\mathbb{R}^d} x \, P_X(\mathrm{d}x) = \int_{\Omega} X(\omega) \, \mathrm{d}\mu(\omega).$$

If X takes only countably many values $\mathcal{X} = \{x_1, x_2, \ldots\} \subset \mathbb{R}^d$, this becomes

$$\mathbb{E}[X] = \sum_{x \in \mathcal{X}} x \, P_X(\{x\}) = \sum_{x \in \mathcal{X}} x \, \mu\big(X^{-1}(\{x\})\big).$$

Example 5: Expectation of the number of waiting vehicles

Recall the random variable

$$X(\omega) = \begin{cases} 2, & \omega = \text{low}, \\ 5, & \omega = \text{moderate}, \\ 10, & \omega = \text{high}. \end{cases}$$

Assume the traffic condition probabilities are

$$\mu\big(\{\text{low}\}\big) = 0.6, \qquad \mu\big(\{\text{moderate}\}\big) = 0.3, \qquad \mu\big(\{\text{high}\}\big) = 0.1.$$

The distribution of X is therefore

$$P_X(\{2\}) = 0.6, \qquad P_X(\{5\}) = 0.3, \qquad P_X(\{10\}) = 0.1.$$

Taking the weighted sum gives

$$\mathbb{E}[X] \;=\; 2 \cdot 0.6 \;+\; 5 \cdot 0.3 \;+\; 10 \cdot 0.1 \;=\; 1.2 + 1.5 + 1.0 \;=\; 3.7.$$

Hence, on average, about 3.7 vehicles are waiting at the intersection.

In many situations, extra information allows us to refine our predictions. For instance, knowing that the ground is wet changes our estimate of whether it has rained. The concept of *conditional expectation* formalises this idea, providing a way to update the expectation of a random variable in light of new evidence. Such evidence is represented by another random variable that encodes the observable data on which we condition.

To make this precise, we need a way to describe the information revealed by a random variable. For any random variable $Y : \Omega \to \mathbb{R}^m$, the *$\sigma$-algebra generated by Y*, denoted $\sigma(Y)$, is defined as

$$\sigma(Y) \;=\; \big\{\, Y^{-1}(A) \,:\, A \in \mathcal{B}(\mathbb{R}^m) \big\}. \tag{2.1}$$

It is the smallest σ-algebra with respect to which Y is measurable, and can be interpreted as capturing all information that is observable through Y.

Definition 6: Conditional expectation

Consider a probability space (Ω, Σ, μ). Let $X : \Omega \to \mathbb{R}^d$ be an integrable random variable, and let $Y : \Omega \to \mathbb{R}^m$ be any random variable.

The *conditional expectation* of X given Y, denoted $\mathbb{E}[X \mid Y]$, is the $\sigma(Y)$-measurable random variable that satisfies

$$\int_{Y^{-1}(A)} X \, \mathrm{d}\mu = \int_{Y^{-1}(A)} \mathbb{E}[X \mid Y] \, \mathrm{d}\mu \qquad A \in \mathcal{B}(\mathbb{R}^m).$$

Example 6: Conditional expectation given time of day

Using the same queue-length variable

$$X(\omega) = \begin{cases} 2, & \omega = \text{low}, \\ 5, & \omega = \text{moderate}, \\ 10, & \omega = \text{high}, \end{cases}$$

assume

$$\mu\big(\{\text{low}\}\big) = 0.6, \qquad \mu\big(\{\text{moderate}\}\big) = 0.3, \qquad \mu\big(\{\text{high}\}\big) = 0.1.$$

Suppose we only learn whether the period is *off-peak* or *peak*. Capture this with

$$Y(\omega) = \begin{cases} \text{off-peak}, & \omega = \text{low}, \\ \text{peak}, & \omega \in \{\text{moderate}, \text{high}\}. \end{cases}$$

For $Y = \text{off-peak}$ we must have $\omega = \text{low}$, so

$$\mathbb{E}[X \mid Y = \text{off-peak}] = 2.$$

For $Y = \text{peak}$ the state is either moderate or high. Renormalising the probabilities inside that event,

$$\mu(\text{moderate} \mid \text{peak}) = \frac{0.3}{0.4} = 0.75, \qquad \mu(\text{high} \mid \text{peak}) = \frac{0.1}{0.4} = 0.25,$$

gives

$$\mathbb{E}[X \mid Y = \text{peak}] = 5\,(0.75) + 10\,(0.25) = 6.25.$$

The unconditional mean was $\mathbb{E}[X] = 3.7$; knowing whether the time is off-peak or peak sharpens that estimate to

$$\mathbb{E}[X \mid Y] = \begin{cases} 2, & Y = \text{off-peak}, \\ 6.25, & Y = \text{peak}. \end{cases}$$

The concept of conditional expectation captures how our beliefs about one random variable should change when new information becomes available. The natural counterpart is *independence*, which describes situations where such information is irrelevant. Two random variables X and Y are independent when knowing the value of one does not alter the distribution of the other. Formally, for all measurable sets $A \in \mathcal{B}(\mathbb{R}^d)$ and $B \in \mathcal{B}(\mathbb{R}^m)$,

$$\mu\big(X^{-1}(A) \cap Y^{-1}(B)\big) \;=\; \mu\big(X^{-1}(A)\big) \cdot \mu\big(Y^{-1}(B)\big). \tag{2.2}$$

2 Reinforcement Learning

Having established how to model uncertainty using probability theory, we now turn to the question at the heart of this thesis: how can an agent make good decisions in an uncertain environment? This is the core problem addressed by *reinforcement learning* (RL), a framework for learning how to act optimally through interaction with an environment [Sutton and Barto, 2018]. In RL, the agent does not have access to a complete model of the world but instead learns from experience. It takes actions, observes their outcomes, and adapts its behaviour to maximise a long-term objective. Reinforcement learning thus blends probabilistic modelling, optimisation, and sequential decision-making into a coherent paradigm.

2.1 Markov Decision Processes

To formalise the RL setting, we introduce the notion of a *Markov Decision Process* (MDP) [Puterman, 1994]. MDPs provide the structural backbone of RL: they specify the environment dynamics, how rewards are assigned, and what information the agent has access to. All subsequent concepts in this chapter, and indeed much of this thesis, build on the MDP framework.

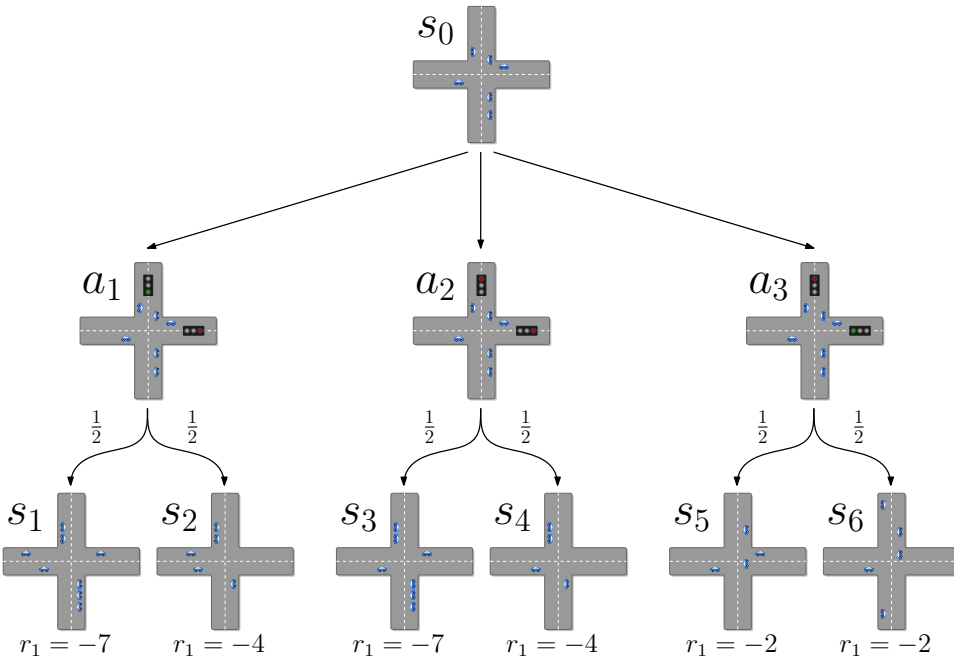


Figure 2.1: The MDP models the stochastic evolution of queue lengths at the intersection: given the current queues and the chosen signal phase, it specifies a distribution over the queues at the next decision instant. Under the policy described in Example 8, which always gives green to the direction with the longer queue, the agent would select a_1.

Definition 7: Markov Decision Process (MDP)

A *Markov Decision Process* (MDP) is a tuple $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, p, r, p_0, \gamma \rangle$, where:

- $\mathcal{S}$ is a set of states;
- $\mathcal{A}$ is a set of actions;
- $p : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is the transition function, where $p(s, a, s')$ is the probability of transitioning from state s to state s' after taking action a;
- $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, where $r(s, a)$ is the immediate reward received after taking action a in state s;
- $p_0 : \mathcal{S} \to [0, 1]$ is the initial state distribution, which specifies the probability of starting in each state;
- $\gamma \in [0, 1)$ is the discount factor, capturing the agent's preference for immediate over future rewards.

Example 7: Adaptive traffic signal control as an MDP

An adaptive controller for a signalised intersection can be naturally formulated as a Markov decision process, providing a formal framework for optimising traffic flow:

- **States** $\mathcal{S}$: each state s records the queue lengths on all incoming lanes, i.e. $s = (q_1, q_2, q_3, q_4)$.
- **Actions** $\mathcal{A}$: the viable signal phases, such as *north-south green*, *east-west green*, and an *all-red clearance*.
- **Transition function** p: the function determines how many vehicles remain in each queue after the chosen phase clears its approach. This depends on how many vehicles were served and how many new ones arrived.
- **Reward function** r: a negative delay cost, $r(s, a) = -\sum_i q_i$, penalising long queues and thereby encouraging smooth traffic flow.
- **Initial state distribution** p_0: a distribution over initial states, e.g. uniform over all queue lengths.
- **Discount factor** γ: a value close to 1 (e.g. 0.99) that emphasises long-run performance rather than short-term gains.

We illustrate this MDP structure in Fig. 2.1.

The MDP defines the environment's structure but does not specify how the agent should act. This role is fulfilled by a *policy*: a rule that determines which action the agent selects based on the information available. If the transition and reward functions are fully known, optimal policies can be computed offline using classical *planning* algorithms [Bertsekas, 2012, 2019]. In RL, however, the agent typically lacks access to these functions and must instead learn a suitable policy through interaction with the environment, improving its decisions based on the observed states, actions, and rewards.

In what follows, we use $\Delta(\mathcal{X})$ to denote the set of probability distributions over a set $\mathcal{X}$. In the finite case, if $\mathcal{X} = \{1, \dots, k\}$, also written as $\mathcal{X} = [k]$, then $\Delta(\mathcal{X})$ corresponds to the standard $(k-1)$-dimensional simplex

$$\Delta^{k-1} := \left\{ x \in \mathbb{R}^k \,\middle|\, x_i \geq 0 \ \forall i,\ \sum_{i=1}^{k} x_i = 1 \right\}. \tag{2.3}$$

Definition 8: Policy

Let the *history* at time t be the sequence of states and actions up to that time,

$$h_t = (s_0, a_0, s_1, a_1, \dots, s_{t-1}, a_{t-1}, s_t).$$

A *policy* $\pi \in \Pi$ is a mapping from the set of all finite histories $\mathcal{H} = \bigcup_{t\geq 0} \mathcal{H}_t$ to distributions over actions:

$$\pi : \mathcal{H} \to \Delta(\mathcal{A}).$$

A *deterministic policy* is a special case where the distribution collapses to a point mass, meaning that for each history h_t, the policy selects a specific action a_t:

$$\pi : \mathcal{H} \to \mathcal{A}.$$

A *stationary policy* depends only on the current state, selecting the same distribution over actions whenever that state is encountered:

$$\pi : \mathcal{S} \to \Delta(\mathcal{A}).$$

Note that these special cases are not mutually exclusive. For instance, a policy can be both deterministic and stationary, i.e., $\pi : \mathcal{S} \to \mathcal{A}$. Depending on the class of policies considered, this combination can have distinct implications for the learning process, both in theory and in practice [Silver et al., 2014; Montenegro et al., 2024; Patil et al., 2024]. Unless stated otherwise, we restrict attention to stationary policies $\pi : \mathcal{S} \to \Delta(\mathcal{A})$.

Example 8: Queue-based deterministic stationary policy

For the intersection MDP of Example 7, consider a policy that always gives a green phase to the direction with the longer queue. Denote by $\sum_{i\in\text{NS}} q_i$ and $\sum_{j\in\text{EW}} q_j$ the total queue lengths on the north-south and east-west approaches, respectively. The policy is

$$\pi(s) = \begin{cases} \text{north-south green}, & \text{if } \displaystyle\sum_{i\in\text{NS}} q_i \geq \sum_{j\in\text{EW}} q_j, \\ \text{east-west green}, & \text{otherwise.} \end{cases}$$

This mapping depends only on the current state and selects a unique action, so it is both stationary and deterministic.

Expected return. Executing a policy in an MDP together with an initial distribution p_0 induces a probability measure $\mathbb{P}^{\pi}_{p_0}$ on the trajectory space

$$\Omega := (\mathcal{S} \times \mathcal{A})^{\mathbb{N}_0},$$

namely the space of infinite state-action sequences equipped with its product σ-algebra. The discounted return

$$Z^{\pi}(\omega) := \sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)$$

is then a random variable $Z^{\pi} : \Omega \to \mathbb{R}$, and the standard RL objective is to identify an *optimal policy* π^* that maximises its expectation

$$J(\pi) = \mathop{\mathbb{E}}_{\pi, s_0 \sim p_0} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \right]. \tag{2.4}$$

Model-based vs model-free RL. In the process of learning an optimal policy, many reinforcement learning algorithms also attempt to construct an explicit model of the environment, typically consisting of estimated transition and reward functions, which can then be used for planning and policy improvement [Ha and Schmidhuber, 2018; Schrittwieser et al., 2020]. This approach is known as *model-based* RL [Moerland et al., 2023]. In contrast, *model-free* RL bypasses explicit modelling and learns policies directly from interaction with the environment [Mnih et al., 2015; Schulman et al., 2017]. The boundary between planning, model-based, and model-free RL is not always sharp, since many modern algorithms combine elements of both [Gelada et al., 2019; Hafner et al., 2021; Delgrange et al., 2023], but the distinction remains a useful conceptual framework for categorising RL methods [Moerland et al., 2022].

2.2 Value-Based Methods

Once a policy has been defined, a natural question arises: how good is this policy, and how can it be improved? Value-based methods provide an answer by quantifying the expected long-term return of following a policy from a given state or after taking a particular action. They thus establish a bridge between policies and learning signals, allowing the quality of decisions to be assessed without requiring an explicit model of the environment.

Value Functions

At the core of value-based methods are the *value functions*, which assign expected returns to states or state-action pairs under a policy and thereby provide the foundation for policy evaluation and improvement. Intuitively, once we can reliably answer the question "how good is this action?", the control problem reduces to selecting the action with the highest value among those available.

Definition 9: Value functions

For any policy π, we define the following functions:

- **State value function** $V^\pi : \mathcal{S} \to \mathbb{R}$, which gives the expected discounted return starting from state s and following policy π thereafter:

$$V^\pi(s) = \mathop{\mathbb{E}}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \mid s_0 = s\right].$$

- **State-action value function** $Q^\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, which gives the expected discounted return starting from state s, taking action a, and following policy π thereafter:

$$Q^\pi(s, a) = \mathop{\mathbb{E}}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \mid s_0 = s, a_0 = a\right].$$

- **Advantage function** $A^\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, which quantifies the relative benefit of taking action a in state s compared to the expected value of the state:

$$A^\pi(s, a) = Q^\pi(s, a) - V^\pi(s).$$

Example 9: Value functions for the queue-based policy

Under the deterministic queue-based policy of Example 8 that always gives a green phase to the direction with the longer queue, consider the state

$$s = (q_{\text{NS}} = 10,\ q_{\text{EW}} = 4),$$

so the total queue length is 14. The reward equals the negative total queue length,

$$r(s, a) \ = \ -\sum_i q_i,$$

and the discount factor is $\gamma = 0.9$.

Assumptions.

- Selecting a green phase clears all vehicles queued on that approach.
- Exactly one new vehicle arrives at each approach before the next decision instant.

State value. Since $q_{\text{NS}} > q_{\text{EW}}$, the policy selects *north-south green*. Clearing the 10 vehicles in the north-south direction and adding one arrival to each queue results in $q_{\text{NS}} = 1$ and $q_{\text{EW}} = 5$, for a total of 6. Repeating this process yields the trajectory

$$14 \longrightarrow 6 \longrightarrow 3 \longrightarrow 3 \longrightarrow \dots,$$

where the state stabilises at a total queue length of 3. The state value is thus

$$\begin{aligned} V^{\pi}(s) &= -14 + \gamma(-6) + \gamma^2\left(\frac{-3}{1-\gamma}\right) \\ &= -14 - 0.9 \times 6 - 0.9^2 \times \frac{3}{0.1} \\ &= -14 - 5.4 - 24.3 \;=\; -43.7. \end{aligned}$$

State-action value. Suppose instead that the agent initially selects *east-west green*. Clearing the 4 vehicles in that direction and adding arrivals leads to $q_{\text{NS}} = 11$ and $q_{\text{EW}} = 1$, for a total of 12. The subsequent evolution is

$$14 \longrightarrow 12 \longrightarrow 3 \longrightarrow 3 \longrightarrow \dots,$$

again converging to a total of 3. The corresponding state-action value is

$$\begin{aligned} Q^{\pi}(s, a_{\text{EW}}) &= -14 + \gamma(-12) + \gamma^2\left(\frac{-3}{1-\gamma}\right) \\ &= -14 - 0.9 \times 12 - 24.3 \;=\; -49.1. \end{aligned}$$

Advantage. The advantage of selecting *east-west green* in state s is

$$\begin{aligned} A^{\pi}(s, a_{\text{EW}}) &= Q^{\pi}(s, a_{\text{EW}}) - V^{\pi}(s) \\ &= -49.1 - (-43.7) \;=\; -5.4. \end{aligned}$$

Thus, deviating from the policy by choosing a_{EW} instead of a_{NS} is suboptimal, with an advantage of -5.4.

Q-Learning

Value functions are useful not only for evaluating a fixed policy but also for improving it, since favouring actions with higher value yields progressively better behaviour. When the transition and reward functions are known, this evaluation-improvement cycle underlies classical planning methods such as value iteration and policy iteration [Bellman, 1957; Howard, 1960].

In RL the model is typically unknown and cannot be queried directly. Agents must therefore estimate value functions from sampled transitions. Q-learning [Watkins, 1989; Watkins and Dayan, 1992] is a model-free algorithm that performs this estimation, learning the optimal state-action value function $Q^*(s, a)$ directly from experience. By repeatedly updating its estimates with samples, it converges to Q^* under suitable conditions [Jaakkola et al., 1994; Tsitsiklis, 1994]. These guarantees assume a sequence of learning rates (α_t) satisfying the classical stochastic approximation conditions $\sum_t \alpha_t = \infty$ and $\sum_t \alpha_t^2 < \infty$ [Robbins and Monro, 1951]. Algorithm 1 outlines the procedure.

Algorithm 1: Q-Learning [Watkins, 1989; Watkins and Dayan, 1992]

Input: learning rate schedule $\alpha(\cdot)$, discount factor γ, exploration schedule $\varepsilon(\cdot)$, episodes N
Result: optimised action-value function Q

initialise $Q(s, a)$ arbitrarily
for $n \leftarrow 1$ **to** N **do**
  reset environment and observe start state s
  while *episode not terminated* **do**
    if Bernoulli($\varepsilon(n)$) $= 1$ **then**
      select a uniformly at random
    else
      $a \leftarrow \arg\max_{a'} Q(s, a')$
    execute a, observe r and s'
    $Q(s, a) \leftarrow Q(s, a) + \alpha(n) \left[r + \gamma \max_{a'} Q(s', a') - Q(s, a) \right]$
    $s \leftarrow s'$

Overview. Q-learning maintains an estimate of the optimal action-value function and updates it incrementally as experience is gathered. Each iteration consists of three essential steps: selecting an action, typically using an *ε-greedy rule* to balance exploration and exploitation, executing it to obtain a transition and reward, and applying a *temporal-difference* update based on the *Bellman optimality operator*. In what follows, we examine these components in more detail, beginning with the Bellman operator, its stochastic approximation, and the role of exploration schedules.

Bellman optimality operator. The Bellman optimality operator $\mathcal{B}$ acts on any candidate function $Q : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ as

$$\big(\mathcal{B}Q\big)(s, a) = \underbrace{r(s, a)}_{\text{immediate reward}} + \underbrace{\gamma \sum_{s'} p(s, a, s') \max_{a'} Q\big(s', a'\big)}_{\text{discounted expected future return}}. \tag{2.5}$$

Intuitively, $\mathcal{B}$ performs a single look-ahead: it combines the immediate reward with the best achievable value in the successor state, weighted by the transition probabilities. Repeated application of $\mathcal{B}$ therefore propagates information backwards through time until the fixed point Q^* is reached.

Stochastic approximation of $\mathcal{B}$. Applying the Bellman operator in its exact form requires knowledge of the transition probabilities, which are not available to the agent in RL. Instead, the agent observes sample transitions (s_t, a_t, r_t, s_{t+1}) during interaction with the environment and uses these to construct a stochastic approximation of Eq. (2.5).

This approximation updates the current estimate based on the temporal-difference (TD) error, which measures the gap between the present estimate and a one-step bootstrap target:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \underbrace{\left[r_t + \gamma \max_{a'} Q\left(s_{t+1}, a'\right) - Q(s_t, a_t) \right]}_{\text{TD error}}, \tag{2.6}$$

where $\alpha \in (0, 1]$ is the learning rate. When the TD error vanishes, the estimate $Q(s_t, a_t)$ satisfies the Bellman equation, and learning can be viewed as gradually driving these discrepancies towards zero across the visited transitions.

Exploration strategy. Because the value estimates are unreliable early in training, effective exploration is essential. A common choice is the ε-greedy strategy: at time t, with probability $\varepsilon(t)$ the agent selects a random action, and with probability $1 - \varepsilon(t)$ it selects the greedy action $\arg\max_a Q(s, a)$. The exploration rate $\varepsilon(t)$ is typically annealed from a high initial value towards zero as learning progresses. Note that some degree of randomisation is critical, both to balance *exploitation* (leveraging current knowledge) with *exploration* (trying new actions) and because theoretical convergence guarantees require that all actions continue to be sampled [Watkins and Dayan, 1992; Jaakkola et al., 1994]. Other exploration strategies are also possible, such as Boltzmann or softmax selection, which samples actions according to a distribution that favours higher-value actions while still allowing exploration [Sutton and Barto, 2018].

Behaviour and target policies. In Q-learning, the actions that generate experience come from the ε-greedy *behaviour policy*, whereas the update target uses $\max_{a'} Q(s', a')$, which corresponds to a greedy *target policy*. Since these policies differ, Q-learning is called an *off-policy* method. By contrast, a closely related variant known as SARSA updates using the value of the actual action taken under the same ε-greedy policy. In this case the behaviour and target policies coincide, making SARSA an *on-policy* algorithm [Rummery and Niranjan, 1994; Sutton, 1995].

Algorithm 2: Deep Q-Network (DQN) [Mnih et al., 2015]

Input: initial network parameters θ, target update period K, replay-buffer capacity C, batch size B
Result: optimised parameters θ

initialise replay buffer $\mathcal{D} \leftarrow \varnothing$; set $\bar{\theta} \leftarrow \theta$
for *interaction step* $t = 1, 2, \ldots$ **do**
  select a_t using an ε-greedy policy w.r.t. Q_θ
  execute a_t, observe r_t and s_{t+1}; store (s_t, a_t, r_t, s_{t+1}) in $\mathcal{D}$
  if $|\mathcal{D}| > C$ **then**
    discard oldest transition
  sample mini-batch $\mathcal{B}$ of transitions from $\mathcal{D}$
  compute targets $y = r + \gamma \max_{a'} Q_{\bar{\theta}}(s', a')$
  update θ by one SGD step on $\frac{1}{B} \sum_{(s,a)\in\mathcal{B}} \big(y - Q_\theta(s, a)\big)^2$
  if $t \bmod K = 0$ **then**
    $\bar{\theta} \leftarrow \theta$

Deep Q-Learning

In the classical tabular version of Q-learning, the action-value function is represented by an explicit table with one entry for each state-action pair. While precise, this approach is only feasible in small, finite environments. Most real-world RL problems involve state spaces that are far too large for such a representation, such as high-dimensional visual inputs in video games or continuous sensor readings in robotics [Bellemare et al., 2013; Makoviychuk et al., 2021]. In these cases, the action-value function is instead approximated by a parametrised function $Q_\theta(s, a)$, typically a deep neural network. The network parameters are trained with stochastic gradient descent [Bottou, 2010], most often using Adam [Kingma and Ba, 2015] or AdamW [Loshchilov and Hutter, 2019]. This shift from tabular to function approximation led to the Deep Q-Network (DQN) algorithm [Mnih et al., 2015], shown in Algorithm 2.

Overview. Similar to Q-learning, DQN aims to learn the optimal state-action value function by following an ε-greedy policy. However, rather than maintaining a tabular representation, each observed transition is stored in a *replay buffer*. During training, batches of transitions are sampled from this buffer and used to update the Q-network parameters θ by minimising the empirical squared TD error,

$$\mathcal{L} = \mathop{\mathbb{E}}_{(s,a,r,s')\sim\mathcal{D}} \left[\left(r + \gamma \max_{a'} Q_{\bar{\theta}}\big(s', a'\big) - Q_\theta(s, a) \right)^2 \right], \tag{2.7}$$

where $\mathcal{D}$ denotes the replay buffer of past transitions and $\bar{\theta}$ are the parameters of a slowly updated *target network*.

Replay buffers. Sequential observations generated by online interaction are strongly correlated and can degrade learning. DQN addresses this by storing transitions in a replay buffer and sampling uniformly from it when updating the network. Drawing random mini-batches breaks temporal correlations and improves data efficiency. Extensions further enhance this idea through prioritised sampling [Schaul et al., 2016] or synthetic augmentation [Lu et al., 2023a; Wang et al., 2024], enabling more effective use of the buffer. Replay buffers also introduce trade-offs, including memory overhead, slower data throughput in large-scale GPU training, and potential non-stationarity since the sampled distribution may lag behind the current policy [Horgan et al., 2018; Fedus et al., 2020].

Target networks. Because the TD target depends on the parameters being optimised, naïve bootstrapping can lead to large oscillations. DQN mitigates this by maintaining a separate target network with parameters $\bar{\theta}$, which are synchronised with θ only every K optimisation steps. This decoupling reduces harmful feedback loops and promotes more stable convergence. An alternative approach uses *soft updates*, where the target network is updated via an exponential moving average of the current parameters:

$$\bar{\theta} \leftarrow \tau\theta + (1-\tau)\bar{\theta}, \tag{2.8}$$

with $\tau \in (0, 1)$ typically set to a small value (e.g. 0.005). This technique, a form of Polyak averaging [Polyak, 1964], is widely adopted in modern reinforcement learning algorithms such as DDPG [Lillicrap et al., 2016] and SAC [Haarnoja et al., 2018].

Variants. Numerous extensions of DQN have sought to improve specific components of the algorithm, including Double DQN [van Hasselt et al., 2016] for reducing overestimation bias, Dueling Networks [Wang et al., 2016] for better state value estimation, C51 [Bellemare et al., 2017] for distributional value learning, and Rainbow [Hessel et al., 2018] which integrates several of these advances. Importantly, some work has also explored avoiding the replay buffer and target network altogether, as these introduce additional memory and implementation complexity [Gallici et al., 2025].

2.3 Policy-Gradient Methods

Value-based algorithms improve behaviour only indirectly: the agent first approximates a value function and subsequently acts greedily with respect to that estimate. In large or continuous action spaces this procedure becomes cumbersome and ties exploration quality to the fidelity of the value approximation. Policy-gradient methods remove this coupling by representing the policy itself as a differentiable mapping π_θ and adjusting θ to maximise the expected return. The policy-gradient theorem [Williams, 1992; Sutton et al.,

1999] gives an expression for $\nabla_\theta J(\theta)$ that avoids differentiating through the dynamics:

$$\nabla_\theta J(\theta) = \mathbb{E}_{\rho^{\pi_\theta}}\big[\nabla_\theta \log \pi_\theta(a_t \mid s_t)\big(Q^{\pi_\theta}(s_t, a_t) - b(s_t)\big)\big]. \tag{2.9}$$

Here the expectation is taken with respect to the discounted state visitation distribution $\rho^{\pi_\theta}(s)$ induced by π_θ. This measure differs from the on-policy distribution induced by π_θ, a subtlety that has important theoretical implications, but is usually ignored in practice [Nota and Thomas, 2020].

Advantage Actor-Critic (A2C). In the policy gradient update, the term $Q^{\pi_\theta}(s_t, a_t) - b(s_t)$ measures how much better an action is relative to a baseline. Choosing the value function as the baseline yields the advantage function, $A^{\pi_\theta}(s_t, a_t)$, and is known as *advantage actor-critic* (A2C) [Mnih et al., 2016]. Using the advantage reduces the variance of gradient estimates while keeping them unbiased. The full procedure is summarised in Algorithm 3.

Proximal Policy Optimisation (PPO). *Proximal Policy Optimisation* (PPO) [Schulman et al., 2017] improves actor-critic training by constraining each update to remain close to the previous policy through a clipped surrogate objective. This restriction prevents overly large parameter changes that might destabilise learning [Schulman et al., 2015]. PPO alternates between collecting trajectories with the current policy and maximising the surrogate objective on mini-batches, thereby balancing exploration and exploitation while preserving stability. Although reported to be challenging to implement correctly [Huang et al., 2022], PPO has become a de facto standard owing to its robustness across diverse domains [Ziegler et al., 2019; Andrychowicz et al., 2020; Li et al., 2025; Huang et al., 2024]. An outline appears in Algorithm 4.

Generalised Advantage Estimation (GAE). In practice, the advantage used in A2C and PPO is not learned directly. Instead, the critic learns the value function V^{π_θ}, and the advantage A^{π_θ} is approximated using *Generalised Advantage Estimation* (GAE) [Schulman et al., 2016]. GAE constructs a weighted sum of multi-step temporal-difference residuals,

$$\hat{A}_t^{\pi_\theta} \;=\; \sum_{l=0}^{\infty} (\gamma\lambda)^l \, \delta_{t+l}, \qquad \delta_t = r_t + \gamma V^{\pi_\theta}(s_{t+1}) - V^{\pi_\theta}(s_t), \tag{2.10}$$

where $\lambda \in [0, 1]$ controls the trade-off between bias and variance. Setting $\lambda = 0$ recovers the one-step TD estimate, while $\lambda = 1$ yields the Monte Carlo return with high variance. In practice, the infinite sum is truncated at the rollout horizon, and intermediate λ values provide a practical compromise, making GAE a standard component in modern policy gradient methods.

Algorithm 3: Advantage Actor-Critic (A2C) [Mnih et al., 2016]

Input: initial actor parameters θ, critic parameters ϕ, discount factor γ, GAE parameter λ, rollout length T
Result: optimised parameters θ, ϕ

repeat
  collect T transitions under π_θ; store (s_t, a_t, r_t, s_{t+1})
  compute advantages A_t using critic V_ϕ and GAE(λ)
  set returns $G_t \leftarrow A_t + V_\phi(s_t)$
  update θ by one gradient ascent step on

$$J^{\text{actor}}(\theta) = \frac{1}{T}\sum_{t=0}^{T-1} A_t \, \log \pi_\theta(a_t \mid s_t)$$

  update ϕ by one gradient descent step on $\frac{1}{T}\sum_{t=0}^{T-1}\big(V_\phi(s_t) - G_t\big)^2$
until convergence

Algorithm 4: Proximal Policy Optimisation (PPO) [Schulman et al., 2017]

Input: initial policy and critic parameters θ, ϕ, clip parameter ε, rollout length T, mini-batch size B, epochs K, discount factor γ, GAE parameter λ
Result: optimised parameters θ, ϕ

set $\theta_{\text{old}} \leftarrow \theta$
repeat
  collect T transitions under $\pi_{\theta_{\text{old}}}$; store $(s_i, a_i, r_i, s_i', \pi_{\theta_{\text{old}}}(a_i \mid s_i))$
  compute advantages A_i using critic V_ϕ and GAE(λ)
  set returns $G_i \leftarrow A_i + V_\phi(s_i)$
  for $k = 1, \dots, K$ **do**
    sample mini-batches $\mathcal{B}$ of size B
    foreach *mini-batch* $\mathcal{B}$ **do**
      compute importance ratios $\rho_i(\theta) = \frac{\pi_\theta(a_i|s_i)}{\pi_{\theta_{\text{old}}}(a_i|s_i)}$ for all $i \in \mathcal{B}$
      update θ by one gradient ascent step on

$$J^{\text{clip}}(\theta) = \frac{1}{|\mathcal{B}|}\sum_{i\in\mathcal{B}} \min\Big(\rho_i(\theta)A_i,\ \text{clip}\big(\rho_i(\theta), 1-\varepsilon, 1+\varepsilon\big)A_i\Big)$$

      update ϕ by one gradient descent step on $\frac{1}{|\mathcal{B}|}\sum_{i\in\mathcal{B}}\big(V_\phi(s_i) - G_i\big)^2$
  set $\theta_{\text{old}} \leftarrow \theta$
until convergence

3 Multi-Objective Decision-Making

Many real-world problems involve multiple, often conflicting objectives, and their complexity is rarely captured by a single scalar reward. In traffic signal control, for example, minimising vehicle delay by rapidly alternating green phases can increase pedestrian risk. These trade-offs motivate the field of *multi-objective decision-making* [Roijers et al., 2013; Greco et al., 2016], which aims to reason about and optimise across several reward dimensions simultaneously.

Vector notation. From this point onward, vectors are written in bold lowercase (e.g. $\boldsymbol{x}$, $\boldsymbol{y}$), with components denoted by subscripts (x_i, y_i). Random vectors are written in bold uppercase such as $\boldsymbol{Z}$. All vectors are assumed to lie in $\mathbb{R}^d$ for an appropriate dimension d. Expressions involving a vector and a scalar are interpreted componentwise: for example, $\boldsymbol{x} + a$ denotes $(x_1 + a, \dots, x_d + a)$. Likewise, operations between two vectors of equal dimension are applied element-wise unless stated otherwise; for instance,

$$\frac{\boldsymbol{x}}{\boldsymbol{y}} := \left(\frac{x_1}{y_1}, \dots, \frac{x_d}{y_d}\right).$$

3.1 Utility-Based Approach

Humans routinely make decisions even in the presence of uncertainty and competing objectives. In decision theory, this capacity is modelled through a *utility function* [von Neumann and Morgenstern, 1944; Kahneman and Tversky, 1979; Keeney and Raiffa, 1993], which encodes preferences over outcomes. In this thesis we adopt a utility-based perspective on multi-objective decision-making: preferences determine which solutions are desirable, rather than an axiomatic focus on any particular solution set [Roijers et al., 2013; Rădulescu et al., 2020a; Dyer, 2005].

The utility-based approach assumes neither explicit symbolic knowledge of the utility function nor direct access to it. Instead, the minimal assumption is that when presented with a set of options, the decision-maker chooses according to an underlying utility function. Formally, this function maps a vector of values to a scalar that expresses the overall evaluation of an outcome and thereby enables comparison between alternatives. Its form is left unrestricted and may be linear or non-linear, depending on the structure of the objectives and the preferences of the decision-maker.

Definition 10: Utility function

A *utility function* is a mapping $u : \mathbb{R}^d \to \mathbb{R}$ that assigns a scalar value to each vector of d objectives. Given two reward vectors $\boldsymbol{r}, \boldsymbol{r}' \in \mathbb{R}^d$, we say that $\boldsymbol{r}$ is preferred to $\boldsymbol{r}'$ (denoted $\boldsymbol{r} \succ \boldsymbol{r}'$) if $u(\boldsymbol{r}) > u(\boldsymbol{r}')$.

Example 10: Utility function in traffic control

Consider a traffic signal controller that must evaluate two possible settings. Each setting yields a pair of outcomes in terms of average *negative delay* (minutes per vehicle) and a *safety score* (on a scale from 0 to 10):

$$\boldsymbol{r}_1 = (-\text{Delay} = -10,\ \text{Safety} = 9),$$
$$\boldsymbol{r}_2 = (-\text{Delay} = -8,\ \text{Safety} = 7).$$

To compare these alternatives, the decision-maker combines the two objectives using a linear utility function

$$u(\boldsymbol{r}) = w_1 \cdot -\text{Delay} + w_2 \cdot \text{Safety},$$

where w_1 and w_2 reflect the relative importance assigned to delay and safety. Choosing $w_1 = 0.3$ and $w_2 = 0.7$ places greater emphasis on safety.

Applying this utility function yields:

$$u(\boldsymbol{r}_1) = 0.3 \times -10 + 0.7 \times 9 = 3.3,$$
$$u(\boldsymbol{r}_2) = 0.3 \times -8 + 0.7 \times 7 = 2.5.$$

Under this utility function, $\boldsymbol{r}_1$ is preferred over $\boldsymbol{r}_2$, illustrating how a decision-maker's weighting of objectives shapes the final choice.

Linear utility. The form of the utility function determines how a decision-maker trades off improvements across objectives. A particularly tractable case is the *linear utility function*:

$$u(\boldsymbol{v}) = \boldsymbol{w}^\top \boldsymbol{v}, \tag{2.11}$$

where $\boldsymbol{w} \in \mathbb{R}^d_{\geq 0}$ are *weights* specifying the relative importance of each objective [Roijers et al., 2013]. In this form, trade-offs are constant: the rate at which one objective can be exchanged for another does not depend on the current outcome [Keeney and Raiffa, 1993; Varian, 2014]. Optimisation also becomes simpler, since maximising u over the feasible region reduces directly to a weighted single-objective problem and coincides with evaluating a convex coverage set [Roijers and Whiteson, 2017]. Owing to these advantages, linear utility remains a central modelling assumption in much of the multi-objective decision-making literature [Roijers, 2016; Yang et al., 2019; Xu et al., 2020; Alegre et al., 2023].

Concave, convex, and quasiconcave utility. Beyond linearity, more general shapes of utility functions capture richer attitudes towards trade-offs. A utility function is *concave*

if for all $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$ and $\lambda \in [0,1]$,

$$u\big(\lambda \boldsymbol{x} + (1-\lambda)\boldsymbol{y}\big) \;\geq\; \lambda\, u(\boldsymbol{x}) + (1-\lambda)\, u(\boldsymbol{y}). \tag{2.12}$$

It is *strictly concave* if the inequality is strict whenever $\boldsymbol{x} \neq \boldsymbol{y}$ and $\lambda \in (0,1)$. A function is *convex* if the reverse inequality holds. Concavity encodes diminishing marginal value and a preference for diversification: averaged outcomes are (weakly) preferred to extremes [Mas-Colell et al., 1995; Varian, 2014]. Convex utilities, by contrast, reflect an anti-diversification attitude: extremes are (weakly) preferred to averages. These shapes are also closely tied to *risk attitudes*, where a concave utility function represents risk aversion, while a convex utility function represents risk-seeking behaviour [Keeney and Raiffa, 1993]. Geometrically, concavity ensures that all upper contour sets $\{\boldsymbol{x} : u(\boldsymbol{x}) \geq \alpha\}$ are convex, facilitating optimisation and yielding stable trade-offs.

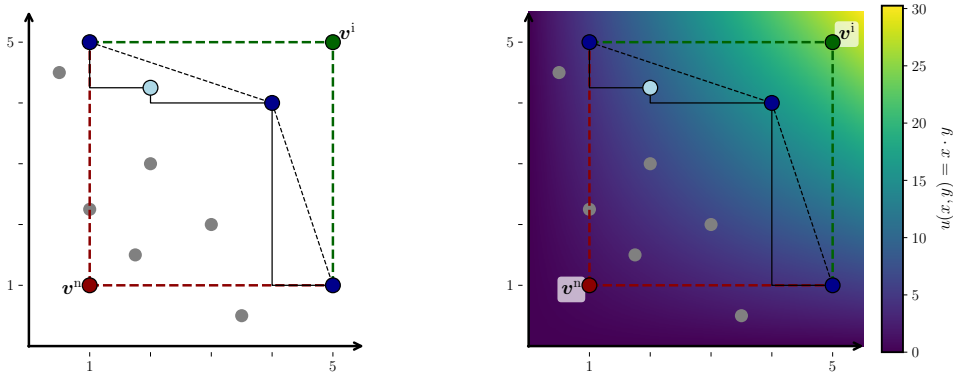
A weaker but widely used condition is *quasiconcavity*, which requires only that for all $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$ and $\lambda \in [0,1]$,

$$u\big(\lambda \boldsymbol{x} + (1-\lambda)\boldsymbol{y}\big) \;\geq\; \min\{u(\boldsymbol{x}), u(\boldsymbol{y})\}. \tag{2.13}$$

Quasiconcavity guarantees convex upper contour sets, but without imposing diminishing marginal returns. Intuitively, it expresses a weak preference for balance: averaged outcomes are never worse than extremes. The dual notion of *quasiconvexity* requires convex lower contour sets, reflecting the opposite bias towards extremes.

Real-world utility functions. Although linear utility is often adopted for analytical and computational convenience [Hayes et al., 2022a], empirical studies consistently show that real decision-makers exhibit non-linear preferences. In travel behaviour, travellers balance time, cost, and comfort in ways that cannot be captured by linear trade-offs [Koppelman, 1981]. Health economics likewise models interventions by combining longevity and quality-of-life through non-linear aggregation [Bleichrodt et al., 1999]. Water resource management requires weighing economic, environmental, and social objectives, where multi-attribute utilities have been explicitly applied [Keeney and Wood, 1977]. In agriculture, non-linear utility functions help farmers navigate between maximising profit, avoiding risk, and minimising indebtedness [André and Riesgo, 2007].

From a modelling perspective, several functional forms have been proposed to capture such behaviours. A first step is to relax linearity while retaining *utility independence*, which ensures that multi-attribute utilities can be decomposed into additive or multiplicative forms [Keeney, 1974; Keeney and Raiffa, 1993]. Additive models with *piecewise-linear marginals* are especially popular, as they flexibly represent diminishing returns while remaining straightforward to elicit [Dyer and Sarin, 1979; Jacquet-Lagrèze et al., 1987; Siskos et al., 2005]. The *Choquet integral* has become a standard tool in modern multi-objective decision-making, allowing the modeller to represent synergies or redundancies between criteria [Grabisch and Labreuche, 2010;



(a) Solution sets

(b) Utility landscape

Figure 2.2: Illustration of two perspectives on multi-objective decision-making. **(a)** The light and dark blue points joined by the solid line form the Pareto front. The dark blue points joined by the dashed line comprise the convex coverage set. Grey points are dominated and excluded from both sets. The green and red points mark the ideal and nadir, respectively, which define opposite corners of the bounding box around the Pareto front. **(b)** A visualisation of a non-linear utility function $u(x, y) = x \cdot y$ over the objective space, with colour denoting utility values.

Corrente et al., 2016]. In economics, concave utility functions have been used to model risk-averse behaviour, capturing the diminishing marginal utility of wealth [Mas-Colell et al., 1995; Varian, 2014].

3.2 Solution Concepts

While the utility-based approach assumes an underlying function guiding decision-making, in practice such a function is often unknown or too complex to elicit, and thus cannot be optimised directly. A common strategy is instead to identify a *solution set* of candidate decisions that capture principled trade-offs across objectives. Such sets do not prescribe a single outcome but guarantee that, once preferences are specified, a suitable option is available. Two central examples are the *Pareto front*, which collects outcomes not dominated on any objective, and the *convex coverage set*, which suffices for all linear utility functions, as illustrated in Fig. 2.2.

Pareto Front

Pareto optimality is a widely used criterion for evaluating solutions. A decision is Pareto optimal if no objective can be improved without worsening at least one other and the set of all such decisions forms the *Pareto front*. Pareto-optimal vectors are especially relevant when the decision-maker's utility function is *monotonically increasing*, since in this case if $\boldsymbol{v} \succeq_{\mathrm{p}} \boldsymbol{v}'$ then $\boldsymbol{v}$ is preferred to $\boldsymbol{v}'$ regardless of how the decision-maker prioritises the objectives.

Definition 11: Pareto dominance, Pareto optimality, and Pareto front

Let $\mathcal{R} \subseteq \mathbb{R}^d$ be a set of vectors. For $\boldsymbol{r}, \boldsymbol{r}' \in \mathcal{R}$, we say $\boldsymbol{r}'$ *Pareto dominates* $\boldsymbol{r}$, written $\boldsymbol{r}' \succ_{\mathrm{p}} \boldsymbol{r}$, if

$$r'_i \geq r_i \text{ for all } i \in \{1, \dots, d\}, \quad \text{and} \quad r'_j > r_j \text{ for at least one } j.$$

A vector $\boldsymbol{r}^* \in \mathcal{R}$ is *Pareto optimal* if no $\boldsymbol{r} \in \mathcal{R}$ satisfies $\boldsymbol{r} \succ_{\mathrm{p}} \boldsymbol{r}^*$.

The *Pareto front* of $\mathcal{R}$, denoted $\mathcal{F}(\mathcal{R})$ or simply $\mathcal{F}$ when $\mathcal{R}$ is clear from context, is the set of all Pareto-optimal vectors:

$$\mathcal{F}(\mathcal{R}) \;=\; \{\, \boldsymbol{r} \in \mathcal{R} \mid \nexists\, \boldsymbol{r}' \in \mathcal{R} \,:\, \boldsymbol{r}' \succ_{\mathrm{p}} \boldsymbol{r} \,\}.$$

Example 11: Pareto front in traffic control

Consider a traffic signal control problem with two objectives: minimising vehicle delay (minutes per vehicle) and maximising safety (on a scale from 0 to 10). We evaluate four candidate control configurations, each yielding the following outcome vectors:

$$\boldsymbol{r}_1 = (-10,\ 9), \qquad \boldsymbol{r}_2 = (-8,\ 7), \qquad \boldsymbol{r}_3 = (-12,\ 7), \qquad \boldsymbol{r}_4 = (-9,\ 7.5).$$

- $\boldsymbol{r}_1$ offers better safety but longer delay compared to $\boldsymbol{r}_2$; neither dominates the other.
- $\boldsymbol{r}_3$ has longer delay than $\boldsymbol{r}_2$ while providing the same safety, so $\boldsymbol{r}_2 \succ_{\mathrm{p}} \boldsymbol{r}_3$.
- $\boldsymbol{r}_4$ is not pairwise dominated, nor does it dominate $\boldsymbol{r}_1$ or $\boldsymbol{r}_2$.

The Pareto front consists of $\boldsymbol{r}_1$, $\boldsymbol{r}_2$, and $\boldsymbol{r}_4$; only $\boldsymbol{r}_3$ is dominated and excluded.

The Pareto landscape. When a vector $\boldsymbol{v}$ Pareto dominates or is equal to another vector $\boldsymbol{v}'$, we write $\boldsymbol{v} \succeq_{\mathrm{p}} \boldsymbol{v}'$. We say that $\boldsymbol{v}$ *strictly Pareto dominates* $\boldsymbol{v}'$, denoted $\boldsymbol{v} > \boldsymbol{v}'$, when

$\forall i \in \{1, \dots, d\} : v_i > v'_i$. A vector is *weakly Pareto optimal* if no other vector strictly Pareto dominates it.

Two reference points associated with the Pareto front are particularly important. The *ideal point* $\boldsymbol{v}^{\text{i}}$ corresponds to the least upper bound over all objectives, while the *nadir point* $\boldsymbol{v}^{\text{n}}$ represents the greatest lower bound (see Fig. 2.2a). These points delineate the range of attainable trade-offs and can be used to constrain the search space [Miettinen, 1998].

Approximating the Pareto front. In practice, two challenges arise. First, the Pareto front may be continuous, making it impossible to enumerate all solutions explicitly. Second, even when the front is finite, computing all Pareto-optimal solutions may be prohibitively expensive, particularly as the number of objectives increases [Bentley et al., 1978; Hughes, 2005]. In both cases, it is common to consider an *approximate Pareto front* $\mathcal{F}^{\tau}$ with tolerance τ. This set satisfies the condition that for every $\boldsymbol{v}^* \in \mathcal{F}^*$, there exists $\boldsymbol{v} \in \mathcal{F}^{\tau}$ such that $\|\boldsymbol{v}^* - \boldsymbol{v}\|_\infty \leq \tau$.

Convex Coverage Set

Rather than exhaustively constructing the entire Pareto front, it is often more practical to compute a *convex coverage set*, which is a solution set for decision-makers with linear preferences [Roijers and Whiteson, 2017]. This set enables efficient exploration of the solution space without requiring explicit identification of every Pareto-optimal point. Notably, when convex combinations of Pareto-optimal solutions yield valid outcomes, the convex coverage set coincides with the full Pareto front [Roijers and Whiteson, 2017].

Definition 12: Convex coverage set

Let $\mathcal{R} \subseteq \mathbb{R}^d$ be a set of vectors. A *convex coverage set* of $\mathcal{R}$, denoted $\mathcal{C}(\mathcal{R})$, is the subset of vectors in $\mathcal{R}$ that are not Pareto dominated by any convex combination of other vectors in the set:

$$\mathcal{C}(\mathcal{R}) = \left\{ \boldsymbol{r} \in \mathcal{R} \;\middle|\; \nexists\, \mu \in \Delta(\mathcal{R}) \text{ such that } \mathop{\mathbb{E}}_{\boldsymbol{r}' \sim \mu} \left[\boldsymbol{r}'\right] \succ_{\text{p}} \boldsymbol{r} \right\}.$$

Example 12: Convex coverage set in traffic control

Consider again the traffic signal control problem from Example 11. As established earlier, $\boldsymbol{r}_3$ is pairwise dominated and excluded from the Pareto front, while $\boldsymbol{r}_1$, $\boldsymbol{r}_2$, and $\boldsymbol{r}_4$ remain.

When considering convex combinations, a 50–50 mixture of $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ yields

$$\frac{1}{2}\boldsymbol{r}_1 + \frac{1}{2}\boldsymbol{r}_2 = \frac{1}{2}(-10,\ 9) + \frac{1}{2}(-8,\ 7) = (-9,\ 8),$$

which Pareto dominates $\boldsymbol{r}_4 = (-9,\ 7.5)$, since the delay is equal but safety is higher. Thus, while $\boldsymbol{r}_4$ is Pareto optimal, it is not included in the convex coverage set.

3.3 Multi-Objective Reinforcement Learning

We focus on decision-making within the framework of *Multi-Objective Reinforcement Learning* (MORL). This section introduces the formalism of *Multi-Objective Markov Decision Processes* (MOMDPs) and explains how the components introduced in Section 2 extend to the multi-objective setting.

Definition 13: Multi-Objective Markov Decision Process (MOMDP)

A Multi-Objective Markov Decision Process (MOMDP) is a tuple $\boldsymbol{\mathcal{M}} = \langle \mathcal{S}, \mathcal{A}, p, \boldsymbol{r}, p_0, \gamma \rangle$, where:

- $\mathcal{S}, \mathcal{A}, p, p_0$ and γ are analogous to the MDP definition;
- $\boldsymbol{r} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$ is the reward function, where d is the number of objectives and $\boldsymbol{r}(s, a)$ is a vector of rewards.

Observe that, to distinguish between scalar and vector-valued reward functions, we denote the latter in bold as $\boldsymbol{r}$.

Example 13: Multi-objective traffic signal control

In the multi-objective extension of the traffic signal control problem, the only change to Example 7 is to the reward function, which is now vector-valued. For each state s and action a, the reward is defined as

$$\boldsymbol{r}(s, a) \ = \ \big(-\,\mathrm{Delay}(s, a),\ \mathrm{Safety}(s, a)\big) \in \mathbb{R}^2.$$

- $\mathrm{Delay}(s, a) = \sum_i q_i$ denotes the total queue length, as in the scalar model. Shorter queues yield higher (less negative) values.
- $\mathrm{Safety}(s, a)$ is the expected fraction of time the intersection remains fully clear after the phase, which increases with longer all-red intervals and conservative timings.

Value functions. With appropriate modifications, the value functions introduced in Definition 9 extend naturally to the multi-objective setting by replacing scalar returns

with vector-valued ones. The underlying structures remain unchanged from the scalar case and are distinguished by bold notation. For example, the multi-objective state-value and state-action value functions are denoted by $\boldsymbol{V}^{\pi}$ and $\boldsymbol{Q}^{\pi}$, respectively.

Scalarising returns. The utility-based approach extends naturally to MORL, where a utility function is applied to vector-valued rewards [Rodriguez-Soto et al., 2024]. A central question, however, is *when* to apply the utility function. This choice is critical in the presence of stochastic rewards, as it influences how outcomes are evaluated and decisions are made. Let $\boldsymbol{X} = (X_1, \ldots, X_d)$ be a random vector of objective values. Its utility can be assessed in two distinct ways:

- *Expected utility:* apply the utility function first, then take the expectation:

$$\mathbb{E}[u(\boldsymbol{X})].$$

- *Utility of the expectation:* take the expectation first, then apply the utility:

$$u\big(\mathbb{E}[\boldsymbol{X}]\big).$$

The former is often referred to as the Expected Scalarised Returns (ESR) criterion, while the latter is known as the Scalarised Expected Returns (SER) criterion [Hayes et al., 2022a]. These two expressions generally diverge, particularly when the utility function is non-linear. ESR captures variability by applying the utility function to each outcome before aggregation, whereas SER ignores variability and evaluates only the mean. Recent work has further clarified the relationship between the two, showing that any policy ordering induced under ESR can be reproduced by a suitable utility function under SER [Subramani et al., 2024]. This distinction between SER and ESR is central to the semantics of optimality in MORL and recurs throughout the theoretical results and algorithms developed in this thesis.

Optimal policies. In the multi-objective setting, the definition of an *optimal policy* must be reconsidered. If the utility function is known, the optimal policy is the one maximising utility under either the ESR or SER criterion. More often, however, the utility function is unavailable, making direct comparison of policies less straightforward. A common approach is to extend dominance relations, such as Pareto dominance, to policies. Under Pareto optimality, policies are compared through their expected return vectors: a policy π *Pareto dominates* π' if $\boldsymbol{v}^{\pi} \succ_{\mathrm{p}} \boldsymbol{v}^{\pi'}$, where

$$\boldsymbol{v}^{\pi} = \mathop{\mathbb{E}}_{s \sim p_0}\left[\boldsymbol{V}^{\pi}(s)\right] \tag{2.14}$$

is the value vector of π, averaged over the initial state distribution p_0. The *Pareto front of policies*, denoted $\mathcal{F}(\Pi)$, is the set of all policies not dominated by any other. Throughout

this thesis, we use $\mathcal{F}$ to refer either to the set of undominated value vectors or to the corresponding Pareto-optimal policies, with the intended meaning clear from context.

Algorithms. A common approach in multi-objective reinforcement learning (MORL) is to condition policies on a *preference embedding*, typically represented as a weight vector when assuming linear utility functions. This idea, known as the use of *extended networks* [Abels et al., 2019], involves modifying standard deep RL algorithms, such as DQN or PPO, to take both the state and the preference embedding as input [Felten et al., 2024]. These networks are trained to produce policies that adapt to the provided preferences, thereby enabling the agent to generate solutions that reflect different trade-offs between objectives.

Although the architectural modifications are often ad hoc, they are guided by the intuition that embedding preferences directly into the network allows for efficient reuse of learned behaviours across the objective space. The resulting algorithms can produce high-quality, diverse solution sets tailored to the decision-maker's preferences.

A distinguishing factor among such methods is how they select the preference embeddings used during training. A common strategy samples preference vectors from a predefined distribution, often uniform over the weight simplex [Yang et al., 2019; Lu et al., 2023b; Basaklar et al., 2023]. Other techniques include evolving promising weights through prediction-guided search [Xu et al., 2020], or using targeted sampling to refine solutions in desired directions [Reymond et al., 2022; Liu et al., 2025]. More principled approaches have recently emerged, where the weight selection mechanism is grounded in theoretical guarantees that ensure recovery of a desired solution set [Alegre et al., 2023].

4 Dealing with Multiple Agents

Agents typically do not operate in isolation. A self-driving car shares the road with other vehicles, and an adaptive traffic signal must coordinate with neighbouring intersections. In many real-world scenarios, multiple agents interact, and their decisions directly influence one another. This brings us to the study of *multi-agent systems* (MAS), where the objective is to analyse how agents can coordinate, compete, or collaborate to achieve their goals [Shoham and Leyton-Brown, 2008].

4.1 Game-Theoretic Foundations

To formally model such interactions, we introduce *normal-form games* (NFGs), which provide a foundational mathematical framework for multi-agent decision-making [Leyton-Brown, 2008]. In an NFG, each agent's payoff depends not only on its own decisions but also on the choices made by others.

Definition 14: Normal-Form Game (NFG)

An NFG is a tuple $\mathcal{G} = \langle N, \mathcal{A}, r \rangle$, where

- N is a finite set of n players;
- $\mathcal{A} = A_1 \times \cdots \times A_n$ is the finite set of joint actions, where A_i denotes the actions available to player i;
- $r = r_1 \times \cdots \times r_n$ is the joint payoff function, where each $r_i : \mathcal{A} \to \mathbb{R}$ gives the payoff to player i.

Players in the game choose their actions simultaneously, and the joint action determines the payoffs for all players.

Example 14: Adaptive traffic signal control as an NFG

Consider two neighbouring intersections, I_1 and I_2, controlling a shared arterial. The players are $N = \{I_1, I_2\}$, and each chooses a phase length from the set $A_1 = A_2 = \{\text{short}, \text{long}\}$. The joint action space is $\mathcal{A} = A_1 \times A_2$.

Each intersection seeks to minimise vehicle delay. Accordingly, the payoff functions are defined as the negative mean delay observed at intersection I_i during the next cycle:

$$r_i(a_1, a_2) = -\text{Delay}_i(a_1, a_2), \qquad i \in \{1, 2\}.$$

We represent this interaction as a normal-form game, with I_1 as the row player and I_2 as the column player. Each cell in the matrix below shows the pair of payoffs (r_1, r_2) corresponding to a joint action.

		I_2: short	I_2: long
I_1	short	$-20, -20$	$-25, -15$
	long	$-15, -25$	$-30, -30$

For example, if I_1 chooses a long phase and I_2 chooses a short phase, the joint action (long, short) yields payoffs of -15 for I_1 and -25 for I_2. If both choose long phases, the joint action (long, long) results in higher delays of -30 for both intersections.

We also consider *continuous games* (CGs), which extend NFGs to continuous action spaces, allowing for a formal treatment of scenarios where players can choose from a continuous space of actions rather than discrete options. We follow the formalisation of continuous games by Stein et al. [2008], while recognising alternative definitions that

impose different assumptions on strategy sets or utilities [Ganzfried, 2021; Hsieh et al., 2021; Adam et al., 2021].

Definition 15: Continuous Game (CG)

A Continuous Game (CG) $\mathcal{C} = \langle N, \mathcal{X}, r \rangle$ is a normal-form game where the action spaces are non-empty compact metric spaces and the payoff functions are continuous.

Example 15: Continuous traffic signal game

Consider two adjacent intersections, I_1 and I_2, each controlling the green phase for their traffic signal. In the continuous game variant, each intersection chooses a green phase duration from a continuous range, e.g. $[10, 60]$ seconds.

Strategies. In Section 2, policies were introduced to describe how agents select actions in MDPs. In game theory, these are typically referred to as *strategies*. A strategy is said to be *pure* when the player deterministically selects a single action, and *mixed* when the player randomises over multiple actions. Throughout this thesis, we adopt the convention that $\pi = (\pi_i, \pi_{-i})$ denotes a *joint strategy*, with π_i representing the strategy of player i and π_{-i} the joint strategy of all other players. Payoff functions are naturally extended to mixed strategies by defining the expected payoff of player i under π as

$$r_i(\pi) = \mathbb{E}_{a \sim \pi}\big[r_i(a)\big]. \tag{2.15}$$

Nash equilibria. Unlike single-agent settings, which focus on optimal policies, multi-agent problems require *equilibrium* concepts that capture the interdependence of agents' decisions. The most fundamental of these is the *Nash equilibrium* [Nash, 1951].

Definition 16: Nash equilibrium

Let $\mathcal{G} = \langle N, \mathcal{A}, r \rangle$ be a game with n players, where each player i has an action space A_i and a payoff function $r_i : \mathcal{A} \to \mathbb{R}$. A joint mixed strategy $\pi^* = (\pi_1^*, \dots, \pi_n^*)$ is a *Nash equilibrium* if, for each player $i \in N$, the strategy π_i^* is a best response to the strategies of the other players:

$$r_i(\pi^*) \geq r_i(\pi_i, \pi_{-i}^*) \quad \text{for all } \pi_i \in \Delta(A_i),$$

If each π_i^* places all probability mass on a single action, the equilibrium is called a *pure-strategy Nash equilibrium*. Otherwise, it is called a *mixed strategy Nash equilibrium*.

Example 16: Nash equilibrium in a NFG

Consider the NFG shown in Example 14, where each intersection I_i aims to minimise delay. The joint action (long, short) is a Nash equilibrium since neither intersection can improve its payoff by unilaterally changing its action:

- If I_1 switches to short, its delay increases from -15 to -20.
- If I_2 switches to long, its delay increases from -25 to -30.

In practice, it is often useful to relax this concept to allow for ε-*Nash equilibria*, where each player can improve their payoff by at most ε by deviating from their strategy. A key property of Nash equilibria is their guaranteed existence in all finite normal-form games and continuous games [Nash, 1951; Glicksberg, 1952]. This foundational result ensures that there always exists a stable outcome in which no player has an incentive to deviate.

4.2 Computational Approaches to Nash Equilibria

Computing Nash equilibria is a central problem in game theory and underpins many multi-agent learning settings. However, the task is computationally demanding: even for two-player games it is PPAD-complete, that is, it belongs to a complexity class capturing problems believed to be intractable in general [Daskalakis et al., 2006]. This difficulty has motivated a variety of algorithmic approaches, which we survey next.

Exact algorithms. Methods such as support enumeration [Dickhaut and Kaplan, 1993; Porter et al., 2008], Lemke-Howson pivoting [Lemke and Howson J. T., 1964], its extensions [Wilson, 1971], and mixed-integer programming formulations [Sandholm et al., 2005] provide exact solutions. Although provably correct, these algorithms scale poorly with the size of the game and are often restricted to two-player settings.

Approximation schemes. When exact computation is infeasible, one may resort to computing ε-Nash equilibria. These methods employ small-support strategies [Lipton et al., 2003], regret minimisation algorithms [Cesa-Bianchi and Lugosi, 2006; Hsieh et al., 2021], or discretisation techniques [Babichenko et al., 2014]. More recent approaches use deep learning to directly approximate Nash equilibria [Duan et al., 2023; Liu et al., 2024]. Such methods balance computational tractability with solution quality and often provide explicit guarantees on the approximation error ε.

Learning-based techniques. Learning dynamics iteratively approximate equilibria through repeated interaction. Among the most well-known examples is replicator dynamics, a family of continuous-time methods grounded in evolutionary game theory [Mertikopoulos and Sandholm, 2016]. In this thesis, we focus on classical learning-based methods that assume full access to the payoff matrix and aim to compute exact mixed strategies. Our primary baseline is *fictitious play* (FP) [Robinson, 1951; Ganzfried,

2021], where each player repeatedly best responds to the empirical distribution of the opponent's past actions. The algorithm is summarised in Algorithm 5.

Algorithm 5: Two-Player Fictitious Play [Robinson, 1951]

Input: A two-player NFG $\mathcal{G} = \langle N, \mathcal{A}, r \rangle$ and horizon T
Output: Mixed strategy profile $\pi = (\pi_1, \pi_2)$

for $i \in \{1, 2\}$ **do**
  Initialise action counts $h_i(a_i) \leftarrow 0$ for all $a_i \in A_i$
for $t = 1$ **to** T **do**
  for $i \in \{1, 2\}$ **do**
    Let $j = 3 - i$
    Define opponent's empirical strategy: $\tilde{\pi}_j(a_j) \leftarrow \frac{h_j(a_j)}{t}$ for all $a_j \in A_j$
    Compute best response

$$a_i \leftarrow \arg\max_{a_i \in A_i} \sum_{a_j \in A_j} \tilde{\pi}_j(a_j) r_i(a_i, a_j)$$

  Play joint action (a_1, a_2)
  Update counts: $h_1(a_1) \leftarrow h_1(a_1) + 1$, $h_2(a_2) \leftarrow h_2(a_2) + 1$
return $\pi = \left(\frac{h_1}{T}, \frac{h_2}{T}\right)$

4.3 Multi-Objective Games

As the focus of this thesis is on multi-objective decision-making, we extend the normal-form game framework to accommodate multiple objectives. This leads us to *multi-objective normal-form games* (MONFGs) [Blackwell, 1954], where each player's payoff is a vector-valued function of the joint actions.

Definition 17: Multi-Objective Normal-Form Game (MONFG)

A Multi-Objective Normal-Form Game (MONFG) $\mathcal{G} = \langle N, \mathcal{A}, \boldsymbol{r} \rangle$ is a normal-form game where the reward function is vector-valued.

Example 17: Multi-objective traffic signal game

Consider two intersections I_1 and I_2, each selecting a green phase duration. In the *multi-objective normal-form game* setting, each intersection receives a vector-valued

reward consisting of its negative delay and a safety score,

$$\boldsymbol{r}_i(a_1, a_2) = \left(-\text{Delay}_i(a_1, a_2),\ \text{Safety}_i(a_1, a_2)\right),$$

capturing the trade-off between efficiency and safety. Delays are measured in vehicle-minutes, while the safety component is scaled to a value in $[0, 10]$.

Assume both players choose between *short* and *long* phases. The resulting multi-objective payoff matrix is:

		I_2: short	I_2: long
I_1	short	$(-20,\ 9.5),\ (-20,\ 9.3)$	$(-25,\ 9.2),\ (-15,\ 9.6)$
	long	$(-15,\ 9.0),\ (-25,\ 9.1)$	$(-30,\ 8.8),\ (-30,\ 8.9)$

For example, (long, long) yields delays of -30 and safety indices of 8.8 and 8.9 for I_1 and I_2, respectively. The joint action (short, long) results in delays of 25 and 15 minutes, with safety indices of 9.2 and 9.6.

Similar to MORL, multi-objective games complicate the notion of optimality as a strategy may turn out to improve on some objectives while worsening others. We therefore differentiate between *utility-agnostic* and *utility-aware* equilibria [Wang, 2025].

Utility-agnostic equilibria. Two equilibrium concepts in multi-objective games mirror the solution sets in multi-objective optimisation. A *Pareto-Nash equilibrium* extends Pareto optimality to strategic settings: a joint strategy is an equilibrium if no player can deviate unilaterally to one that Pareto dominates their current payoff vector [Shapley and Rigby, 1959; Zeleny, 1975; Lozovanu et al., 2005; Somasundaram and Baras, 2009; Ismaili, 2018]. The strategic analogue of the convex coverage set is the *weighted Nash equilibrium*. Here, each player applies a linear scalarisation of their objectives and plays a Nash equilibrium in the resulting scalar game [Wang, 1993; Corley, 1985; Yu and Yuan, 1998; Qu et al., 2015].

We note that most existing work evaluates strategies in terms of their expected returns (SER). For tractability, these analyses typically also restrict to linear utility functions, under which SER reduces to ESR and the setting collapses to a standard normal-form game. As a result, utility-agnostic equilibria under the true ESR formulation remain largely unexplored.

Utility-aware equilibria. When the utility function is directly available, the standard notion of Nash equilibrium can be extended to incorporate it. This can be done either by applying the utility to the expected payoff (SER) or by applying it to each realised payoff before taking the expectation (ESR). While ESR reduces to a standard normal-form game, the SER formulation does not, and Nash equilibria are not guaranteed to exist in

this case [Rădulescu et al., 2020b]. This discrepancy warrants careful consideration and is the main focus of Chapter 5. Beyond Nash equilibria, *correlated equilibria* [Aumann, 1974] have also been examined in multi-objective games [Rădulescu et al., 2020b]. In cooperative contexts, coalition formation has been studied where agents must form groups based on private utilities derived from publicly observable values of the relevant objectives [Igarashi and Roijers, 2017].

3 Learning a Pareto Front of Policies

This chapter is based on *Divide and Conquer: Provably Unveiling the Pareto Front with Multi-Objective Reinforcement Learning* [Röpke et al., 2025], a project initiated during a research visit at the University of Galway. An earlier version was presented at the European Workshop on Reinforcement Learning (EWRL) 2024. All proofs are integrated into the main text and updated for improved presentation. Our code is available at `https://github.com/wilrop/ipro`.

1 Introduction

Many sequential decision-making problems involve multiple, often conflicting, objectives. For instance, managing a water reservoir requires balancing environmental, economic, and social considerations [Castelletti et al., 2013]. In such cases, decision-makers must ultimately determine an appropriate trade-off between competing goals. Multi-Objective Reinforcement Learning (MORL) provides a principled framework for computing a diverse set of candidate policies that represent the best available trade-offs, enabling informed selection based on individual preferences [Hayes et al., 2022a].

This chapter addresses the following challenge introduced in Chapter 1:

> **Challenge 1:** *How can recent advances in single-objective RL be leveraged to efficiently learn a Pareto front?*

We focus on learning the *Pareto front*, defined as the set of policies whose expected returns are not dominated by any other policy. When stochastic policies are allowed, the Pareto front is convex [Roijers and Whiteson, 2017], which enables the use of established solution methods, e.g. [Yang et al., 2019; Xu et al., 2020; Alegre et al., 2023]. However, deterministic policies are often preferred for reasons of safety, accountability, or interpretability, and in such cases, the Pareto front may exhibit concave regions. Algorithms addressing this setting have been elusive, with successful solutions limited to purely deterministic environments [Reymond et al., 2022].

To overcome this limitation, we introduce *Iterated Pareto Referent Optimisation* (IPRO), a general-purpose algorithm that decomposes the task of learning the Pareto front into a sequence of *constrained* single-objective problems. Decomposition is a well-established paradigm in multi-objective optimisation (MOO), allowing the application of powerful single-objective solvers to each subproblem [Zhang and Li, 2007; Roijers, 2016]. Existing MORL algorithms for convex fronts adopt a similar strategy by applying single-objective reinforcement learning to solve scalarised variants of the original problem [Lu et al., 2023b; Alegre et al., 2023].

IPRO builds on this idea and extends it to the general case. It is an anytime algorithm that incrementally constructs the Pareto front by iteratively identifying new Pareto-optimal policies. Each such policy is obtained by solving a constrained single-objective problem for which principled methods are derived. Our theoretical results show that IPRO guarantees convergence to a Pareto front and provides an explicit upper bound on the distance to undiscovered solutions at each step. We further show that, for a fixed number of objectives, the algorithm requires only a polynomial number of iterations to approximate this front. Although IPRO applies to arbitrary policy classes, we highlight its practical benefits for deterministic policies. Empirical results demonstrate that IPRO performs competitively with or outperforms existing algorithms, even when these rely on stronger assumptions or domain-specific knowledge.

Contributions. The main contributions of this chapter are as follows:

1. **IPRO: a decomposition-based MORL algorithm.** We propose IPRO, which reduces the task of learning the Pareto front to a sequence of constrained single-objective problems that can be solved with standard RL methods.

2. **Anytime guarantees.** IPRO maintains guarantees throughout its execution. We prove convergence to the true Pareto front in the limit and provide an explicit upper bound on the number of iterations required for approximate solutions.

3. **Generality.** Unlike prior work, IPRO applies to arbitrary MOMDPs and policy classes, and inherits the guarantees of the single-objective solver used within.

4. **Strong empirical performance.** We demonstrate that IPRO achieves state-of-the-art performance, matching or exceeding methods that rely on additional structural assumptions or domain knowledge.

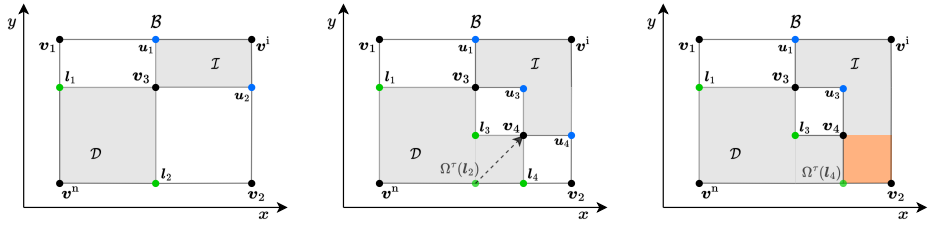
Related work. IPRO decomposes obtaining a Pareto front into an *outer loop* that formulates subproblems and an *inner loop* that solves each using a single-objective algorithm. When learning a single policy in MOMDPs, conventional methods often adapt single-objective RL algorithms. For example, Siddique et al. [2020] extend DQN, A2C and PPO to learn a fair policy by optimising the generalised Gini index of the expected returns. Reymond et al. [2023] extend this to general non-linear functions and establish a policy gradient theorem for this setting. When maximising a concave function of the expected returns, efficient methods exist that guarantee global convergence [Zhang et al., 2020; Zahavy et al., 2021; Geist et al., 2022].

Decomposition is a promising technique for MORL due to its ability to leverage strong single-objective methods as a subroutine [Felten et al., 2024]. When the Pareto front is convex, many techniques rely on the fact that it can be decomposed into a sequence of single-objective RL problems where the scalar reward is a convex combination of the original reward vector [Yang et al., 2019; Alegre et al., 2023]. When the Pareto front is non-convex, Van Moffaert et al. [2013] learn deterministic policies on the Pareto front by decomposing the problem using the Chebyshev scalarisation function but do not provide any theoretical guarantees and only evaluate on discrete settings.

In MOO, a related methodology was proposed by Legriel et al. [2010] to obtain approximate Pareto fronts. Their approach iteratively proposes queries to an oracle and uses the return value to trim sections from the search space. In contrast, we introduce an alternative technique for query selection that ensures convergence to the true Pareto front and aims to minimise the number of iterations. Moreover, we introduce a procedure that deals with imperfect oracles and contribute novel results that are particularly useful for MORL.

2 Iterated Pareto Referent Optimisation

We now present *Iterated Pareto Referent Optimisation* (IPRO), a general-purpose algorithm for learning the Pareto front in MOMDPs. IPRO constructs a sequence of constrained single-objective subproblems and maintains sets of lower and upper bounds that enclose the current approximation of the Pareto front. An illustrative example is shown in Fig. 3.1. For all definitions of terms used in this section, we refer to Chapter 2 Section 3.



(a) (b) (c)

Figure 3.1: (**a**) The bounding box $\mathcal{B}$, defined by the nadir $\boldsymbol{v}^{\mathrm{n}}$ and ideal $\boldsymbol{v}^{\mathrm{i}}$, contains all Pareto-optimal solutions. The dominated set $\mathcal{D}$ and infeasible set $\mathcal{I}$ are defined by the current approximation to the Pareto front $\mathcal{F} = \{\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3\}$ and are shaded. The lower bounds $\boldsymbol{l} \in \mathcal{L}$ are highlighted in green, while the upper bounds $\boldsymbol{u} \in \mathcal{U}$ are highlighted in blue. (**b**) After querying the Pareto oracle Ω^{τ} with $\boldsymbol{l}_2, \boldsymbol{v}_4$ is added to the Pareto front and $\mathcal{L}$ and $\mathcal{U}$ are updated to represent the new corners of $\mathcal{D}$ and $\mathcal{I}$ respectively. (**c**) When the Pareto oracle cannot find a feasible solution strictly dominating $\boldsymbol{l}_4$, it is added to the completed set $\mathcal{C}$ and the shaded orange area is added to the infeasible set $\mathcal{I}$.

2.1 Algorithm Overview

The core idea behind IPRO is to maintain explicit boundaries of the search space that may contain value vectors corresponding to undiscovered Pareto-optimal policies, and to iteratively eliminate regions of this space using information from a *Pareto oracle*. Each iteration selects a candidate region, queries the oracle for a new solution, and updates the bounding structures accordingly. A detailed description is provided in Algorithm 6.

Bounding the search space. IPRO first identifies a bounding region $\mathcal{B}$ that encloses all potentially Pareto-optimal value vectors. This region is defined by the *ideal point* $\boldsymbol{v}^{\mathrm{i}}$ and the *nadir point* $\boldsymbol{v}^{\mathrm{n}}$ (Fig. 3.1a), which represent respectively the least upper and greatest lower bounds across all objectives. The ideal point is constructed by maximising each objective independently, yielding d value vectors that are also added to the initial approximation set $\mathcal{F}$. Because computing the exact nadir is generally intractable [Miettinen, 1998], we approximate it by minimising each objective in isolation. To ensure a strict lower bound on $\boldsymbol{v}^{\mathrm{n}}$, as required for theoretical correctness (see Section 3), we subtract a small constant from each coordinate of the resulting vector.

Once $\boldsymbol{v}^{\mathrm{i}}$ and the lower bound on $\boldsymbol{v}^{\mathrm{n}}$ are established, we immediately check whether the initial approximation $\mathcal{F}$ contains a single policy that simultaneously maximises all objectives. If so, this single policy constitutes the full Pareto front and can be returned directly. The check is performed using a pruning algorithm such as PPrune [Roijers and Whiteson, 2017].

Algorithm 6: The IPRO algorithm.

Input: A Pareto oracle Ω^τ with tolerance τ
Output: A τ-Pareto front $\mathcal{F}$

Get maximal points $\{\boldsymbol{v}^1, \dots, \boldsymbol{v}^d\}$ to create the ideal $\boldsymbol{v}^{\mathrm{i}}$
Get minimal points to estimate the nadir $\boldsymbol{v}^{\mathrm{n}}$
Form a bounding box $\mathcal{B}$ from $\boldsymbol{v}^{\mathrm{n}}$ and $\boldsymbol{v}^{\mathrm{i}}$
$\mathcal{U} \leftarrow \{\boldsymbol{v}^{\mathrm{i}}\}$, $\mathcal{L} \leftarrow \{\boldsymbol{v}^{\mathrm{n}}\}$
$\mathcal{F} \leftarrow \textsc{PPrune}\big(\{\boldsymbol{v}^1, \dots, \boldsymbol{v}^d\}\big)$ and $\mathcal{C} \leftarrow \varnothing$
if $|\mathcal{F}| = 1$ **then**
    return $\mathcal{F}$
for $\boldsymbol{v} \in \{\boldsymbol{v}^1, \dots, \boldsymbol{v}^d\}$ **do**
    $\mathcal{L} \leftarrow \textsc{update}(\boldsymbol{v}, \mathcal{L})$
while $\max_{\boldsymbol{u} \in \mathcal{U}} \min_{\boldsymbol{v}' \in \mathcal{F}} \|\boldsymbol{u} - \boldsymbol{v}'\|_\infty > \tau$ **do**
    $\boldsymbol{l} \leftarrow \textsc{Select}(\mathcal{L})$
    $\textsc{success}, \boldsymbol{v}^* \leftarrow \Omega^\tau(\boldsymbol{l})$
    if *success* **then**
        $\mathcal{F} \leftarrow \mathcal{F} \cup \{\boldsymbol{v}^*\}$
        $\mathcal{L} \leftarrow \textsc{update}(\boldsymbol{v}^*, \mathcal{L})$, $\mathcal{U} \leftarrow -\textsc{update}(-\boldsymbol{v}^*, -\mathcal{U})$
    else
        $\mathcal{C} \leftarrow \mathcal{C} \cup \{\boldsymbol{l}\}$
        $\mathcal{L} \leftarrow \mathcal{L} \setminus \{\boldsymbol{l}\}$, $\mathcal{U} \leftarrow -\textsc{update}(-\boldsymbol{l}, -\mathcal{U})$
return $\mathcal{F}$
Function *update*$(\boldsymbol{v}^*, \mathcal{X})$**:**
    $\mathcal{X}' \leftarrow \{\}$
    for $\boldsymbol{v} \in \mathcal{X}$ **do**
        if $\boldsymbol{v}^* > \boldsymbol{v}$ **then**
            $\mathcal{X}' \leftarrow \mathcal{X}' \cup \{(\boldsymbol{v}_{-j}, \boldsymbol{v}^*_j) \mid j \in [d]\}$
        else
            $\mathcal{X}' \leftarrow \mathcal{X}' \cup \{\boldsymbol{v}\}$
    $\mathcal{X}' \leftarrow \textsc{Prune}(\mathcal{X}')$
    return $\mathcal{X}'$

Obtaining a Pareto-optimal policy. To explore the Pareto front, IPRO relies on a *Pareto oracle* (defined in Section 2.2) to produce a policy whose value vector lies in an unexplored *target region* of the objective space. Suppose the user specifies a tolerance $\tau > 0$, indicating the desired resolution for a τ-Pareto front $\mathcal{F}^\tau$. In this setting, the

oracle receives a *referent* $\boldsymbol{r}$ and attempts to return a Pareto-optimal value vector $\boldsymbol{v}^\pi$ that improves at least τ over the referent $\boldsymbol{r}$, i.e., $\boldsymbol{v}^\pi \succeq_\mathrm{p} \boldsymbol{r} + \tau$.

If successful (Fig. 3.1b), the returned solution extends the approximation $\mathcal{F}$, and the space dominated by $\boldsymbol{v}^\pi$ can be safely excluded from further consideration. Moreover, any point that would dominate $\boldsymbol{v}^\pi$ must be infeasible, as it would have been returned instead. Conversely, if no such solution is found (Fig. 3.1c), all points dominating $\boldsymbol{r}$ can be ruled out: they are either infeasible or already covered within tolerance by an existing solution. This process allows IPRO to efficiently prune the objective space and incrementally refine its approximation of the Pareto front.

Tracking the boundaries. To represent the ruled-out regions of the objective space, IPRO tracks two sets. The *dominated set* $\mathcal{D}$ contains points that are dominated by the current approximation $\mathcal{F}$, while the *infeasible set* $\mathcal{I}$ includes points that no feasible policy can reach. Both are illustrated in Fig. 3.1a.

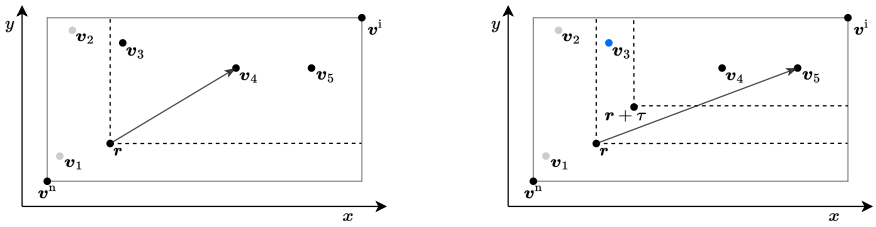
Rather than representing $\mathcal{D}$ and $\mathcal{I}$ explicitly, IPRO maintains compact representations of their boundaries. A set of *lower bounds* $\mathcal{L}$ captures the inner corners of $\mathcal{D}$, while a set of *upper bounds* $\mathcal{U}$ does the same for $\mathcal{I}$. These summaries are sufficient to localise the remaining search space: any undiscovered Pareto-optimal solution must strictly dominate some $\boldsymbol{l} \in \mathcal{L}$ and be dominated by some $\boldsymbol{u} \in \mathcal{U}$.

Iterative refinement. In principle, one could randomly select referents within the bounding box and expand $\mathcal{D}$ and $\mathcal{I}$ via repeated oracle queries. However, this would be inefficient when oracle evaluations are expensive, such as when learning policies in a MOMDP. Instead, IPRO selects referents from $\mathcal{L}$, ensuring that each query targets a maximally unexplored region.

After each oracle call, the sets $\mathcal{F}$, $\mathcal{L}$, and $\mathcal{U}$ are updated depending on the result. If the oracle returns a new solution $\boldsymbol{v}^*$, it is added to $\mathcal{F}$, and the region it dominates is incorporated into $\mathcal{D}$, refining the boundary captured by $\mathcal{L}$. Conversely, if the oracle fails, the region strictly dominating the referent is added to $\mathcal{I}$, and the referent is removed from $\mathcal{L}$ and placed into the *completed set* $\mathcal{C}$. In both cases, $\mathcal{U}$ is also updated to reflect the new infeasible region (see Figs. 3.1b and 3.1c).

This process continues until every $\boldsymbol{u} \in \mathcal{U}$ is within τ of some solution in $\mathcal{F}$, ensuring that the returned set constitutes a τ-Pareto front. In practice, IPRO prioritises referents from $\mathcal{L}$ using a hypervolume-based heuristic, favouring those with the greatest potential to improve coverage.

The bi-objective case. When there are two objectives, the structure of the search space simplifies considerably. Each unexplored region corresponds to an axis-aligned rectangle, defined by a lower bound $\boldsymbol{l} \in \mathcal{L}$ and an upper bound $\boldsymbol{u} \in \mathcal{U}$ (Fig. 3.1). Adding a new solution modifies only the local region, requiring at most two new points to be added to the boundary. This results in a particularly efficient variant, referred to as IPRO-2D.



(a) A weak Pareto oracle.

(b) An approximate Pareto oracle.

Figure 3.2: Solutions inside the target region are black, while solutions outside the target region are grey. **(a)** The weak Pareto oracle returns $\boldsymbol{v}_4$, which is in the target region but is only weakly Pareto optimal as it is dominated by $\boldsymbol{v}_5$. **(b)** The approximate Pareto oracle returns a Pareto-optimal solution $\boldsymbol{v}_5$, but cannot find $\boldsymbol{v}_3$, shown in blue.

The rectangular structure allows for several algorithmic simplifications. The distance between bounds reduces to either the width or height of the rectangle, eliminating the need for the max-min computation used in higher dimensions (see line 10 in Algorithm 6). Moreover, the area of each rectangle can be computed directly, allowing IPRO-2D to prioritise regions by size. This supports a priority queue implementation that greedily selects the region with the largest remaining error. These geometric features allow IPRO-2D to reduce the approximation error quickly while minimising oracle calls.

2.2 Pareto Oracle

A central component of IPRO is the *Pareto oracle*, which receives a referent and returns a Pareto-optimal policy whose value vector strictly dominates it, if such a policy exists. This mechanism enables IPRO to explore specific regions of the objective space and incrementally build the Pareto front. We define two oracle variants, which differ in the guarantees they provide and the precision with which they adhere to the target region. To illustrate their difference, we show a possible evaluation of both oracles in Fig. 3.2. We note that related concepts have been studied in multi-objective optimisation [Papadimitriou and Yannakakis, 2000] and planning [Chatterjee et al., 2006].

Recovering the true Pareto front. The Pareto oracle aims to return a solution that strictly dominates the referent and is Pareto optimal. While such an oracle would be exact, it cannot always be realised in practice. In particular, for achievement scalarising functions (see Section 4), it is known that strict improvement over the referent cannot generally be guaranteed together with Pareto optimality [Wierzbicki, 1982]. To ensure that the search process remains well-defined, we therefore need to relax either the quality

of the returned solution or the region in which it must be found[1]. When the goal is to recover the true Pareto front, however, the region around the referent must be respected without any tolerance, leaving no room to relax the search region. Consequently, we relax the solution quality instead and only require that the returned point be weakly Pareto optimal. This leads to the definition of the *weak Pareto oracle*.

Definition 18: Weak Pareto oracle

A weak Pareto oracle Ω^τ with tolerance $\tau = 0$ maps a referent $\boldsymbol{r} \in \mathbb{R}^d$ to a weakly Pareto-optimal policy $\pi \in \Pi$ such that $\boldsymbol{v}^\pi > \boldsymbol{r}$, or returns FALSE when no such policy exists.

Although weakly Pareto-optimal solutions may still be dominated by others within the same region, the oracle guarantees that each result strictly extends the covered front. This suffices to construct the true Pareto front incrementally, and enables IPRO to converge in the limit under reasonable assumptions (see Section 3.1). Moreover, when the Pareto front forms a continuous path, as in MOMDPs under stationary policies [Altman, 1999; Lu et al., 2023b], any weakly optimal solution that strictly dominates the referent is in fact Pareto optimal, making the weak condition equivalent to true optimality in this setting.

Obtaining an approximate Pareto front. For most applications, recovering the exact Pareto front is unnecessary or impractical and the decision-maker may be satisfied with any solution within a given tolerance τ. In such settings, it is desirable for each oracle query to yield meaningful progress towards a bounded approximation. To this end, we define the *approximate Pareto oracle*, which guarantees that returned solutions are Pareto optimal and enforces a user-specified improvement threshold over the referent.

Definition 19: Approximate Pareto oracle

An approximate Pareto oracle Ω^τ with tolerance $\tau > 0$ maps a referent $\boldsymbol{r} \in \mathbb{R}^d$ to a Pareto-optimal policy $\pi \in \Pi$ such that $\boldsymbol{v}^\pi \succeq_{\mathrm{p}} \boldsymbol{r} + \tau$ or returns FALSE when no such policy exists.

Setting $\tau = 0$ is technically permissible, but it breaks the progress guarantee required by IPRO: since our lower bounds are constructed from previously found Pareto-optimal points, $\tau = 0$ would make an already discovered Pareto-optimal solution feasible again in the target region. Returning such a solution yields no improvement and can lead to

[1] In finite, model-based MDPs, one could in principle implement an exact Pareto oracle by solving a linear program over occupancy measures that searches for a value vector $\boldsymbol{v}$ with $v_i > r_i$ and maximises a strictly positive scalarisation. Strict inequalities can be reduced to ordinary linear programs in polynomial time [Papadimitriou and Steiglitz, 1998, Chapter 8.7.1], and linear programming itself admits polynomial-time algorithms. We do not pursue this construction, as it would restrict our framework to discrete environments with known dynamics.

cycling. Requiring $\tau > 0$ rules this out, at the cost that Pareto-optimal points within the resulting gap may be missed.

2.3 Dealing with Imperfect Pareto Oracles

While IPRO relies on a Pareto oracle that solves the scalar problem exactly, this condition cannot always be guaranteed in practice when dealing with function approximators or heuristic solvers. To overcome this, we introduce a backtracking procedure that maintains the sequence $\{(\boldsymbol{l}_t, \boldsymbol{v}_t)\}_{t\in\mathbb{N}}$ of lower bounds and retrieved solutions in each iteration. When, at iteration n, the returned solution $\boldsymbol{v}_n$ strictly dominates a point $\boldsymbol{c} \in \mathcal{C}_n$ or $\boldsymbol{v}^* \in \mathcal{F}_n$, it indicates an incorrect oracle evaluation in a previous iteration and we initiate a replay of the sequence.

Let $\bar{t}$ represent the time step when the incorrect result was returned. For the subsequence $\{(\boldsymbol{l}_t, \boldsymbol{v}_t)\}_{0\leq t<\bar{t}}$, we replay the pairs using standard IPRO updates and treat $\boldsymbol{v}_n$ as the solution retrieved for $\boldsymbol{l}_{\bar{t}}$. For the subsequent pairs $\{(\boldsymbol{l}_t, \boldsymbol{v}_t)\}_{\bar{t}<t<n}$, we verify for each $(\boldsymbol{v}_t, \boldsymbol{l}_t)$ whether the original evaluation succeeded. If so, $\boldsymbol{v}_t$ was weakly Pareto optimal, and if a new lower bound $\boldsymbol{l}'$ exists that is strictly dominated by $\boldsymbol{v}_t$, we perform an update with $(\boldsymbol{l}', \boldsymbol{v}_t)$. If the evaluation failed, $\boldsymbol{l}_t$ was marked as complete, and we check whether a new lower bound $\boldsymbol{l}'$ dominates $\boldsymbol{l}_t$. If so, $\boldsymbol{l}'$ is also marked as complete. This mechanism corrects earlier mistakes and reuses previous iteration outcomes as efficiently as possible.

3 Theoretical Analysis of IPRO

The structured decomposition underpinning IPRO, when combined with a suitably defined Pareto oracle, provides a principled basis for theoretical analysis. This section establishes guarantees on convergence to the Pareto front, the approximation quality at each iteration, and a finite upper bound on the number of iterations required to compute a τ-Pareto front.

3.1 IPRO Foundations

We now formalise the assumptions and key objects that will support the theoretical analysis of IPRO.

Assumptions. Throughout, we assume that the Pareto oracles employed in IPRO satisfy Definitions 18 and 19 exactly. In addition, we assume that the reward function is uniformly bounded, a standard condition in both single- and multi-objective settings. This ensures that value functions remain well-defined and analytically tractable, enabling convergence and approximation guarantees.

Assumption 1: Bounded rewards

Let $\mathcal{M} = \langle S, \mathcal{A}, p, \boldsymbol{r}, p_0, \gamma \rangle$ be the MOMDP under consideration. We assume the existence of a constant $\boldsymbol{R}_{\max} \in \mathbb{R}^d$ such that for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$, $\boldsymbol{R}_{\max} \succeq_{\mathrm{p}} |\boldsymbol{r}(s,a)|$.

Definitions. Assumption 1 implies that the ideal and nadir points are both well-defined. Let $\mathcal{B} = \prod_{j=1}^{d} [v_j^{\mathrm{n}}, v_j^{\mathrm{i}}]$ denote the bounding box defined by a strict lower bound on the true nadir $\boldsymbol{v}^{\mathrm{n}}$ and the ideal point $\boldsymbol{v}^{\mathrm{i}}$. At each time step t, the current Pareto front is denoted by $\mathcal{F}_t$, while $\mathcal{C}_t$ collects all referents for which the oracle failed to produce a valid solution. Based on these, we define two essential subsets of $\mathcal{B}$: the dominated set $\mathcal{D}_t$ and the infeasible set $\mathcal{I}_t$.

Definition 20: Dominated and infeasible sets at time t

We define two subsets of the bounding box $\mathcal{B}$ at time step t as follows:

- The *dominated set* $\mathcal{D}_t$ contains all points in $\mathcal{B}$ that are dominated by or equal to a point in the current Pareto front:

$$\mathcal{D}_t = \left\{ \boldsymbol{v} \in \mathcal{B} \mid \exists \boldsymbol{v}' \in \mathcal{F}_t,\ \boldsymbol{v}' \succeq_{\mathrm{p}} \boldsymbol{v} \right\}.$$

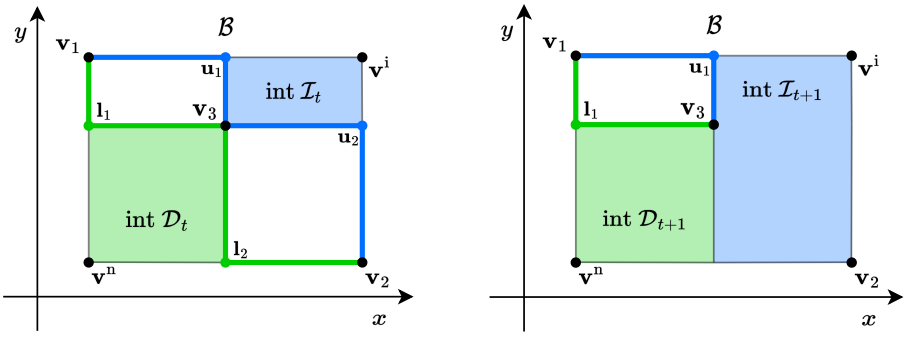
- The *infeasible set* $\mathcal{I}_t$ contains all points in $\mathcal{B}$ that dominate or are equal to a point in the union of the current Pareto front and the set of completed referents $\mathcal{C}_t$:

$$\mathcal{I}_t = \left\{ \boldsymbol{v} \in \mathcal{B} \mid \exists \boldsymbol{v}' \in \mathcal{F}_t \cup \mathcal{C}_t,\ \boldsymbol{v} \succeq_{\mathrm{p}} \boldsymbol{v}' \right\}.$$

The infeasible set includes not only points dominated by the current Pareto front but also those dominated by previously attempted referents that yielded no improvement. These sets are used to identify and eliminate regions of the search space that can no longer yield Pareto-optimal solutions.

To identify unexplored regions that may still contain undiscovered Pareto-optimal points, we define the boundary and interior of both $\mathcal{D}_t$ and $\mathcal{I}_t$ with respect to $\mathcal{B}$. Let $\overline{S}$ denote the topological closure of a subset S, and let ∂S denote its boundary. We define the boundary of $\mathcal{D}_t$ as $\partial \mathcal{D}_t = \overline{(\mathcal{B} \setminus \mathcal{D}_t)} \cap \overline{\mathcal{D}_t}$, and its interior as $\operatorname{int} \mathcal{D}_t = \mathcal{D}_t \setminus \partial \mathcal{D}_t$. The same definitions apply analogously to $\mathcal{I}_t$.

Of particular interest are the *reachable boundaries* of these sets, which characterise the portions of the boundary not yet excluded by either dominance or infeasibility. These define the remaining search region and are illustrated in Fig. 3.3 for the two-objective case.



(a) (b)

Figure 3.3: **(a)** The reachable boundaries of $\mathcal{D}_t$ (green) and $\mathcal{I}_t$ (blue) indicated with solid lines and their interiors (shaded) when no section is completed. **(b)** When completing the section at $\boldsymbol{l}_2$, parts of the reachable boundary at time step t become unreachable at time step $t+1$.

Definition 21: Reachable boundaries at time t

We define the *reachable boundaries* of $\mathcal{D}_t$ and $\mathcal{I}_t$ at time step t as follows:

- The reachable boundary of $\mathcal{D}_t$, denoted $\partial^r \mathcal{D}_t$, is

$$\partial^r \mathcal{D}_t = \partial \mathcal{D}_t \setminus \mathcal{I}_t.$$

- The reachable boundary of $\mathcal{I}_t$, denoted $\partial^r \mathcal{I}_t$, is

$$\partial^r \mathcal{I}_t = \partial \mathcal{I}_t \setminus \mathcal{D}_t.$$

We lastly define two critical subsets of the reachable boundaries, representing the extrema of the remaining search space. The set of lower bounds $\mathcal{L}$ contains the points on the reachable boundary of the dominated set such that no other point on the reachable boundary exists which is dominated by it. Similarly, the set of upper bounds $\mathcal{U}$ contains the points on the reachable boundary of the infeasible set such that there is no other point on the reachable boundary that dominates it. Conceptually, these points are the inner corners of their respective boundary as observed in Fig. 3.3.

Definition 22: Lower and upper bounds at time t

We define the following sets over the reachable boundaries at time step t:

- The *set of lower bounds* $\mathcal{L}_t$ consists of all points on the reachable boundary of the dominated region that are not strictly dominated by any other such point:

$$\mathcal{L}_t = \left\{ \boldsymbol{l} \in \partial^r \mathcal{D}_t \mid \nexists \boldsymbol{v} \in \partial^r \mathcal{D}_t,\ \boldsymbol{l} \succ_{\mathrm{p}} \boldsymbol{v} \right\}.$$

- The *set of upper bounds* $\mathcal{U}_t$ consists of all points on the reachable boundary of the infeasible region that do not strictly dominate any other such point:

$$\mathcal{U}_t = \left\{ \boldsymbol{u} \in \partial^r \mathcal{I}_t \mid \nexists \boldsymbol{v} \in \partial^r \mathcal{I}_t,\ \boldsymbol{v} \succ_{\mathrm{p}} \boldsymbol{u} \right\}.$$

3.2 Supporting Lemmas

To support the main results, we first establish a number of technical lemmas that formalise the structure and contents of the sets introduced in Section 3.1, and clarify their relation to the remaining feasible solutions. These lemmas form the basis for bounding the search space in IPRO and ensuring that progress can be made.

We begin by characterising the interior of the infeasible set. Specifically, we show that any point in the interior must either be genuinely infeasible or lie within the oracle's tolerance of a known Pareto-optimal point. This result follows from the fact that every interior point has a strictly positive distance to the boundary of $\mathcal{I}_t$, which allows us to reason about its local neighbourhood.

Lemma 1: Infeasible interior

Given an oracle Ω^τ with tolerance τ, at all time steps t and for all $\boldsymbol{v} \in \operatorname{int} \mathcal{I}_t$, $\boldsymbol{v}$ is infeasible or within the tolerance τ of a point on the current estimate of the Pareto front $\mathcal{F}_t$.

Proof. **1. Interior point and dominated referent.**
For $\boldsymbol{v} \in \operatorname{int} \mathcal{I}_t$ there exists $r > 0$ with $B_r(\boldsymbol{v})$ (the open ball of radius r around $\boldsymbol{v}$) contained in $\operatorname{int} \mathcal{I}_t$. Choose $\boldsymbol{v}' \in B_r(\boldsymbol{v})$ such that $\boldsymbol{v} > \boldsymbol{v}'$, e.g. $\boldsymbol{v}' = \boldsymbol{v} - \delta$ for some $0 < \delta < r$. By Definition 20 there is $\bar{\boldsymbol{v}} \in \mathcal{F}_t \cup \mathcal{C}_t$ with $\boldsymbol{v}' \succ_{\mathrm{p}} \bar{\boldsymbol{v}}$, and by transitivity $\boldsymbol{v} > \bar{\boldsymbol{v}}$.

2. Case analysis on $\bar{\boldsymbol{v}}$.
(A) $\bar{\boldsymbol{v}} \in \mathcal{F}_t$. If $\boldsymbol{v}$ were feasible, it would strictly dominate $\bar{\boldsymbol{v}}$ which is a point of the current Pareto front, contradicting the oracle's guarantee that every returned vector is weakly Pareto optimal. Hence $\boldsymbol{v}$ is infeasible.

(B) $\bar{\boldsymbol{v}} \in C_t$. In this case, $\bar{\boldsymbol{v}}$ was chosen as the referent in some iteration, but the oracle did not return a solution. We consider the two types of oracle separately:

(B1) Weak oracle ($\tau = 0$). Because $\boldsymbol{v} > \bar{\boldsymbol{v}}$, a weak oracle would have returned $\boldsymbol{v}$ if it were feasible. Thus $\boldsymbol{v}$ is infeasible.

(B2) Approximate oracle ($\tau > 0$).

1. $\boldsymbol{v} \succeq_p \bar{\boldsymbol{v}} + \tau$. The oracle would again have returned $\boldsymbol{v}$ if feasible, so $\boldsymbol{v}$ is infeasible.
2. $\boldsymbol{v} \not\succeq_p \bar{\boldsymbol{v}} + \tau$. By construction of the lower-bound set $\mathcal{L}_t$ there exists $\boldsymbol{v}^* \in \mathcal{F}_t$ with $\boldsymbol{v}^* + \tau \succeq_{\mathrm{p}} \boldsymbol{v}$. Hence $\|\boldsymbol{v} - \boldsymbol{v}^*\|_\infty \leq \tau$ and $\boldsymbol{v}$ lies within tolerance of $\mathcal{F}_t$.

3. Conclusion.
In every case, $\boldsymbol{v}$ is either infeasible or within τ of $\mathcal{F}_t$, completing the proof. □

Given this characterisation of the infeasible region, we next turn our attention to the remaining feasible solutions. These lie in the region outside both the dominated set and the interior of the infeasible set. The following lemma shows that all such solutions are strictly lower bounded by the set $\mathcal{L}_t$, which lies on the reachable boundary of the dominated region. The structure of the proof relies on tracing a strictly decreasing path from any feasible solution to the nadir, which necessarily intersects the reachable boundary of the dominated set, thereby guaranteeing the existence of a lower bound in $\mathcal{L}_t$.

Lemma 2: Lower bounds

At any time step t, the set of lower bounds $\mathcal{L}_t$ contains a strict lower bound for all remaining feasible solutions, i.e.,

$$\boldsymbol{v} \in \mathcal{B} \setminus (\operatorname{int} \mathcal{I}_t \cup \mathcal{D}_t) \implies \exists \boldsymbol{l} \in \mathcal{L}_t \text{ such that } \boldsymbol{v} > \boldsymbol{l}.$$

Proof. **1. From a feasible point to a dominated boundary point.**
Let $\boldsymbol{v} \in \mathcal{B} \setminus (\operatorname{int} \mathcal{I}_t \cup \mathcal{D}_t)$ be feasible. Because the nadir $\boldsymbol{v}^{\mathrm{n}}$ is a strict lower bound on every coordinate, the line segment joining $\boldsymbol{v}^{\mathrm{n}}$ to $\boldsymbol{v}$ is strictly increasing. This segment intersects $\partial\mathcal{D}_t$ at some point $\bar{\boldsymbol{v}}$ with $\boldsymbol{v} > \bar{\boldsymbol{v}}$.

2. Ensuring the boundary point is reachable.
If $\bar{\boldsymbol{v}} \notin \partial^r \mathcal{D}_t$, then by Definition 21 we have $\bar{\boldsymbol{v}} \in \mathcal{I}_t$. Strict dominance $\boldsymbol{v} > \bar{\boldsymbol{v}}$ would place $\boldsymbol{v}$ inside int $\mathcal{I}_t$, contradicting the premise. Hence $\bar{\boldsymbol{v}} \in \partial^r \mathcal{D}_t$.

3. Extracting a lower-bound vector.
Two possibilities arise:

1. $\bar{\boldsymbol{v}} \in \mathcal{L}_t$. Then $\boldsymbol{v} > \bar{\boldsymbol{v}}$ already exhibits the desired $\boldsymbol{l} = \bar{\boldsymbol{v}}$.
2. $\bar{\boldsymbol{v}} \notin \mathcal{L}_t$. By Definition 22 there exists $\boldsymbol{l} \in \mathcal{L}_t$ with $\bar{\boldsymbol{v}} \succ_{\mathrm{p}} \boldsymbol{l}$. Transitivity of strict dominance gives $\boldsymbol{v} > \boldsymbol{l}$.

4. Conclusion.
In all cases we have produced $\boldsymbol{l} \in \mathcal{L}_t$ such that $\boldsymbol{v} > \boldsymbol{l}$, completing the proof. □

We now provide the dual statement for the upper set. Here, we show that all remaining feasible solutions are upper bounded by a point in $\mathcal{U}_t$. As in the previous case, this set lies on the reachable boundary, now of the infeasible region, and contains all points that are not dominated by any other such point.

Lemma 3: Upper bounds

During IPRO's execution, the upper set contains an upper bound for all remaining feasible solutions, i.e.,

$$\boldsymbol{v} \in \mathcal{B} \setminus (\operatorname{int} \mathcal{I}_t \cup \mathcal{D}_t) \implies \exists \boldsymbol{u} \in \mathcal{U}_t \text{ such that } \boldsymbol{u} \succeq_{\mathrm{p}} \boldsymbol{v}.$$

Proof. Proceed exactly as in Lemma 2, with a single substitution:

Segment direction. Start from the ideal point $\boldsymbol{v}^{\mathrm{i}}$ (rather than the nadir $\boldsymbol{v}^{\mathrm{n}}$) and follow the coordinate-non-increasing segment to $\boldsymbol{v}$. Its first intersection with $\partial \mathcal{I}_t$ yields a point $\bar{\boldsymbol{v}} \succeq_{\mathrm{p}} \boldsymbol{v}$ that lies on $\partial^r \mathcal{I}_t$.

All subsequent steps mirror those in Lemma 2, with $\succeq_{\mathrm{p}}$ in place of $>$ and the set $\mathcal{U}_t$ in place of $\mathcal{L}_t$, giving the required $\boldsymbol{u} \in \mathcal{U}_t$ such that $\boldsymbol{u} \succeq_{\mathrm{p}} \boldsymbol{v}$. □

3.3 Upper Bounding the Error

At each time step, IPRO maintains a finite set $\mathcal{U}_t$ that bounds the remaining feasible solutions from above. This structure enables an explicit upper bound on the current approximation error, which is the distance between the current Pareto front estimate $\mathcal{F}_t$ and the true front $\mathcal{F}^*$.

We define the true approximation error at time step t as the worst-case distance between any true Pareto-optimal point and its nearest approximation:

$$\varepsilon_t^* = \sup_{\boldsymbol{v}^* \in \mathcal{F}^*} \min_{\boldsymbol{v} \in \mathcal{F}_t} \|\boldsymbol{v}^* - \boldsymbol{v}\|_\infty. \tag{3.1}$$

Since Lemma 3 guarantees that every undiscovered Pareto-optimal point is dominated by or equal to some $\boldsymbol{u} \in \mathcal{U}_t$, and $\mathcal{U}_t$ is finite for all $t \in \mathbb{N}$, we can upper bound ε_t^* by evaluating only the points in $\mathcal{U}_t$. This yields the following result:

Theorem 1: Approximation guarantee

Let $\mathcal{F}^*$ denote the true Pareto front and let $\mathcal{F}_t$ be the approximation produced by IPRO at time step t. For every $t \in \mathbb{N}$ the true approximation error ε_t^* satisfies

$$\varepsilon_t^* \leq \max_{\boldsymbol{u} \in \mathcal{U}_t} \min_{\boldsymbol{v} \in \mathcal{F}_t} \|\boldsymbol{u} - \boldsymbol{v}\|_\infty.$$

Proof. **1. Each undiscovered Pareto point is dominated by an upper bound.**
By Lemmas 1 and 3, every Pareto-optimal vector that is not yet in $\mathcal{F}_t$ is either within its tolerance or still feasible and therefore dominated by some $\boldsymbol{u} \in \mathcal{U}_t$:

$$\forall \boldsymbol{v}^* \in \mathcal{F}^* \setminus (\operatorname{int} \mathcal{I}_t \cup \mathcal{F}_t)\ \exists \boldsymbol{u} \in \mathcal{U}_t \ :\ \ \boldsymbol{u} \succeq_{\mathrm{p}} \boldsymbol{v}^*. \tag{1}$$

2. Bounding the worst-case distance.
Given (1), we can replace the supremum over $\mathcal{F}^*$ in Eq. (3.1) with a maximum over the finite set $\mathcal{U}_t$, which yields

$$\varepsilon_t^* \ = \ \sup_{\boldsymbol{v}^* \in \mathcal{F}^*} \min_{\boldsymbol{v} \in \mathcal{F}_t} \|\boldsymbol{v}^* - \boldsymbol{v}\|_\infty \ \leq \ \max_{\boldsymbol{u} \in \mathcal{U}_t} \min_{\boldsymbol{v} \in \mathcal{F}_t} \|\boldsymbol{u} - \boldsymbol{v}\|_\infty. \qquad \square$$

Fig. 3.1b provides an example where $\mathcal{U}_t = \{\boldsymbol{u}_1, \boldsymbol{u}_3, \boldsymbol{u}_4\}$ captures the worst-case gap to the true front. We define $\varepsilon_t := \max_{\boldsymbol{u} \in \mathcal{U}_t} \min_{\boldsymbol{v} \in \mathcal{F}_t} \|\boldsymbol{u} - \boldsymbol{v}\|_\infty$ and refer to ε_t as the *error bound*. While our results are stated using the L^∞ norm, the same reasoning extends to other metrics.

A useful corollary of Theorem 1 is that the sequence of error bounds is monotonically decreasing. This follows immediately since the upper bounds are only adjusted downwards and our approximation of the Pareto front only improves.

Corollary 1: Monotonicity of the error sequence

The sequence of error bounds $(\varepsilon_t)_{t \in \mathbb{N}}$ is monotonically decreasing.

Proof. **1. The approximation set expands.**
IPRO only appends new Pareto-optimal vectors, so $\mathcal{F}_t \subseteq \mathcal{F}_{t+1}$ for every t.

2. Upper bounds tighten.
By construction, every remaining feasible vector at time $t+1$ is dominated by or equal to some $\boldsymbol{u} \in \mathcal{U}_{t+1}$. But the same already held at time t for some $\bar{\boldsymbol{u}} \in \mathcal{U}_t$; hence for each $\boldsymbol{u} \in \mathcal{U}_{t+1}$ there exists $\bar{\boldsymbol{u}} \in \mathcal{U}_t$ with $\bar{\boldsymbol{u}} \succeq_{\mathrm{p}} \boldsymbol{u}$.

3. Putting the pieces together.
Because $\mathcal{F}_{t+1}$ is a superset of $\mathcal{F}_t$, the inner minimum can only decrease. Because every $\boldsymbol{u} \in \mathcal{U}_{t+1}$ is dominated by or equal to some $\bar{\boldsymbol{u}} \in \mathcal{U}_t$, the outer maximum can only decrease as well. Consequently,

$$\max_{\boldsymbol{u} \in \mathcal{U}_{t+1}} \min_{\boldsymbol{v} \in \mathcal{F}_{t+1}} \|\boldsymbol{u} - \boldsymbol{v}\|_\infty \ \leq \ \max_{\boldsymbol{u} \in \mathcal{U}_t} \min_{\boldsymbol{v} \in \mathcal{F}_t} \|\boldsymbol{u} - \boldsymbol{v}\|_\infty,$$

which proves $\varepsilon_{t+1} \leq \varepsilon_t$ for all $t \in \mathbb{N}$. $\square$

3.4 Convergence to an Approximate Pareto Front

As IPRO progresses, the error bound ε_t decreases monotonically and converges to zero. This behaviour is visualised in Fig. 3.1b, where the addition of a new Pareto-optimal solution reduces the distance to the upper bounds. Similarly, when a section is completed, as shown in Fig. 3.1c, the corresponding upper bound is removed, further shrinking the unexplored region.

IPRO halts when the true approximation error ε_t^* is guaranteed to be at most equal to the oracle's tolerance τ. At that point, no region of the objective space remains that could contain a Pareto-optimal solution more than τ away from the current approximation. The result is a valid τ-Pareto front. The following theorem formalises this guarantee.

Theorem 2: Approximate Pareto oracle convergence

Given an approximate Pareto oracle Ω^τ with tolerance $\tau > 0$, IPRO converges to a τ-Pareto front in a finite number of iterations.

Proof. **1. τ-partition of the search space.**
Partition the bounding box $\mathcal{B}$ into a finite grid of axis-aligned hypercubes ("cells") of side τ. A cell is *active* at time step t if it intersects neither the dominated set $\mathcal{D}_t$ nor the infeasible set int $\mathcal{I}_t$.

2. Eliminating active cells.
Let $\boldsymbol{l} \in \mathcal{L}_t$ be the current lower bound, and query the oracle at the shifted point $\boldsymbol{l} + \tau$, located in an active cell C. Exactly one of the following occurs:

- *Oracle succeeds.* The new Pareto-optimal vector is added to $\mathcal{F}_{t+1}$, and every cell now intersecting with the dominated or infeasible set, including C, is deactivated.
- *Oracle fails.* Because $\boldsymbol{l} + \tau$ is infeasible, every cell whose points dominate $\boldsymbol{l} + \tau$ is infeasible and is likewise removed from the active set, deactivating C.

Hence at least one active cell is removed at each iteration. If the referent $\boldsymbol{l} + \tau$ happens to lie in a cell that is already infeasible, then $\boldsymbol{l}$ itself is discarded, and the algorithm selects a new lower bound. Repeating this procedure eventually exhausts $\mathcal{L}_t$, at which point no active cells remain and IPRO terminates.

3. Finite termination.
Because the grid is finite, repeated deactivation exhausts the active set after finitely many iterations. As such, IPRO will eventually reach a time step T where no active cells remain.

4. τ-Pareto optimality.
By Theorem 1, the true approximation error is bounded above by ε_t. Since IPRO only

stops when $\varepsilon_T \leq \tau$, the set $\mathcal{F}_T$ forms a valid τ-Pareto front and IPRO terminates at or before time step T. □

Having established that IPRO converges to a valid τ-Pareto front in finite time, we now turn to bounding the number of iterations required. While the algorithm is guaranteed to terminate, the number of steps may still be large in practice. In particular, the complexity of IPRO depends on both the oracle tolerance τ and the number of objectives d. As with related approaches, we find that the iteration count grows polynomially with $1/\tau$ but exponentially with d [Papadimitriou and Yannakakis, 2000; Chatterjee et al., 2006].

Theorem 3: Upper bounding the number of iterations

Given a Pareto oracle Ω^τ and tolerance $\tau > 0$, let $\forall j \in [d], k_j = \lceil (v_j^{\mathrm{i}} - v_j^{\mathrm{n}})/\tau \rceil$. IPRO constructs a τ-Pareto front in at most

$$\prod_{j=1}^{d} k_j - \prod_{j=1}^{d} (k_j - 1)$$

iterations, which is a polynomial in $1/\tau$ but exponential in the number of objectives d.

Proof. We consider an approximate Pareto oracle with tolerance $\tau > 0$. Let $\mathcal{B}$ be the bounding box defined by the nadir $\boldsymbol{v}^{\mathrm{n}}$ and the ideal $\boldsymbol{v}^{\mathrm{i}}$. Partition $\mathcal{B}$ into axis-aligned hypercubes of side length τ, forming a grid of cells. The total number of cells in $\mathcal{B}$ is then equal to $\prod_{j=1}^{d} k_j$, where $k_j = \left\lceil (\boldsymbol{v}_j^{\mathrm{i}} - \boldsymbol{v}_j^{\mathrm{n}})/\tau \right\rceil$.
The proof consists of three steps:

1. We show that it is always possible to construct a worst-case run of IPRO in which every oracle call succeeds;
2. We use this to argue that the worst case occurs when Pareto-optimal vectors lie along the Pareto-optimal facets of $\mathcal{B}$;
3. We compute the number of such cells, yielding the desired iteration bound.

1. Reduction to successful oracle calls.
An oracle failure removes the current lower bound $\boldsymbol{l}_t$ from $\mathcal{L}_t$ but does not introduce any new lower bounds. A successful call does the same and may add additional lower bounds corresponding to newly discovered Pareto-optimal points. Hence replacing a failure by a success can only increase (never decrease) the number of future iterations. Therefore the longest possible run of IPRO is obtained when every oracle call succeeds. We may restrict attention to that case when deriving an upper bound on the total number of iterations.

2. Dominance-free sets.
Let $\mathcal{S}$ be a set of grid cells, each containing a Pareto-optimal point. In the worst case, $\mathcal{S}$ has maximal cardinality subject to the condition that no cell in $\mathcal{S}$ strictly dominates another. We now construct from $\mathcal{S}$ a new set $\mathcal{S}'$ that lies entirely on the Pareto-optimal facets of $\mathcal{B}$.
For each $\boldsymbol{c} \in \mathcal{S}$, define

$$\boldsymbol{c}' = \begin{cases} \boldsymbol{c} + 1, & \text{if } \boldsymbol{c} + 1 \in \mathcal{B}, \\ \boldsymbol{c}, & \text{otherwise.} \end{cases}$$

Let $\mathcal{S}' = \{\boldsymbol{c}' \mid \boldsymbol{c} \in \mathcal{S}\}$. We show that: (i) $\mathcal{S}'$ contains no strictly dominating pairs, and (ii) $|\mathcal{S}'| = |\mathcal{S}|$.
If $\boldsymbol{c}'_1 > \boldsymbol{c}'_2$ for some $\boldsymbol{c}'_1, \boldsymbol{c}'_2 \in \mathcal{S}'$, then one of the following cases must hold:

1. $\boldsymbol{c}_1 = \boldsymbol{c}'_1$ and $\boldsymbol{c}_2 = \boldsymbol{c}'_2$;
2. $\boldsymbol{c}_1 + 1 = \boldsymbol{c}'_1$ and $\boldsymbol{c}_2 = \boldsymbol{c}'_2$;
3. $\boldsymbol{c}_1 = \boldsymbol{c}'_1$ and $\boldsymbol{c}_2 + 1 = \boldsymbol{c}'_2$;
4. $\boldsymbol{c}_1 + 1 = \boldsymbol{c}'_1$ and $\boldsymbol{c}_2 + 1 = \boldsymbol{c}'_2$.

But (1), (3), and (4) would imply $\boldsymbol{c}_1 > \boldsymbol{c}_2$, contradicting that $\mathcal{S}$ contains no strictly dominating pairs. Case (2) is also impossible, as $\boldsymbol{c}'_2$ lies on the boundary of $\mathcal{B}$ and the only points that could strictly dominate it are outside $\mathcal{B}$.
Next, we demonstrate that $|\mathcal{S}| = |\mathcal{S}'|$. Since $\mathcal{S}$ was constructed as a set of maximal size that contains no cell strictly dominating another, and since $\mathcal{S}'$ does not contain such dominance, we have $|\mathcal{S}| \geq |\mathcal{S}'|$. Suppose $|\mathcal{S}| > |\mathcal{S}'|$. Then, at least two cells, $\boldsymbol{c}_1, \boldsymbol{c}_2 \in \mathcal{S}$, must map to the same $\boldsymbol{c}' \in \mathcal{S}'$. Since $\boldsymbol{c}_1 \neq \boldsymbol{c}_2$, we have $\boldsymbol{c}_1 + 1 \neq \boldsymbol{c}_2 + 1$, implying $\boldsymbol{c}_1 = \boldsymbol{c}_2 + 1$ or $\boldsymbol{c}_2 = \boldsymbol{c}_1 + 1$. However, this leads to a contradiction as $\boldsymbol{c}_1 > \boldsymbol{c}_2$ or $\boldsymbol{c}_2 > \boldsymbol{c}_1$.

3. Counting non-dominated cells.
Repeatedly applying this operation yields a fixed point $\mathcal{S}^*$, where all $\boldsymbol{c}^* \in \mathcal{S}^*$ lie on the Pareto-optimal facets of $\mathcal{B}$. The number of iterations required is then equal to the number of cells on these facets, which is

$$\prod_{j=1}^{d} k_j - \prod_{j=1}^{d} (k_j - 1).$$

This yields the claimed iteration bound. □

3.5 Convergence to the True Pareto Front

Finally, we consider the case of a weak Pareto oracle with zero tolerance. Unlike approximate oracles, which are assumed to always return Pareto-optimal solutions, weak oracles may return any weakly optimal point. To guarantee convergence under this weaker condition, we must exclude pathological behaviours such as repeatedly selecting suboptimal referents or stalling near the Pareto front without ever reaching it. While this setting is unlikely to arise in practice, it provides valuable theoretical insight into the limits of IPRO's applicability.

Assumptions

Before introducing the assumptions, we require two notions that describe which parts of the search space are still critical at a given scale η.

Definition 23: η-critical sets

For a threshold $\eta > 0$ define the η-critical upper and lower sets

$$\mathcal{U}_t^\eta := \left\{ \boldsymbol{u} \in \mathcal{U}_t : \min_{\boldsymbol{v} \in \mathcal{F}_t} \|\boldsymbol{u} - \boldsymbol{v}\|_\infty \geq \eta \right\}, \quad \mathcal{L}_t^\eta := \left\{ \boldsymbol{l} \in \mathcal{L}_t : \exists\, \boldsymbol{u} \in \mathcal{U}_t^\eta \text{ with } \boldsymbol{u} > \boldsymbol{l} \right\}.$$

Intuitively, $\mathcal{U}_t^\eta$ identifies the regions of the search space where the algorithm has not yet achieved η-level accuracy, while $\mathcal{L}_t^\eta$ tracks the referents that "guard" those regions. Together, they mark the parts of the search space that still matter at resolution η.

Definition 24: Error potential at scale η

For $\eta > 0$, define the multiplicity of η-critical uppers $m_t(\eta) := |\mathcal{U}_t^\eta|$ and the error potential

$$\mathcal{P}_t(\eta) := \varepsilon_t^* + \eta\, m_t(\eta).$$

The error potential extends the true error ε_t^* by penalising the presence of many unresolved η-critical uppers. In this way, $\mathcal{P}_t(\eta)$ measures not only how close the current approximation is, but also how much unexplored territory remains at scale η. We can now state the two assumptions required for convergence with a weak Pareto oracle. The first one, called η-fair selection, ensures that the selection rule does not indefinitely neglect the search space that still matters at scale η.

Assumption 2: η-eventual selection

Let $(\eta_k)_{k \geq 0}$ be decreasing with $\eta_k \downarrow 0$. For each k and every t, if $\mathcal{L}_t^{\eta_k} \neq \varnothing$ then there exists a (random) time $s \geq t$ such that $\textsc{Select}(\mathcal{L}_s) \in \mathcal{L}_s^{\eta_k}$.

Assumption 2 only requires *eventual* service of any active η-critical set. This is mild and, in particular, is already satisfied by a certificate-greedy rule that selects a lower under an upper with maximal certificate gap. It is also satisfied by stochastic selection with full support or by heuristics that periodically force a pick from $\mathcal{L}_t^{\eta_k}$. Greedy rules based on other scores (e.g. area/hypervolume gain) may violate the assumption unless paired with such a gate. The next assumption ensures that whenever the algorithm does look into a region that still matters at scale η, there is a strictly positive chance of making tangible progress: the error potential will fall by at least $\Delta(\eta)$.

Assumption 3: Progress at scale η

There exist functions $p : (0, \varepsilon_{\max}] \to (0, 1]$ and $\Delta : (0, \varepsilon_{\max}] \to (0, \infty)$, with $\varepsilon_{\max} := \|\boldsymbol{v}^{\mathrm{i}} - \boldsymbol{v}^{\mathrm{n}}\|_\infty$ such that the following holds. Fix $\eta > 0$ and a time t with $\varepsilon_t^* \geq \eta$. If a referent $\boldsymbol{l} \in \mathcal{L}_t^\eta$ is queried, then with probability at least $p(\eta)$,

$$\mathcal{P}_{t+1}(\eta) \;\leq\; \mathcal{P}_t(\eta) \;-\; \Delta(\eta).$$

(Lower envelopes.) For every fixed $\eta > 0$, the quantities $p_\downarrow(\eta) := \inf_{x \in [\eta, \varepsilon_{\max}]} p(x) > 0$ and $\Delta_\downarrow(\eta) := \inf_{x \in [\eta, \varepsilon_{\max}]} \Delta(x) > 0$ are strictly positive.

Importantly, we do not assume that every query leads to improvement, nor that improvements occur at a fixed rate. Progress may be sporadic, and temporary stalling is allowed. The only requirement is that genuine progress never becomes impossible. Without such a safeguard, one can construct pathological scenarios in which the search becomes trapped, making increasingly smaller improvements and converging to a front that is arbitrarily far from optimal.

Motivating example. Figure 3.4 illustrates why both Assumptions 2 and 3 are necessary. Consider a weak Pareto oracle with zero tolerance, and suppose the feasible set consists of $\boldsymbol{v}_1$ together with the continuous line segment connecting $\boldsymbol{l}_1$ and $\boldsymbol{v}_2$. The Pareto-optimal points are $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$, yet a subset of the line is weakly optimal since $\boldsymbol{v}_2$ dominates it only along the y-axis.

If Assumption 2 is omitted, a depth-first selection rule that first explores $\boldsymbol{l}_2$ and its descendants may never return to $\boldsymbol{l}_1$, and thus never discover $\boldsymbol{v}_1$. Although the search may still approach $\boldsymbol{v}_2$ asymptotically, it would do so without ever resolving the other region.

Conversely, without Assumption 3, the algorithm might keep "walking the line" toward $\boldsymbol{v}_2$ through infinitesimal improvements, converging to a suboptimal limit $\bar{\boldsymbol{v}}$ under a weak oracle (effectively exhibiting Zeno-like behaviour). Assumption 3 prevents this behaviour by ensuring that, as long as a positive error remains, there is always a non-zero chance of making a meaningful improvement. Intuitively, the η-critical sets act as the active cells in the proof of Theorem 2, guaranteeing that the algorithm cannot remain indefinitely stuck in any region that still contributes to the error at scale η.

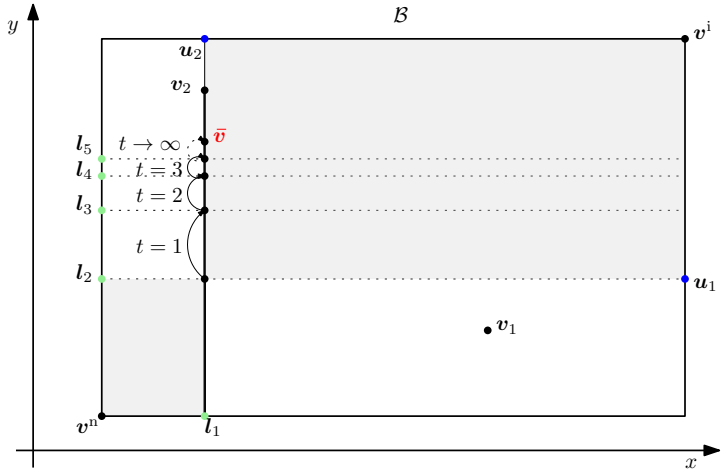


Figure 3.4: An illustrative front demonstrating the necessity of Assumptions 2 and 3. $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ are the only two remaining Pareto-optimal solutions in the bounding box. The line from $\boldsymbol{l}_1$ to $\boldsymbol{v}_2$ contains all other feasible solutions. The green points mark the successive lower bounds generated by IPRO.

Proof

With the assumptions in place, we are ready to establish the main result. The key idea is straightforward: IPRO's sequential treatment of lower and upper bounds combined with the conditions above guarantee that the true error vanishes over time, and hence that IPRO converges to the true Pareto front.

Theorem 4: Weak Pareto oracle convergence

Under Assumptions 2 and 3, $\varepsilon_t^* \to 0$ almost surely as $t \to \infty$.

Proof. Fix any $\eta > 0$. We prove that $\mathbb{P}(\limsup_{t\to\infty} \varepsilon_t^* \geq \eta) = 0$. The argument proceeds as follows.

1. (*Non-emptiness at scale η*) Whenever $\varepsilon_t^* \geq \eta$, there are still unresolved regions at scale η, i.e. $\mathcal{U}_t^\eta \neq \varnothing$ and thus $\mathcal{L}_t^\eta \neq \varnothing$.

2. (*η-critical queries occur infinitely often*) On the event $\{\varepsilon_t^* \geq \eta \text{ i.o.}\}$, the algorithm selects an element of $\mathcal{L}_t^\eta$ infinitely many times.

3. (*Progress has a uniform floor*) By Assumption 3, each η-critical query has probability at least $p_\downarrow(\eta) > 0$ (uniform over time and past) of decreasing the

potential

$$\mathcal{P}_t(\eta) \;:=\; \varepsilon_t^* + \eta\, m_t(\eta), \quad m_t(\eta) = |\mathcal{U}_t^\eta|$$

by at least $\Delta_\downarrow(\eta) > 0$.

4. (*Infinitely many drops with positive size*) Using the preceding and a standard multiplicative bound with conditional probabilities, the events "$\mathcal{P}$ drops by $\geq \Delta_\downarrow(\eta)$ at an η-critical query" occur infinitely often with probability one.

5. (*Contradiction*) Because $\mathcal{P}_t(\eta) \geq 0$ always, it cannot decrease by a positive amount infinitely often. Hence $\mathbb{P}(\limsup_t \varepsilon_t^* \geq \eta) = 0$. Letting η range over the positive rationals yields $\varepsilon_t^* \to 0$ almost surely.

1. Non-emptiness at scale η.
By Theorem 1, $\varepsilon_t^* \geq \eta$ implies $\varepsilon_t \geq \eta$, so there exists $\boldsymbol{u} \in \mathcal{U}_t$ with $\min_{\boldsymbol{v}\in\mathcal{F}_t} \|\boldsymbol{u} - \boldsymbol{v}\|_\infty \geq \eta$, i.e. $\boldsymbol{u} \in \mathcal{U}_t^\eta \neq \varnothing$. By Lemmas 2 and 3, any still-feasible $\boldsymbol{v}^*$ is squeezed between some $\boldsymbol{l} \in \mathcal{L}_t$ and some $\boldsymbol{u} \in \mathcal{U}_t$, hence for every $\boldsymbol{u} \in \mathcal{U}_t^\eta$ there exists $\boldsymbol{l} \in \mathcal{L}_t$ with $\boldsymbol{u} > \boldsymbol{l}$. Therefore $\mathcal{L}_t^\eta \neq \varnothing$.

2. η-critical queries occur infinitely often.
Define the (random, increasing) sequence of *η-critical query times*

$$\begin{aligned}
\iota_1 &:= \inf\{t \geq 0 \;:\; \mathcal{L}_t^\eta \neq \varnothing,\ \textsc{Select}(\mathcal{L}_t) \in \mathcal{L}_t^\eta\},\\
\iota_{k+1} &:= \inf\{t > \iota_k \;:\; \mathcal{L}_t^\eta \neq \varnothing,\ \textsc{Select}(\mathcal{L}_t) \in \mathcal{L}_t^\eta\} \quad (k \geq 1).
\end{aligned}$$

We claim that on $\{\varepsilon_t^* \geq \eta \text{ i.o.}\}$ the sequence $(\iota_k)_{k\geq 1}$ is infinite. Assume, for contradiction, that only finitely many η-critical queries occur on $\{\varepsilon_t^* \geq \eta \text{ i.o.}\}$. Then there exists a (random) time T such that for all $s \geq T$ we have $\textsc{Select}(\mathcal{L}_s) \notin \mathcal{L}_s^\eta$. By Step 1, on the event $\{\varepsilon_t^* \geq \eta \text{ i.o.}\}$ there are infinitely many $t \geq T$ with $\mathcal{L}_t^\eta \neq \varnothing$. By Assumption 2, for each such t there exists $s \geq t$ with $\textsc{Select}(\mathcal{L}_s) \in \mathcal{L}_s^\eta$, which contradicts the definition of T. Hence (ι_k) must be infinite on $\{\varepsilon_t^* \geq \eta \text{ i.o.}\}$.

3. Uniform floor on progress.
By Assumption 3, there exist $p_\downarrow(\eta) \in (0, 1]$ and $\Delta_\downarrow(\eta) > 0$ such that whenever an η-critical referent is queried at time t with $\varepsilon_t^* \geq \eta$,

$$\mathbb{P}\big(\mathcal{P}_{t+1}(\eta) \;\leq\; \mathcal{P}_t(\eta) - \Delta_\downarrow(\eta)\big) \;\geq\; p_\downarrow(\eta),$$

with the bound holding uniformly (no dependence on t or the past).

4. Infinitely many drops with positive size.
Let $\iota_1 < \iota_2 < \cdots$ be the (infinite) η-critical query times from Step 2, and define

$$G_k \;:=\; \big\{\mathcal{P}_{\iota_k+1}(\eta) \leq \mathcal{P}_{\iota_k}(\eta) - \Delta_\downarrow(\eta)\big\}, \qquad F_k \;:=\; \neg G_k.$$

By iterating the chain rule and using the uniform lower bound from Step 3, for any $N \geq n$,

$$\mathbb{P}\left(\bigcap_{k=n}^{N} F_k\right) = \prod_{k=n}^{N} \mathbb{P}\left(F_k \,\middle|\, \bigcap_{j=n}^{k-1} F_j\right) \leq (1 - p_{\downarrow}(\eta))^{N-n+1} \xrightarrow[N\to\infty]{} 0.$$

Hence $\mathbb{P}(G_k \text{ i.o.}) = 1$.

5. Contradiction and conclusion.
Each occurrence of G_k reduces $\mathcal{P}(\eta)$ by at least $\Delta_{\downarrow}(\eta)$, so along (ι_k),

$$\mathcal{P}_{\iota_n}(\eta) \;\leq\; \mathcal{P}_{\iota_1}(\eta) \;-\; n\,\Delta_{\downarrow}(\eta) \xrightarrow[n\to\infty]{} -\infty,$$

contradicting $\mathcal{P}_t(\eta) \geq 0$ for all t. Therefore $\mathbb{P}(\limsup_{t\to\infty} \varepsilon_t^* \geq \eta) = 0$.
To extend this to all $\eta > 0$, note that

$$\{\limsup_{t\to\infty} \varepsilon_t^* > 0\} \;=\; \bigcup_{q\in\mathbb{Q}_{>0}} \{\limsup_{t\to\infty} \varepsilon_t^* \geq q\}.$$

The equality holds because if the limit superior is positive, it must exceed some rational $q > 0$, and conversely any such event implies $\limsup \varepsilon_t^* > 0$. Since $\mathbb{Q}_{>0}$ is countable and each event on the right has probability zero, their union also has probability zero. Hence

$$\mathbb{P}\left(\limsup_{t\to\infty} \varepsilon_t^* > 0\right) = 0,$$

which implies $\varepsilon_t^* \to 0$ almost surely. □

Weakening Assumption 3. Assumption 3 is imposed on the oracle to ensure a uniform probability of progress at scale η across all settings. One way to relax this requirement is to work with the natural filtration of the algorithm (the increasing sequence of σ-algebras representing the information revealed up to time t; see Klenke, 2020). Rather than imposing an unconditional lower bound $p(\eta)$, one could require that the conditional probability of progress at scale η, given the past, is almost surely bounded away from zero. This allows the progress probability to depend on the realised history while still precluding degenerate behaviour, and may be amenable to analysis via tools such as the conditional Borel-Cantelli lemma [Klenke, 2020].

An alternative route is to explore structural assumptions on the geometry of the Pareto front or on the feasible set itself. For example, one could imagine regularity conditions ensuring that progress is unavoidable once a certain region has been reached. Such assumptions might offer a different perspective on convergence guarantees without relying directly on probabilistic lower bounds.

Note on the error bound. Although Theorem 4 ensures that $\varepsilon_t^* \to 0$, the upper bound

$$\varepsilon_t \;:=\; \max_{\boldsymbol{u}\in\mathcal{U}_t}\; \min_{\boldsymbol{v}\in F_t} \|\boldsymbol{u}-\boldsymbol{v}\|_\infty$$

need not decrease under a weak oracle and may remain large even when the true error is small. Consider a bi-objective front with extrema $(x,0)$ and $(0,x)$, nadir $(0,0)$ and ideal (x,x), and suppose the only undiscovered Pareto point is the centre $\boldsymbol{v}^* = (\frac{x}{2},\frac{x}{2})$. Assume further that all other feasible (but non-optimal) solutions lie on the weakly optimal facet $\{(\frac{x}{2},y) \;:\; y \in (0,\frac{x}{2}]\}$. After the first iteration, the upper bound $\boldsymbol{u} = (\frac{x}{2},x)$ enters $\mathcal{U}_1$. Under Assumption 3, IPRO guarantees that $\varepsilon_t^* \to 0$ and so approaches $\boldsymbol{v}^*$. However, $\boldsymbol{v}^*$ need not be reached at any finite time, so $\boldsymbol{u}$ can persist in $\mathcal{U}_t$ for all t, yielding $\varepsilon_t = \frac{x}{2}$ at every step. Thus the bound can substantially overstate the residual error under weak oracles. A similar pattern can also be observed in Fig. 3.4, where if we already obtained $\boldsymbol{v}_1$, it is possible that the upper bound $\boldsymbol{u}_2$ remains in $\mathcal{U}_t$ as $t \to \infty$, thereby never decreasing the upper bound while actually decreasing the true error to zero.

A strictly stronger, and practically more informative, variant of Assumption 3 would require progress in the bound itself, e.g. with positive probability either $\varepsilon_{t+1} \le \varepsilon_t - \Delta_\downarrow(\eta)$ whenever $\varepsilon_t \ge \eta$, or the multiplicity of η-critical uppers decreases. Such a condition implies the true-error guarantee while also precluding the pathology above, at the cost of placing additional demands on how updates prune $\mathcal{U}_t$.

4 What Makes a Reliable Pareto Oracle

The reliance of IPRO on the Pareto oracle is convenient from a theoretical perspective, but places strict requirements on the oracle. In this section, we take a closer look at these oracles and propose several implementations that have both the necessary theoretical justification as well as practical appeal.

4.1 Achievement Scalarising Functions as Oracles

The structure of a Pareto oracle that receives a referent and returns a strictly improving Pareto-optimal solution aligns naturally with implementations based on *achievement scalarising functions* (ASFs) [Miettinen, 1998]. These functions scalarise a multi-objective problem such that an optimal solution to the single-objective problem is (weakly) Pareto optimal. We now introduce the relevant definitions and show how ASFs can be used to construct Pareto oracles.

Formalisation. Let $\mathcal{X}$ denote the set of feasible solutions, and let $f : \mathcal{X} \to \mathbb{R}^d$ map each solution to its d-dimensional return. Define the Euclidean distance between a point $\boldsymbol{v} \in \mathbb{R}^d$ and a set $\mathcal{Y} \subseteq \mathbb{R}^d$ as $\mathrm{dist}(\boldsymbol{v},\mathcal{Y}) = \inf_{\boldsymbol{y}\in\mathcal{Y}} \|\boldsymbol{v}-\boldsymbol{y}\|$. We also define the set $\mathbb{R}^d_\delta =$

$\{\boldsymbol{v} \in \mathbb{R}^d \mid \operatorname{dist}(\boldsymbol{v}, \mathbb{R}^d_{\geq 0}) \leq \delta\|\boldsymbol{v}\|\}$, where $\delta \in [0, 1)$ is a fixed scalar. We now formally define *order representing* and *order approximating* ASFs, which form the basis of our oracle construction.

Definition 25: Achievement scalarising functions

Fix any reference point $\boldsymbol{r} \in \mathbb{R}^d$, and let $x, y \in \mathcal{X}$ with $f(x) = \boldsymbol{x}$ and $f(y) = \boldsymbol{y}$. Let $s_{\boldsymbol{r}} : \mathbb{R}^d \to \mathbb{R}$ be an achievement scalarisation function such that $s_{\boldsymbol{r}}(\boldsymbol{r}) = 0$. We distinguish two classes:

- *Order representing*: $s_{\boldsymbol{r}}$ is strictly increasing and only assigns non-negative values in the target region,

$$\boldsymbol{x} > \boldsymbol{y} \implies s_{\boldsymbol{r}}(\boldsymbol{x}) > s_{\boldsymbol{r}}(\boldsymbol{y}) \quad \text{and} \quad \{\boldsymbol{v} \in \mathbb{R}^d \mid s_{\boldsymbol{r}}(\boldsymbol{v}) \geq 0\} = \boldsymbol{r} + \mathbb{R}^d_{\geq 0}.$$

- *Order approximating*: $s_{\boldsymbol{r}}$ is strongly increasing but may assign non-negative values outside the target region,

$$\boldsymbol{x} \succ_{\mathrm{p}} \boldsymbol{y} \implies s_{\boldsymbol{r}}(\boldsymbol{x}) > s_{\boldsymbol{r}}(\boldsymbol{y}) \quad \text{and} \quad \boldsymbol{r} + \mathbb{R}^d_{\bar{\delta}} \subset \{\boldsymbol{v} \in \mathbb{R}^d \mid s_{\boldsymbol{r}}(\boldsymbol{v}) \geq 0\} \subset \boldsymbol{r} + \mathbb{R}^d_{\delta},$$

  where $\delta > \bar{\delta} > 0$.

Example 18: Achievement scalarising functions

Consider the referent $\boldsymbol{r} = (0, 0)$ and three vectors:

$$\boldsymbol{v}_1 = (1, 2), \quad \boldsymbol{v}_2 = (1, 1), \quad \boldsymbol{v}_3 = (-0.1, 10),$$

where $\boldsymbol{v}_1$ Pareto dominates $\boldsymbol{v}_2$ (i.e., $\boldsymbol{v}_1 \succ_{\mathrm{p}} \boldsymbol{v}_2$), but not strictly in all dimensions and $\boldsymbol{v}_3$ is not dominated by either.

An *order representing* ASF $s_{\boldsymbol{r}}$ is only required to distinguish strictly better vectors. It may therefore assign equal scalar values:

$$s_{\boldsymbol{r}}(\boldsymbol{v}_1) = s_{\boldsymbol{r}}(\boldsymbol{v}_2) \geq 0,$$

even though $\boldsymbol{v}_1 \succ_{\mathrm{p}} \boldsymbol{v}_2$. However, it is certain that $\boldsymbol{v}_3$ does not get a positive value since it does not lie above the referent.

In contrast, an *order approximating* ASF must respect the full dominance relation and satisfy

$$s_{\boldsymbol{r}}(\boldsymbol{v}_1) > s_{\boldsymbol{r}}(\boldsymbol{v}_2).$$

However, it may assign a positive value to $\boldsymbol{v}_3$ if it lies within the tolerance region $\mathbb{R}^d_\delta$. This may complicate the interpretation of the results, as it could suggest that $\boldsymbol{v}_3$ is a viable alternative even though it is not inside the target region.

Weak Pareto oracle. We first show that evaluating a weak Pareto oracle Ω^τ can be framed as maximising an order representing ASF over a set of allowed policies Π. Since such ASFs guarantee that their maximum is reached within the target region at some weakly optimal solution, Theorem 5 follows immediately.

Theorem 5: ASF as a weak Pareto oracle

Let $s_{\boldsymbol{r}}$ be an order representing ASF. Then $\Omega^\tau(\boldsymbol{r}) = \arg\max_{\pi\in\Pi} s_{\boldsymbol{r}}(\boldsymbol{v}^\pi)$ with tolerance $\tau = 0$ is a valid weak Pareto oracle.

Proof. Let $s_{\boldsymbol{r}}$ be an order representing achievement scalarising function, and define a Pareto oracle $\Omega^\tau : \mathbb{R}^d \to \Pi$ by

$$\Omega^\tau(\boldsymbol{r}) = \arg\max_{\pi\in\Pi} s_{\boldsymbol{r}}(\boldsymbol{v}^\pi) = \pi^*.$$

Let $\boldsymbol{v}^*$ denote the expected return of π^*.

1. Case $\boldsymbol{v}^* \not> \boldsymbol{r}$: infeasibility.
If $\boldsymbol{v}^* \not> \boldsymbol{r}$, then by Definition 25 we have $s_{\boldsymbol{r}}(\boldsymbol{v}^*) \leq 0$. In that case, suppose for contradiction that there exists a feasible policy π' with $\boldsymbol{v}' > \boldsymbol{r}$ and π' weakly Pareto optimal. Then $s_{\boldsymbol{r}}(\boldsymbol{v}') > 0 \geq s_{\boldsymbol{r}}(\boldsymbol{v}^*)$, which contradicts the optimality of π^*. Hence, no such feasible weakly Pareto-optimal solution can exist, as required.

2. Case $\boldsymbol{v}^* > \boldsymbol{r}$: weak Pareto optimality.
If $\boldsymbol{v}^* > \boldsymbol{r}$, then π^* is a feasible policy. Suppose it is not weakly Pareto optimal. Then there exists another policy π' with return $\boldsymbol{v}'$ such that $\boldsymbol{v}' > \boldsymbol{v}^*$. Since $s_{\boldsymbol{r}}$ is order representing, it must satisfy $s_{\boldsymbol{r}}(\boldsymbol{v}') > s_{\boldsymbol{r}}(\boldsymbol{v}^*)$, contradicting the maximality of π^*.

In both cases, the oracle either returns a weakly Pareto-optimal solution that satisfies $\boldsymbol{v}^* > \boldsymbol{r}$, or it returns a point $\boldsymbol{v}^* \not> \boldsymbol{r}$ and thus certifies infeasibility. This completes the proof. □

Approximate Pareto oracle. By definition, an order approximating ASF attains its maximum at a Pareto-optimal solution. However, since such ASFs assign non-negative values to solutions outside the target region, this maximum may occur outside the desired area. To mitigate this, we introduce an *inherent tolerance* $\bar{\tau}$ and assume that the user-provided tolerance τ is strictly greater. In the following theorem, we show that this approach is valid and that the oracle returns a Pareto-optimal solution given the user-provided tolerance whenever one exists.

Theorem 6: ASF as an approximate Pareto oracle

Let $s_{\boldsymbol{r}}$ be an order approximating ASF and let $\boldsymbol{l} \in \mathbb{R}^d$ be a lower bound such that only referents $\boldsymbol{r}$ are selected when $\boldsymbol{r} \succeq_{\mathrm{p}} \boldsymbol{l}$. Then $s_{\boldsymbol{r}}$ has an inherent oracle tolerance $\bar{\tau} > 0$ and for any user-provided tolerance $\tau > \bar{\tau}$, $\Omega^{\tau}(\boldsymbol{r}) = \arg\max_{\pi \in \Pi} s_{\boldsymbol{r}+\tau}(\boldsymbol{v}^{\pi})$ is a valid approximate Pareto oracle.

Proof. Throughout, write

$$\mathbb{R}^d_{\delta} \;=\; \left\{ \boldsymbol{v} \in \mathbb{R}^d \,\middle|\, \operatorname{dist}\!\left(\boldsymbol{v}, \mathbb{R}^d_{\geq 0}\right) \leq \delta \|\boldsymbol{v}\| \right\}, \qquad 0 \leq \delta < 1.$$

We denote by $B(\boldsymbol{x}, \boldsymbol{y})$ the box defined by $\boldsymbol{x}$ and $\boldsymbol{y}$, i.e., $B(\boldsymbol{x}, \boldsymbol{y}) = \{\boldsymbol{v} \in \mathbb{R}^d \mid \boldsymbol{y} \succeq_{\mathrm{p}} \boldsymbol{v} \succeq_{\mathrm{p}} \boldsymbol{x}\}$.

1. No member of $\boldsymbol{l} + \mathbb{R}^d_{\delta}$ is dominated by $\boldsymbol{l}$.

Assume, towards a contradiction, that $\boldsymbol{v} = \boldsymbol{l} + \boldsymbol{x} \in \boldsymbol{l} + \mathbb{R}^d_{\delta}$ with $\boldsymbol{l} \succ_{\mathrm{p}} \boldsymbol{v}$. Then every component of $\boldsymbol{x}$ is non-positive and $\boldsymbol{x} \neq \boldsymbol{0}$, so the closest point in the non-negative orthant is $\boldsymbol{0}$:

$$\operatorname{dist}\!\left(\boldsymbol{x}, \mathbb{R}^d_{\geq 0}\right) = \|\boldsymbol{x}\|.$$

Since $\|\boldsymbol{x}\| \leq \delta \|\boldsymbol{x}\|$ is impossible when $\delta < 1$ and $\boldsymbol{x} \neq \boldsymbol{0}$, $\boldsymbol{l}$ is not dominated by any element of $\boldsymbol{l} + \mathbb{R}^d_{\delta}$.

2. Bounding the minimal shift $\bar{\tau}$.
We now establish a minimal shift $\bar{\tau} > 0$ such that every feasible solution $\boldsymbol{v}$ for which $s_{\boldsymbol{l}+\bar{\tau}}(\boldsymbol{v}) \geq 0$, we have $\boldsymbol{v} \succeq_{\mathrm{p}} \boldsymbol{l}$. Because Step 1 forces each $\boldsymbol{v} \in \boldsymbol{l} + \mathbb{R}^d_{\delta}$ either to equal $\boldsymbol{l}$ or to exceed it in some coordinate, such a $\bar{\tau}$ must satisfy

$$0 < \bar{\tau} \leq \left\|\boldsymbol{v}^{\mathrm{i}} - \boldsymbol{l}\right\|_{\infty}$$

Let us now formally define $\bar{\tau}$ for an order approximating ASF with approximation constant δ,

$$\bar{\tau} = \inf \left\{ 0 < \tau \leq \|\boldsymbol{v}^{\mathrm{i}} - \boldsymbol{l}\|_{\infty} \mid \left(\boldsymbol{l} + \tau + \mathbb{R}^d_{\delta}\right) \cap \{\boldsymbol{v} \in \mathbb{R}^d \mid \boldsymbol{v}^{\mathrm{i}} \succeq_{\mathrm{p}} \boldsymbol{v}\} \subseteq B(\boldsymbol{l}, \boldsymbol{v}^{\mathrm{i}}) \right\}.$$

In Fig. 3.5 we illustrate that this shift ensures all feasible solutions with non-negative values are inside the box. Observe, however, that by the nature of this shift, it can also ensure that some feasible solutions in the bounding box are excluded from the non-negative set.

3. Validity for the specific referent $\boldsymbol{r} = \boldsymbol{l}$.
Choose any $\tau > \bar{\tau}$ and set

$$\pi^* = \Omega^{\tau}(\boldsymbol{l}) = \arg\max_{\pi \in \Pi} s_{\boldsymbol{l}+\tau}\!\left(\boldsymbol{v}^{\pi}\right)$$

Assume that there exists a Pareto-optimal policy π' with return $\boldsymbol{v}' \succeq_{\mathrm{p}} \boldsymbol{l} + \tau$. Then, by the definition of the order approximating ASF, we have $s_{\boldsymbol{l}+\tau}(\boldsymbol{v}') \geq 0$. Since π^* maximises $s_{\boldsymbol{l}+\tau}$, it follows that $s_{\boldsymbol{l}+\tau}(\boldsymbol{v}^*) \geq 0$.
By construction of $\bar{\tau}$, any solution $\boldsymbol{v}$ with $s_{\boldsymbol{l}+\tau}(\boldsymbol{v}) \geq 0$ must satisfy $\boldsymbol{v} \succeq_{\mathrm{p}} \boldsymbol{l} + (\tau - \bar{\tau})$, so we conclude that

$$\boldsymbol{v}^* \succeq_{\mathrm{p}} \boldsymbol{l} + (\tau - \bar{\tau}).$$

Moreover, since the ASF is strictly increasing, no solution can dominate $\boldsymbol{v}^*$, and therefore π^* is Pareto optimal.

4. Extension to every dominating referent $\boldsymbol{r} \succeq_{\mathrm{p}} \boldsymbol{l}$.
Let $\boldsymbol{r} = \boldsymbol{l} + \boldsymbol{x}$ for some non-negative vector $\boldsymbol{x} \in \mathbb{R}^d_{\geq 0}$. We show that the same containment property holds under this shift. From Step 2, we have:

$$\left(\boldsymbol{l} + \tau + \mathbb{R}^d_\delta\right) \cap \left\{\boldsymbol{v} \in \mathbb{R}^d \mid \boldsymbol{v}^{\mathrm{i}} \succeq_{\mathrm{p}} \boldsymbol{v}\right\} \subseteq B(\boldsymbol{l}, \boldsymbol{v}^{\mathrm{i}}).$$

This directly implies that,

$$\left(\boldsymbol{l} + \tau + \mathbb{R}^d_\delta\right) \cap \left\{\boldsymbol{v} \in \mathbb{R}^d \mid \boldsymbol{v}^{\mathrm{i}} - \boldsymbol{x} \succeq_{\mathrm{p}} \boldsymbol{v}\right\} \subseteq B(\boldsymbol{l}, \boldsymbol{v}^{\mathrm{i}} - \boldsymbol{x}).$$

By applying the rigid shift $\boldsymbol{x}$ to both the target region and box, and recalling that $\boldsymbol{r} = \boldsymbol{l} + \boldsymbol{x}$, we conclude:

$$\left(\boldsymbol{r} + \tau + \mathbb{R}^d_\delta\right) \cap \left\{\boldsymbol{v} \in \mathbb{R}^d \mid \boldsymbol{v}^{\mathrm{i}} \succeq_{\mathrm{p}} \boldsymbol{v}\right\} \subseteq B(\boldsymbol{r}, \boldsymbol{v}^{\mathrm{i}}).$$

This establishes that the containment guarantee required for oracle validity holds for all referents $\boldsymbol{r} \succeq_{\mathrm{p}} \boldsymbol{l}$, and thus the same reasoning from Step 3 applies. □

IPRO success condition. In the proof of Theorem 6, we allow the oracle to return a Pareto-optimal policy $\boldsymbol{v}^*$ satisfying

$$\boldsymbol{v}^* \succeq_{\mathrm{p}} \boldsymbol{l} + (\tau - \bar{\tau}). \tag{3.2}$$

This differs subtly from Definition 19, which accepts a solution only when $\boldsymbol{v} \succeq_{\mathrm{p}} \boldsymbol{r} + \tau$. To ensure compatibility between the ASF-based oracle and IPRO, we therefore adjust IPRO's success criterion to deem an oracle call successful whenever $\boldsymbol{v} \succeq_{\mathrm{p}} \boldsymbol{l} + (\tau - \bar{\tau})$, where $\bar{\tau}$ is the minimal shift needed to keep the ASF-admissible region inside the local box:

$$\bar{\tau} \;=\; \inf\left\{\, 0 < \tau' \leq \|\boldsymbol{v}^{\mathrm{i}} - \boldsymbol{l}\|_\infty \;\middle|\; \left(\boldsymbol{l} + \tau' + \mathbb{R}^d_\delta\right) \cap \{\boldsymbol{v} \in \mathbb{R}^d \mid \boldsymbol{v}^{\mathrm{i}} \succeq_{\mathrm{p}} \boldsymbol{v}\} \subseteq B(\boldsymbol{l}, \boldsymbol{v}^{\mathrm{i}}) \right\}. \tag{3.3}$$

Because $\tau > \bar{\tau} > 0$, the margin $\tau - \bar{\tau}$ is strictly positive, so IPRO merely accepts a smaller improvement than the user-specified tolerance τ. This adjustment is necessary (see Fig. 3.5b), where $\boldsymbol{v}_4$ is Pareto optimal within the ASF region yet fails $\boldsymbol{v}_4 \succeq_{\mathrm{p}} \boldsymbol{r} + \tau$. The modification is harmless for convergence: each successful iteration still improves the lower bound by at least $\tau - \bar{\tau} > 0$.

Figure 3.5: **(a)** A possible non-negative set (shaded) for an order approximating ASF with referent $\boldsymbol{l}$. **(b)** Shifting $\boldsymbol{l}$ by $\bar{\tau}$ ensures that all feasible solutions with non-negative values are in the box $B(\boldsymbol{l}, \boldsymbol{v}^{\mathrm{i}})$.

4.2 Practical ASF Selection

As a practical choice, we use the augmented Chebyshev scalarisation [Nikulin et al., 2012], specified in Eq. (3.4).

$$s_{\boldsymbol{r}}(\boldsymbol{v}) = \min_{j \in \{1,\ldots,d\}} \lambda_j (v_j - r_j) + \beta \sum_{j=1}^{d} \lambda_j (v_j - r_j) \tag{3.4}$$

Here, $\boldsymbol{\lambda} > 0$ serves as a normalisation constant for the different objectives, and β is a parameter determining the strength of the augmentation term. Selecting $\boldsymbol{\lambda} = (\boldsymbol{v}^{\mathrm{i}} - \boldsymbol{v}^{\mathrm{n}})^{-1}$ scales any vector $\boldsymbol{v}$ relative to the distance between the nadir $\boldsymbol{v}^{\mathrm{n}}$ and ideal $\boldsymbol{v}^{\mathrm{i}}$, thereby ensuring a balanced scale across all objectives. This normalisation prevents the dominance of one objective over another, a challenge that is otherwise difficult to overcome [Abdolmaleki et al., 2020].

Equation (3.4) serves as a weak or approximate Pareto oracle, depending on the augmentation parameter β. When $\beta = 0$, the augmentation term is cancelled and the minimum ensures that only vectors in the target region have non-negative values. However, optimising a minimum may result in weakly Pareto-optimal solutions (e.g. $(1, 2)$ and $(1, 1)$ share the same minimum). For $\beta > 0$, the optimal solution will be Pareto optimal (the sum of $(1, 2)$ is greater than that of $(1, 1)$) but may exceed the target region.

4.3 Alternative Oracle Implementations

While Theorems 5 and 6 establish that Pareto oracles may be implemented using an ASF, optimising the ASF over a given policy class may still be challenging. Here, we show that alternative implementations can be derived from existing literature.

Convex MDP. A convex MDP $\mathcal{M}_{\text{conv}}$ is a generalisation of an MDP, where an agent seeks to minimise a convex function (or equivalently maximise a concave function) over a convex set of admissible occupancy measures [Zahavy et al., 2021]. Let $\boldsymbol{\mathcal{K}}_\gamma$ be the set of discounted state occupancy measures for some discount factor γ. The expected return $\boldsymbol{v}^\pi$ of some policy π can be written as a linear function of the occupancy measure of the policy ρ^π and the reward function of the MDP, $\boldsymbol{v}^\pi = \sum_{s,a} \boldsymbol{r}(s,a)\rho^\pi(s,a)$. Proposition 1 then follows immediately.

Proposition 1: Convex MDP Pareto oracle

Let $\boldsymbol{\mathcal{M}} = \langle \mathcal{S}, \mathcal{A}, p, \boldsymbol{r}, p_0, \gamma \rangle$ be a MOMDP with d objectives. For a given oracle tolerance $\tau \geq 0$ and referent $\boldsymbol{r}$, we define a convex MDP $\mathcal{M}_{\text{conv}}$ with the same $\mathcal{S}, \mathcal{A}, p, \gamma$ and p_0 as $\boldsymbol{\mathcal{M}}$. For $s_{\boldsymbol{r}}$ defined in Eq. (3.4) and Π the set of stationary policies, $\Omega^\tau(\boldsymbol{r}) = \arg\max_{\rho^\pi \in \boldsymbol{\mathcal{K}}_\gamma} s_{\boldsymbol{r}+\tau}(\boldsymbol{v}^\pi)$ is a valid weak or approximate Pareto oracle.

Proof. Since Eq. (3.4) is concave for any referent $\boldsymbol{r}$ and the composition of a concave function and linear function preserves concavity, the problem is concave. Furthermore, $\boldsymbol{\mathcal{K}}_\gamma$ is by definition a convex polytope for the set of stationary policies. As such, $\mathcal{M}_{\text{conv}}$ is a convex MDP and since $s_{\boldsymbol{r}}$ can be constructed as both an order representing and order approximating ASF, Theorem 5 and Theorem 6 can be applied. □

This reformulation enables the use of techniques with strong theoretical guarantees. For instance, Zhang et al. [2020] propose a policy gradient method that converges to the global optimum, and Zahavy et al. [2021] introduce a meta-algorithm using standard RL algorithms that converges to the optimal solution with any tolerance, assuming reasonably low-regret algorithms. Additionally, it has been demonstrated that for any convex MDP, a mean-field game can be constructed, for which any Nash equilibrium in the game corresponds to an optimum in the convex MDP [Geist et al., 2022].

Constrained MDP. A constrained MDP $\mathcal{M}_{\text{const}}$ is an MDP, augmented with a set of m auxiliary cost functions $C_j : \mathcal{S}\times\mathcal{A}\times\mathcal{S} \to \mathbb{R}$ and related limit c_j [Altman, 1999]. Let $J_{C_j}(\pi)$ denote the expected discounted return of policy π for the auxiliary cost function C_j. The feasible policies from a given class of policies Π is then $\Pi_C = \{\pi \in \Pi \mid \forall i, J_{C_j}(\pi) \geq c_j\}$. Finally, the reinforcement learning problem in a CMDP is as follows,

$$\pi^* = \arg\max_{\pi \in \Pi_C} v^\pi. \tag{3.5}$$

Treating the referent as lower bound constraints and maximising the sum of rewards is shown to result in a Pareto-optimal solution inside the target region if one exists. One important advantage of this oracle is that there is no inherent tolerance, as required when employing an order approximating ASF construction (see Theorem 6), and so τ can be chosen freely.

Proposition 2: Constrained MDP Pareto oracle

Let $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, p, \boldsymbol{r}, p_0, \gamma \rangle$ be a MOMDP with d objectives. For a given oracle tolerance $\tau > 0$ and referent $\boldsymbol{r}$, we define a constrained MDP $\mathcal{M}_{\text{const}}$ with the same $\mathcal{S}, \mathcal{A}, p, \gamma$ and p_0 as $\mathcal{M}$. $\mathcal{M}_{\text{const}}$ has d cost functions corresponding to the original d reward function with limits $\boldsymbol{r} + \tau$ and $r(s,a) = \langle \boldsymbol{r}(s,a), \mathbf{1} \rangle$. Then $\Omega^{\tau}(\boldsymbol{r}) = \arg\max_{\pi \in \Pi_C} v^{\pi}$ is a valid approximate Pareto oracle.

Proof. Assume the construction outlined in the theorem and that there exists a Pareto-optimal policy π such that $\boldsymbol{v}^{\pi} \succeq_{\text{p}} \boldsymbol{r} + \tau$. Then Π_C is non-empty and the Pareto oracle $\Omega^{\tau}(\boldsymbol{r}) = \arg\max_{\pi \in \Pi_C} v^{\pi}$ returns a Pareto-optimal policy π^* with expected return $\boldsymbol{v}^*$ such that $\boldsymbol{v}^* \succeq_{\text{p}} \boldsymbol{r} + \tau$. If π^* is not Pareto optimal, there exists a policy π' with expected return $\boldsymbol{v}'$ such that $\boldsymbol{v}' \succ_{\text{p}} \boldsymbol{v}^*$. This then implies that,

$$\sum_{j \in \{1,\dots,d\}} v'_j > \sum_{j \in \{1,\dots,d\}} v^*_j$$

which leads to a contradiction. □

Several algorithms with strong theoretical foundations have been proposed for solving constrained MDPs in a reinforcement learning context [Achiam et al., 2017; Ding et al., 2021]. When the model is known and the state and action sets are finite, an optimal stochastic policy can be computed in polynomial time [Altman, 1999]. Together with Theorem 3, this guarantees that IPRO obtains a Pareto front of stochastic policies in polynomial time, recovering prior guarantees [Papadimitriou and Yannakakis, 2000; Chatterjee et al., 2006]. Although computing optimal stationary deterministic policies in constrained MDPs is NP-complete [Feinberg, 2000], mixed-integer linear programming has been shown to be effective in practice [Dolgov and Durfee, 2005].

5 Deterministic Memory-Based Policies

As shown in Sections 2 and 3, IPRO obtains the Pareto front for any policy class with a valid Pareto oracle. We now develop a Pareto oracle specifically for deterministic memory-based policies, a class for which there is currently no method that can learn non-convex Pareto fronts in general MOMDPs.

5.1 Motivation

In single-objective MDPs, an optimal deterministic policy is always guaranteed to exist. However, in MOMDPs, this result does not hold, and stochastic policies may be required to capture all solutions on the Pareto front. Nevertheless, in practical applications where interpretability, explainability, and safety are critical, deterministic policies remain preferable, as noted in related work [Delgrange et al., 2020; Hayes et al., 2022a]. For example, in medical applications, decisions must be interpretable, with deterministic treatment protocols being essential.

To avoid the need for randomisation in policies, memory can be used to learn additional policies that provide alternative trade-offs for the decision-maker. Consider a pick-up and delivery MOMDP where the agent can either collect a package (yielding a reward of $(3, 0)$) or deliver a package (yielding $(0, 3)$), with both actions returning to the same state. Without memory, deterministic policies are restricted to always collecting or always delivering, resulting in discounted returns of $(3/1-\gamma, 0)$ or $(0, 3/1-\gamma)$. By incorporating memory, the agent can condition its actions on past behaviour, for instance, delivering after each collection, achieving a discounted return of $(3/1-\gamma^2, 3\gamma/1-\gamma^2)$. This demonstrates how memory increases the set of feasible Pareto-optimal policies, as shown earlier by White [1982].

5.2 Practical Implementation

As shown in Section 4.3, Pareto oracles with strong guarantees can be constructed for stochastic policies. In contrast, for memory-based deterministic policies, finding a Pareto-optimal solution that dominates a given referent is NP-hard [Chatterjee et al., 2006]. To address this, we extend single-objective reinforcement learning algorithms to optimise the achievement scalarising function (ASF) in Eq. (3.4). Following standard practice in MORL, we encode memory by augmenting the observation with the accrued reward at time step t, defined as

$$\boldsymbol{q}_t := \sum_{k=0}^{t-1} \gamma^k \boldsymbol{r}(s_k, a_k, s_{k+1}). \tag{3.6}$$

Order approximating ASF. When using Eq. (3.4) with $\beta > 0$, we are guaranteed a Pareto-optimal solution. However, directly determining the associated minimal shift $\bar{\tau}$ is often impractical. As a result, our implementation instead maximises $s_{\boldsymbol{r}}(\boldsymbol{v}^\pi)$ as in the weak Pareto oracle. Although this introduces the risk of producing solutions outside the intended target region, this can be mitigated through additional bookkeeping and, in practice, does not affect performance.

Value-based. We extend the GGF-DQN algorithm, which optimises for the generalised Gini welfare of the expected returns [Siddique et al., 2020], to optimise any scalarisation

function f. We note that GGF-DQN is itself an extension of DQN [Mnih et al., 2015]. Concretely, we train a Q-network such that $\boldsymbol{Q}(s_t, a_t) = \boldsymbol{r} + \gamma \boldsymbol{Q}(s_{t+1}, a^*)$ where the optimal action a^* is computed using the accrued reward and scalarisation function f,

$$a^* = \arg\max_{a \in \mathcal{A}} f\left(\boldsymbol{q}_{t+1} + \gamma^{t+1} \boldsymbol{Q}\left(s_{t+1}, a\right)\right). \tag{3.7}$$

A limitation of this action selection method is that it is not aligned with the optimisation objective, which instead requires evaluating

$$f(\boldsymbol{v}^\pi) = f\left(\mathop{\mathbb{E}}_{\pi, p_0}\left[\boldsymbol{q}_{t+1}\right] + \gamma^{t+1} \boldsymbol{Q}\left(s_{t+1}, a\right)\right). \tag{3.8}$$

Since computing the expectation of $\boldsymbol{q}_{t+1}$ is usually impractical, we use the observed accrued reward as a substitute.

Policy gradient. We extend A2C [Mnih et al., 2016] and PPO [Schulman et al., 2017] to optimise $J(\pi) = f(\boldsymbol{v}^\pi)$, where f is a scalarisation function and π a parametrised policy with parameters θ. For differentiable f, the policy gradient becomes $\nabla_\theta J(\pi) = f'(\boldsymbol{v}^\pi) \cdot \nabla_\theta \boldsymbol{v}^\pi(s_0)$ [Reymond et al., 2023]. To ensure deterministic policies, we take actions according to $\arg\max_{a \in \mathcal{A}} \pi(a|s)$ during policy evaluation. Although this potentially changes the policy, effectively employing a policy that differs from the one initially learned, empirical observations suggest that these algorithms typically converge toward deterministic policies in practice. Furthermore, recent work has theoretically analysed this practice and found that under some assumptions convergence to the optimal deterministic policy is guaranteed [Montenegro et al., 2024].

Extended networks. Rather than making separate calls to one of the previous reinforcement learning methods for each oracle evaluation, we employ extended networks [Abels et al., 2019] to improve sample efficiency. Concretely, we extend our actor and critic networks to take a referent as additional input, enabling their reuse across IPRO iterations. We further introduce a pre-training phase, where a policy is trained on randomly sampled referents for a fixed number of episodes. To maximise the benefit of this pre-training, we perform additional off-policy updates for referents not used in data collection. While this has no effect on DQN, policy gradient methods require alignment between behaviour and target policies. We address this through importance sampling in A2C and an off-policy adaptation of PPO [Meng et al., 2023].

6 Experiments

To test the performance of IPRO, we combine it with the modified versions of DQN, A2C, and PPO proposed in Section 5 as approximate Pareto oracles that optimise the augmented Chebyshev scalarisation function in Eq. (3.4). All experiments are repeated over five seeds.

6.1 Evaluation Metrics

Evaluating MORL algorithms poses significant challenges due to the difficulty in measuring the quality of a Pareto front [Felten et al., 2023]. To address this, we compute two different metrics during learning and one for the final returned front.

We first consider the hypervolume, defined in Eq. (3.9), a well-established measure in MORL. The hypervolume quantifies the volume of the dominated region formed by an estimate of the Pareto front relative to a specified reference point. However, a notable drawback of this metric is that the choice of reference point significantly influences the obtained values, potentially distorting the results. In practice, we use the nadir as the reference point.

$$HV(\mathcal{F}; \boldsymbol{r}) = \text{vol}\left(\bigcup_{\boldsymbol{v}\in\mathcal{F}} [\boldsymbol{r}, \boldsymbol{v}]\right) \tag{3.9}$$

Following the approach outlined by Hayes et al. [2022a], we further evaluate all algorithms using utility-based metrics. Concretely, we compute the maximum utility loss (MUL) [Zintgraf et al., 2015] for a class of utility functions $\mathcal{U}$ compared to the true Pareto front $\mathcal{F}^*$ as

$$MUL(\mathcal{F}_t; \mathcal{F}^*) = \max_{u\in\mathcal{U}} \left[\max_{\boldsymbol{v}\in\mathcal{F}^*} u(\boldsymbol{v}) - \max_{\boldsymbol{v}\in F_t} u(\boldsymbol{v})\right]. \tag{3.10}$$

We generate piecewise linear, monotonically increasing functions $u : [\boldsymbol{v}^{\text{n}}, \boldsymbol{v}^{\text{i}}] \to [0, 1]$ by sampling a grid of positive numbers as gradients. The function value at $\boldsymbol{v}$ is obtained by summing the preceding gradients and rescaling. Our grid uses six cells per dimension, with gradients drawn uniformly from $[0, 5)$. Notably, this method produces functions biased towards risk aversion. Furthermore, we estimate $\mathcal{F}^*$ as the union of all final Pareto fronts obtained by both IPRO and the baseline algorithms across all runs. Lastly, we evaluate the quality of the final Pareto front by its true error as defined in Eq. (3.1). This metric provides an additional measure of how closely the final approximation aligns with the true Pareto front.

6.2 Baselines

As IPRO is the first general-purpose method capable of learning the Pareto front for arbitrary policies in general MOMDPs, we select baselines that are tailored to specific settings. To ensure a fair comparison, we extend all baselines to accumulate their empirical Pareto fronts across evaluation steps, guaranteeing the same monotonic improvement as in IPRO.

Convex hull algorithms. We evaluate two state-of-the-art convex hull algorithms: Generalised Policy Improvement - Linear Support (GPI-LS) [Alegre et al., 2023] and Envelope Q-Learning (EQL) [Yang et al., 2019]. Both algorithms train vectorial Q-networks that can be dynamically adjusted with given weights to produce a scalar return.

Pareto front algorithm. We include Pareto Conditioned Networks (PCN), which were specifically designed to learn the Pareto front of deterministic policies in deterministic MOMDPs [Reymond et al., 2022]. PCN trains a network to generalise across the full Pareto front by predicting the "return-to-go" for any state and selecting the action that best aligns with the desired trade-off.

6.3 Environments

We evaluate the algorithms across three well-known benchmark environments [Felten et al., 2023]. We initialise each experiment with predefined minimal and maximal points to establish the bounding box of the environment. It is important to emphasise that these points can be obtained using conventional reinforcement learning algorithms without requiring any modifications, justifying their omission from our evaluation process.

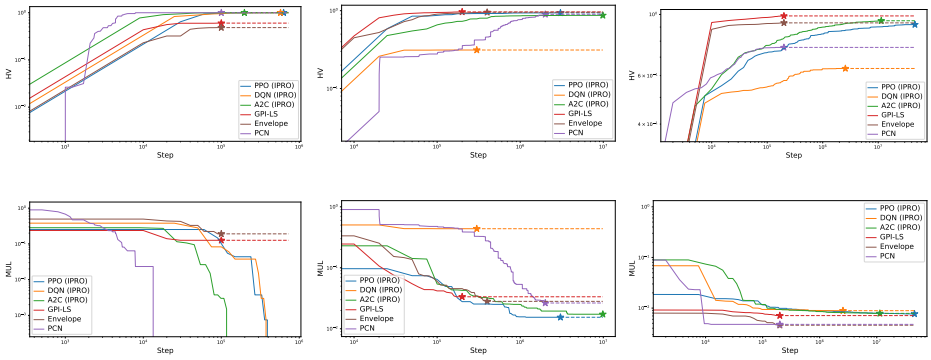
Deep Sea Treasure (DST). We initialise IPRO with $(124, -50)$ and $(1, -1)$ as the maximal points and give $(0, -50)$ as the only minimal point. We set the discount factor to 1, signifying no discounting, and maintain a fixed time horizon of 50 time steps for each episode. We note that we one-hot encode the observations due to the discrete nature of the state space. Finally, we set $\tau = 0$ to allow IPRO to find the complete Pareto front in this environment.

Minecart. In the Minecart environment, we set $\gamma = 0.98$ to align with related work. For minimal points, IPRO is initialised with the nadir $(-1, -1, -200)$ for each dimension. For maximal points, we consider the nadir and set each dimension to its theoretical maximum: $(1.5, -1, -200), (-1, 1.5, -200), (-1, -1, 0)$. Our reference point is also the nadir and the time horizon is 1000. A tolerance of 1×10^{-15} was used.

MO-Reacher. In the Reacher environment, we use $(-50, -50, -50, -50)$ in each dimension as the minimal points, and similarly, set this vector to 40 for each dimension for the maximal points. The discount factor γ is set to 0.99. The reference point is again set to the nadir, a time horizon of 50 was used and tolerance was set to 1×10^{-15}.

6.4 Results

Deep Sea Treasure ($d = 2$). DST is a deterministic environment where a submarine seeks treasure while minimising fuel consumption. DST has a Pareto front with solutions in concave regions [Vamplew et al., 2011], making it impossible for the convex hull algorithms to recover all Pareto-optimal solutions. This limitation is evident in Section 6.3



(a) Deep Sea Treasure (b) Minecart (c) MO-Reacher

Figure 3.6: The mean hypervolume (top) and maximum utility loss (bottom) scaled between zero and one with 95-percentile interval on a log-log scale. Stars indicate when each algorithm finishes. The pre-training phase of IPRO is not shown.

and Fig. 3.6a where GPI-LS and EQL exhibit significantly inferior performance compared to IPRO and PCN. Notably, IPRO and PCN recover the complete Pareto front in the majority of runs; however, IPRO tends to require more samples. This discrepancy can be attributed to the fact that IPRO learns only one Pareto-optimal solution per iteration, whereas PCN concurrently learns multiple policies. Nonetheless, this concurrent learning approach for PCN comes at the expense of theoretical guarantees. When comparing the ε metric (Table 3.1), we observe that IPRO learns high-quality approximations and consistently learns the complete Pareto front when paired with PPO and DQN. The convex hull methods naturally have poorer approximations.

Minecart ($d = 3$)**.** Minecart is a stochastic environment where the agent collects two types of ore while minimising fuel consumption [Abels et al., 2019]. Since this environment was designed to induce a convex Pareto front, GPI-LS and EQL are expected to perform optimally. We find that IPRO achieves comparable hypervolume results and demonstrates superior maximum utility loss (MUL) compared to all other baselines when using policy gradient oracles. The anytime property of IPRO is particularly evident in the MUL results, as its Pareto front continues to improve up to 10^7 steps. In Table 3.1, the ε distances for the policy gradient methods are shown to be competitive. However, we observe that the DQN variant struggles to learn a high-quality Pareto front, which may be attributed to the algorithm's ad hoc nature. This suggests that future research focusing on value-based oracles could provide significant benefits.

Table 3.1: The minimum ε shift necessary to obtain any undiscovered Pareto-optimal solution.

Env	Algorithm	ε
DST	IPRO (PPO)	**0.0** $\pm$ 0.0
	IPRO (A2C)	0.2 $\pm$ 0.4
	IPRO (DQN)	**0.0** $\pm$ 0.0
	PCN	**0.0** $\pm$ 0.0
	GPI-LS	5.2 $\pm$ 2.71
	Envelope	28.6 $\pm$ 46.77
Minecart	IPRO (PPO)	0.66 $\pm$ 0.07
	IPRO (A2C)	0.54 $\pm$ 0.11
	IPRO (DQN)	1.11 $\pm$ 0.01
	PCN	0.67 $\pm$ 0.2
	GPI-LS	**0.42** $\pm$ 0.0
	Envelope	**0.42** $\pm$ 0.01
MO-Reacher	IPRO (PPO)	5.75 $\pm$ 1.22
	IPRO (A2C)	**2.84** $\pm$ 0.39
	IPRO (DQN)	15.02 $\pm$ 1.42
	PCN	18.95 $\pm$ 1.76
	GPI-LS	8.5 $\pm$ 0.12
	Envelope	11.41 $\pm$ 0.62

MO-Reacher ($d = 4$)**.** MO-Reacher is a deterministic environment where four balls are arranged in a circle and the goal is to minimise the distance to each ball. Since it is deterministic and has a mostly convex Pareto front, it suits all baselines. In Fig. 3.6c, we find that IPRO obtains a hypervolume and maximum utility loss competitive to the baselines. Additionally, the policy gradient oracles result in the best approximations to the Pareto front according to the ε metric in Table 3.1. Due to IPRO's iterative mechanism, this comes at the price of increased sample complexity, while the baselines benefit from learning multiple policies concurrently.

7 Conclusion

This chapter addressed the problem of approximating the Pareto front in MOMDPs through a decomposition-based approach. Such an approach enables the reuse of powerful single-objective algorithms, such as DQN, A2C, and PPO, within the multi-objective setting, and allows their theoretical guarantees to be preserved at the level of the overall method.

To this end, we introduced *Iterated Pareto Referent Optimisation* (IPRO), a general framework that iteratively queries a Pareto oracle with strategically selected referents and uses the returned solutions to discard dominated and infeasible regions of the search space. We formalised two types of Pareto oracles, weak and approximate, and showed that both can be implemented using single-objective scalarisation. Each variant supports a different form of theoretical guarantee: weak oracles ensure convergence to the true Pareto front, while approximate oracles guarantee bounded error with a provable upper bound on the number of iterations. Our analysis of IPRO is fully independent of the oracle implementation and of the underlying problem, establishing its broad applicability to various multi-objective settings.

A particular strength of IPRO is its compatibility with deterministic, memory-based policies. Such policies are often preferred in settings where interpretability or safety is critical, yet they remain poorly supported by existing MORL methods. IPRO can directly accommodate these policies and in experiments consistently produces high-quality Pareto fronts without relying on domain knowledge.

Follow-up challenges. While the main challenge posed in this chapter was to develop a principled method that can learn the Pareto front through single-objective RL, several important follow-up challenges remain.

- **Leveraging information across iterations.** A key challenge is to exploit information from earlier iterations to improve the efficiency of later ones. Since nearby referents in objective space often yield similar solutions, prior optimisation runs could be reused to initialise subsequent ones or guide search more effectively. This suggests natural connections to paradigms such as few-shot [Touati and Ollivier, 2021] and transfer learning [Barreto et al., 2017], where prior knowledge is leveraged to accelerate adaptation to new tasks.
- **Concurrent learning of multiple policies.** Extending IPRO to support the simultaneous training of multiple policies may improve sample efficiency and accelerate coverage of the Pareto front. Rather than training each referent to completion, one could alternate between short bursts of training across referents, allowing transfer between partially learned solutions. To preserve theoretical guarantees, it may be preferable to delay updates to the infeasible set and focus initially on expanding the dominated region.
- **Evaluating alternative oracle implementations.** The performance of IPRO under different oracle designs remains an open question. We introduced the convex MDP and constrained MDP oracles in Section 4.3, but deferred their empirical evaluation. Comparing these oracles in terms of sample efficiency, convergence speed, and Pareto front quality would provide valuable insight into their practical trade-offs.

4 Distributional Multi-Objective Reinforcement Learning

This chapter is based on *Distributional Multi-Objective Decision Making* [Röpke et al., 2023b], a project undertaken during a research visit at the University of Galway. All proofs have been integrated into the main text and our code is available at `https://github.com/wilrop/distributional-dominance`.

1 Introduction

In Chapter 3, we focused on learning a Pareto front by leveraging single-objective reinforcement learning. The Pareto front comprises policies that yield Pareto-optimal expected returns and therefore includes all policies that are optimal for decision-makers maximising their utility over these expected outcomes [Hayes et al., 2022a]. Although it is commonly used in practice and often assumed axiomatically to contain an optimal policy for any decision-maker [Roijers et al., 2013], it is known that the Pareto front does not necessarily capture all optimal policies for agents who optimise their *expected utility* directly [Hayes et al., 2022c].

This chapter addresses the following challenge introduced in Chapter 1:

> **Challenge 2:** *If decision-makers maximise expected utility, what solution concept captures optimality in multi-objective settings and how can it be computed?*

To address this challenge, we draw inspiration from advances in *distributional reinforcement learning*, which have proven highly effective in the single-objective setting for improving stability, sample efficiency, and enabling risk-aware decision-making [Bellemare et al., 2023; Lim and Malik, 2022]. Extending this idea to the multi-objective case allows us to reason directly about the full *return distribution* of each policy rather than only its expectation, thereby capturing richer information about uncertainty and trade-offs across objectives. In single-objective decision theory, reasoning about full return distributions offers similar benefits, as dominance relations such as *first-order stochastic dominance* provide a way to determine when one distribution is preferred over another by broad classes of decision-makers, even without knowing their exact utility functions [Fishburn, 1974; Bawa et al., 1985].

We build on these insights to introduce a novel dominance criterion, called *distributional dominance*, which compares policies directly based on their return distributions. This criterion extends first-order stochastic dominance to the multivariate case [Denuit et al., 2013; Levy, 2016a]. Building on this notion, we define the *distributional undominated set (DUS)* as a new solution concept and prove that it includes all policies that are optimal for *multivariate risk-averse* expected utility maximisers [Richard, 1975]. Moreover, we show that the DUS contains the Pareto front and can therefore serve as a principled foundation for decision support. We further propose a smaller solution set, the *convex DUS (CDUS)*, which is also optimal for multivariate risk-averse expected utility maximisers. While the CDUS and the Pareto front are generally incomparable, both sets include the convex hull of achievable expected returns. Overall, we promote a general *learn large, prune later* strategy, in which one first learns a comprehensive solution set and then prunes it in a preference-aware manner. This offers a principled way to support decision-makers in challenging settings where preferences are unknown a priori.

In practical decision support applications, it is crucial to keep solution sets compact without compromising optimality. Defining minimal yet sufficient solution concepts, along with algorithms that efficiently prune excess policies, is therefore essential for tractable and effective deployment [Taboada et al., 2007]. To this end, we develop algorithms to prune a given policy set to its DUS or CDUS. Since the quality of the pruned set depends on the input, we extend Pareto Q-learning [Van Moffaert and Nowé, 2014] to estimate return distributions and retain only those policies that are not distributional dominated. We evaluate our approach on randomly generated MOMDPs of varying sizes, comparing the sizes of the resulting sets after pruning. In addition, we present a case study where the DUS reveals optimal policies that are excluded from the Pareto front. Throughout, the focus is on ensuring that the final sets remain compact and tractable, thus enabling efficient and informed policy selection by decision-makers.

Contributions. The main contributions of this chapter are as follows:

1. **Distributional solution sets.** We introduce the distributional undominated set (DUS) and its convex variant (CDUS), and characterise their optimality for different classes of decision-makers.

2. **A unified taxonomy of solution sets.** We formally describe the relationships between the DUS, CDUS, Pareto front, and convex hull: the DUS contains both the Pareto front and the CDUS, while the latter two are generally incomparable. Both the Pareto front and the CDUS include the convex hull.

3. **Pruning algorithms.** We develop procedures to prune a policy set to its DUS or Pareto front, and introduce a linear programming method for pruning to the CDUS.

4. **Learning the solution set.** We extend Pareto Q-learning to estimate return distributions and learn policies in the DUS. We empirically evaluate the resulting sets at each pruning stage and demonstrate the practical value of the DUS in a decision support case study.

Related work. Stochastic dominance has long been used in finance and economics [Levy, 2016b], and has more recently been applied to decision-making problems in RL. In single-objective settings, Epshteyn and DeJong [2006] employ stochastic dominance to learn optimal policies in MDPs with incomplete specifications. Martin et al. [2020] propose a risk-aware distributional algorithm that uses stochastic dominance at decision time to select actions. Techniques from stochastic dominance have also been used to analyse the theoretical properties of distributional RL [Rowland et al., 2018].

The distributional approach has become an active area of research in both single- and multi-objective settings. For a detailed overview of techniques in the single-objective case, we refer to Bellemare et al. [2023]. In the multi-objective setting, Hayes et al. [2021] and Reymond et al. [2023] introduce single-policy algorithms capable of learning policies for non-linear utility functions under the expected scalarised return (ESR) criterion. Wiltzer et al. [2024] present tractable distributional MORL algorithms with guarantees of convergence to the multivariate return distribution. Beyond single-policy methods, Hayes et al. [2022b] present a multi-policy distributional value iteration algorithm that computes a set of policies for the ESR criterion, referred to as the *ESR set*. This set is the first solution concept designed for multi-objective sequential decision-making under expected utility maximisation, and it uses strict first-order stochastic dominance to determine policy inclusion. The ESR set has been shown to contain all optimal policies for risk-averse decision-makers, but implicitly assumes independence among the components of the return vector [Hayes et al., 2022c].

2 Distributional Decision-Making

While much of multi-objective RL and optimisation focuses on returning the Pareto front, we demonstrate that this does not cover the full range of optimal policies. Specifically, for decision-makers optimising their expected utility, the best policy in the Pareto front may still be significantly worse than a Pareto-dominated policy. To overcome this, we propose a novel dominance criterion and subsequently construct a solution set based on this criterion.

2.1 Motivation

To understand why it is necessary to construct these novel solution sets, and in particular why a distributional approach is appropriate, we first consider a motivating example.

Example 19: Expected utility in a treatment setting

Imagine a patient discussing treatment options with their doctor. The patient wishes to balance two conflicting objectives: *treatment efficacy* (v_1) and *comfort* (v_2, i.e. minimising side effects). They have expressed that they prefer treatments offering a balanced outcome between these two goals. A simple utility function capturing this preference is the product $u(v_1, v_2) = v_1 \cdot v_2$, which is maximised when both outcomes are high and similar in magnitude.

The doctor proposes two plans. Treatment A alternates between being highly effective but uncomfortable, or comfortable but ineffective:

$$A = \begin{cases} (v_1, v_2) = (1.0, 0.0) & \text{with probability } \frac{1}{2}, \\ (v_1, v_2) = (0.0, 1.0) & \text{with probability } \frac{1}{2}, \end{cases}$$

yielding an expected return $\mathbb{E}[A] = (0.5, 0.5)$. Treatment B, in contrast, is moderately effective and tolerable:

$$B = \begin{cases} (v_1, v_2) = (0.45, 0.45) & \text{with probability } 1, \end{cases}$$

with expected return $\mathbb{E}[B] = (0.45, 0.45)$.

Under the standard Pareto criterion, treatment A dominates treatment B since its expected efficacy and comfort are both higher. Yet, when evaluated through the patient's utility function, A yields an expected utility of 0, while B yields $0.45^2 = 0.2025$. As the treatment is applied only once, the patient aims to maximise expected utility and therefore prefers the more balanced, less risky option B.

As this example shows, it is pertinent to consider exactly what the decision-maker aims to optimise for: do they optimise for repeated execution of the same policy, or the expected utility from one execution? In the former case, they may well decide based on the expected value of the distribution. In the latter case, however, taking the full distribution of returns into account is key to effective decision support.

2.2 Classes of Decision-Makers

To provide meaningful decision support, we must first specify the class of decision-makers we aim to support. As stated in Section 3.1, we assume that each decision-maker can be characterised by a utility function that captures their preferences over outcomes. In principle, one could consider the class of all utility functions $u : \mathbb{R}^d \to \mathbb{R}$. However, such generality is unworkable, as it includes "irrational" decision-makers (those preferring negative outcomes over positive ones) and pathological decision-makers (those with constant utility), rendering the concept of an optimal solution vacuous.

The first class of decision-makers we consider consists of those with strongly increasing utility functions, meaning that an improvement in any single objective, while keeping all others fixed, necessarily leads to a higher utility. This class was also implicitly considered in Chapter 3 when we examined Pareto-optimal policies:

$$\mathcal{U} := \left\{ u : \mathbb{R}^d \to \mathbb{R} \mid \forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d,\ \boldsymbol{x} \succ_{\mathrm{p}} \boldsymbol{y} \implies u(\boldsymbol{x}) > u(\boldsymbol{y}) \right\}. \tag{4.1}$$

In some cases, this class is still too broad, and we restrict our attention to a smaller subset. Such restrictions allow the development of specialised algorithmic methods. In particular, we focus on multivariate risk-averse decision-makers [Richard, 1975]. This class captures decision-makers who prefer balanced outcomes. The defining condition, known as *submodularity*, means that the marginal utility from improving one objective diminishes as the level of another increases. For example, the additional satisfaction from a higher income is smaller when one's health is already excellent, indicating that the objectives act as substitutes rather than complements. We restrict our attention to two-dimensional random variables for this class, which allows us to define the set of multivariate risk-averse utility functions as follows:

$$\hat{\mathcal{U}}^2 := \left\{ u : \mathbb{R}^2 \to \mathbb{R} \mid u \in \mathcal{U} \text{ and } \frac{\partial^2 u(x_1, x_2)}{\partial x_1 \partial x_2} \leq 0 \right\}. \tag{4.2}$$

2.3 First-Order Stochastic Dominance

First-order stochastic dominance (FSD) is a well-known dominance criterion from decision theory and economics, which compares distributions directly [Levy, 2016b;

Denuit et al., 2013]. We introduce it here because our goal is to extend multi-objective decision-making to the distributional setting, where policies are evaluated not just by their expected returns but by the full distribution over outcomes. FSD provides a natural way to order such distributions without collapsing them into single-point estimates, making it an effective foundation for multi-objective distributional decision-making.

Formally, let $F_X(\boldsymbol{x}) = P(\boldsymbol{X} \preceq_{\text{p}} \boldsymbol{x})$ denote the cumulative distribution function (CDF) of a random vector $\boldsymbol{X}$, that is, the probability that $\boldsymbol{X}$ takes on a value Pareto-dominated or equal to $\boldsymbol{x}$. Analogous to Pareto dominance, we apply this criterion to policies $\pi \in \Pi$ in an MOMDP by considering their return distributions, denoted $F_\pi(\boldsymbol{x})$.

Definition 26: First-order stochastic dominance

A policy π first-order stochastically dominates a policy π', denoted $\boldsymbol{Z}^\pi \succeq_{\text{FSD}} \boldsymbol{Z}^{\pi'}$, if and only if

$$F_{\boldsymbol{Z}^\pi}(\boldsymbol{v}) \leq F_{\boldsymbol{Z}^{\pi'}}(\boldsymbol{v}) \quad \text{for all } \boldsymbol{v} \in \mathbb{R}^d.$$

Example 20: First-order stochastic dominance for treatment plans

Continuing the hospital scenario, suppose the doctor presents three alternative treatment plans. Each plan involves a trade-off between efficacy (v_1) and comfort (v_2), and each possible outcome occurs with equal probability.

$$\begin{aligned}
&A\ P\big((v_1, v_2) = (0.85,\ 0.45)\big) = \tfrac{1}{2}, \quad P\big((v_1, v_2) = (0.45,\ 0.85)\big) = \tfrac{1}{2},\\
&B\ P\big((v_1, v_2) = (0.80,\ 0.40)\big) = \tfrac{1}{2}, \quad P\big((v_1, v_2) = (0.40,\ 0.80)\big) = \tfrac{1}{2},\\
&C\ P\big((v_1, v_2) = (0.95,\ 0.35)\big) = \tfrac{1}{2}, \quad P\big((v_1, v_2) = (0.35,\ 0.95)\big) = \tfrac{1}{2}.
\end{aligned}$$

Plan A clearly improves on plan B because every possible outcome in A increases both efficacy and comfort by 0.05 compared with the corresponding outcome in B. Formally, $F_A(t_1, t_2) \leq F_B(t_1, t_2)$ for all thresholds (t_1, t_2), hence $A \succeq_{\text{FSD}} B$. In practice this means that regardless of the patient's exact trade-off between efficacy and comfort, plan B will never be strictly preferred to A and can therefore be discarded.

The comparison between A and C is more nuanced. Plan C yields a higher efficacy in one outcome but lower comfort, while the other outcome reverses this pattern. The two distributions intersect, and neither plan dominates the other.

This example shows how first-order stochastic dominance (FSD) supports decision-making. It can eliminate inferior options, as in the case of B, while retaining those that depend on individual preferences, such as A and C. The doctor should therefore

present both remaining options to the patient, who can then decide based on their own priorities.

A critical note. In the single-objective case, first-order stochastic dominance corresponds to pointwise dominance of cumulative distribution functions, which ensures that all monotonically increasing utility functions prefer the dominant variable [Fishburn, 1974]. In Definition 26, we extended this condition to the multivariate setting by applying it to the joint CDF.

After completing the work on which this chapter is based, we became aware that the standard definition of FSD for random vectors is subtly different [Kopa and Petrová, 2018]. Rather than using the joint CDF, it is defined in terms of *upper sets*: for all upper sets $M \subseteq \mathbb{R}^d$, we require $P(\boldsymbol{Z}^\pi \in M) \leq P(\boldsymbol{Z}^{\pi'} \in M)$. In one dimension, this is equivalent to the CDF condition; in higher dimensions, it is strictly stronger. In subsequent related work, this stronger condition has been considered instead [Cai et al., 2023].

The condition used in Definition 26 corresponds to *weak* FSD in the sense of Kopa and Petrová [2018]. Despite being weaker, it offers several advantages. First, it is far more tractable computationally: joint CDFs are straightforward to compute, whereas verifying dominance over all upper sets is substantially more complex, as it involves checking an entire family of measurable regions. Second, because weak FSD compares distributions at fewer points, it excludes a larger set of dominated policies and thus yields smaller, more manageable solution sets. Third, and crucially for our purposes, weak FSD appears sufficient for multivariate risk-averse decision-makers. This aligns with the learn large, prune later principle we follow: we first learn a broad solution set and subsequently prune it to match specific preferences or risk attitudes without retraining. For the remainder of this chapter, we therefore continue to use Definition 26 and refer to it simply as FSD.

2.4 Limitations of First-Order Stochastic Dominance

Our goal is to enable principled decision support for expected utility maximisers. A natural starting point is first-order stochastic dominance (FSD), as it directly considers the return distributions from which expected utility is computed. Contrary to the univariate case, however, in higher dimensions first-order stochastic dominance does not ensure larger (or even equal) expected utility for all increasing utility functions. Moreover, even strict FSD fails to guarantee a strictly larger expected utility for all multivariate risk-averse utilities.

Proposition 3: Limits of FSD for utility maximisation

Fix $d \geq 2$ and let $\boldsymbol{X}, \boldsymbol{Y}$ be d-dimensional random vectors. Then:

1. $\boldsymbol{X} \succ_{\text{FSD}} \boldsymbol{Y}$ does not imply that $\mathbb{E}[u(\boldsymbol{X})] \geq \mathbb{E}[u(\boldsymbol{Y})]$ for all $u \in \mathcal{U}$.
2. $\boldsymbol{X} \succ_{\text{FSD}} \boldsymbol{Y}$ does not imply that $\mathbb{E}[u(\boldsymbol{X})] > \mathbb{E}[u(\boldsymbol{Y})]$ for all $u \in \hat{\mathcal{U}}^2$.

Proof. Both statements follow by counterexample in $d = 2$ using the following pair of distributions where $\boldsymbol{X} \succ_{\text{FSD}} \boldsymbol{Y}$:

$$\boldsymbol{X} = \left\{ P(2,4) = \frac{2}{3},\ P(4,2) = \frac{1}{3} \right\},$$
$$\boldsymbol{Y} = \left\{ P(2,2) = \frac{1}{3},\ P(2,4) = \frac{1}{3},\ P(4,4) = \frac{1}{3} \right\}.$$

1. Weak FSD.
Define $u(v_1, v_2) = \ln\left(1 + e^{v_1}\right) \ln\left(1 + e^{v_2}\right)$. Then $u \in \mathcal{U}$ and is a smooth, strictly increasing approximation of $\max\{0, v_1\}\max\{0, v_2\}$. A direct calculation yields $\mathbb{E}[u(\boldsymbol{X})] \approx 8.55$ and $\mathbb{E}[u(\boldsymbol{Y})] \approx 9.74$, so despite $\boldsymbol{X} \succ_{\text{FSD}} \boldsymbol{Y}$, expected utility is lower for $\boldsymbol{X}$.

2. Strict FSD under multivariate risk aversion.
Let $u(v_1, v_2) = v_1 + v_2$. This u is increasing and satisfies $\partial_{12} u = 0 \leq 0$, hence $u \in \hat{\mathcal{U}}^2$. For the same $(\boldsymbol{X}, \boldsymbol{Y})$ we have $\mathbb{E}[u(\boldsymbol{X})] = \mathbb{E}[u(\boldsymbol{Y})] = 6$ even though $\boldsymbol{X} \succ_{\text{FSD}} \boldsymbol{Y}$. Thus strict FSD does not imply a strictly greater expected utility for all $u \in \hat{\mathcal{U}}^2$. □

These counterexamples hinge on the joint dependence structure (here, identical marginals with differing dependence), showing that multivariate FSD alone cannot support utility comparisons for all increasing or multivariate risk-averse utilities. Part (i) exhibits a strict reversal of expected utility despite $\boldsymbol{X} \succ_{\text{FSD}} \boldsymbol{Y}$. Part (ii) shows that strict FSD need not yield a strict improvement for every $u \in \hat{\mathcal{U}}^2$, since equality can occur. Whether $\boldsymbol{X} \succ_{\text{FSD}} \boldsymbol{Y}$ guarantees $\mathbb{E}[u(\boldsymbol{X})] \geq \mathbb{E}[u(\boldsymbol{Y})]$ for all $u \in \hat{\mathcal{U}}^2$ remains open.

Finally, we demonstrate that strict FSD between two distributions does not imply strict FSD on any of their marginals. This is significant, as it prevents us from asserting that any single objective has a distribution that is always preferred, and it rules out using the marginal distributions to assess the potential for FSD.

Proposition 4: FSD does not imply marginal FSD

Let $\boldsymbol{X}$ and $\boldsymbol{Y}$ be d-dimensional random vectors. Then,

$$\boldsymbol{X} \succ_{\text{FSD}} \boldsymbol{Y} \implies \exists i \in [d] : X_i \succ_{\text{FSD}} Y_i.$$

Proof. Consider the two-dimensional distributions:

$$\boldsymbol{X} = \left\{ P(2,4) = \frac{2}{3},\ P(4,2) = \frac{1}{3} \right\},$$
$$\boldsymbol{Y} = \left\{ P(2,2) = \frac{1}{3},\ P(2,4) = \frac{1}{3},\ P(4,4) = \frac{1}{3} \right\}.$$

It is straightforward to verify that $\boldsymbol{X} \succ_{\text{FSD}} \boldsymbol{Y}$. However, the marginal CDFs satisfy

$$F_{X_1}(2) = F_{Y_1}(2) = \frac{2}{3}, \quad F_{X_1}(4) = F_{Y_1}(4) = 1,$$

$$F_{X_2}(2) = F_{Y_2}(2) = \frac{1}{3}, \quad F_{X_2}(4) = F_{Y_2}(4) = 1,$$

so $F_{X_1} = F_{Y_1}$ and $F_{X_2} = F_{Y_2}$, implying $\nexists i \in [d]$ such that $X_i \succ_{\text{FSD}} Y_i$. □

3 Distributional Dominance

To address the limitations of Pareto dominance, we introduce the *distributional dominance* criterion. This criterion states that a distribution dominates another when it is first-order stochastic dominant and at least one of the marginal distributions *strictly* first-order stochastic dominates the related marginal distribution of the second distribution.

Definition 27: Distributional dominance

A policy π distributional dominates a policy π', denoted $\boldsymbol{Z}^{\pi} \succ_{\text{d}} \boldsymbol{Z}^{\pi'}$, if and only if

$$\boldsymbol{Z}^{\pi} \succeq_{\text{FSD}} \boldsymbol{Z}^{\pi'} \wedge \exists i \in [d] : Z_i^{\pi} \succ_{\text{FSD}} Z_i^{\pi'}.$$

In general, distributional dominance is a stronger condition than strict first-order stochastic dominance as the condition on the marginal distributions implies strict FSD but is not implied by it (see Proposition 4). Defining dominance in this way ensures a strictly greater expected utility for a broad class of decision-makers and leads to the general solution set introduced in Section 4.

Examining the marginals. We now relate first-order stochastic dominance on the joint distribution to the same restriction on all of the marginal distributions. Since Definition 27 includes a strict condition on at least one marginal distribution, we introduce this lemma to lay the groundwork for the proof of Theorem 7 which demonstrates that expected utility maximisation leads to distributional dominance.

Lemma 4: FSD implies FSD on marginals

Let $\boldsymbol{X}$ and $\boldsymbol{Y}$ be d-dimensional random vectors. Then,

$$\boldsymbol{X} \succeq_{\text{FSD}} \boldsymbol{Y} \implies \forall i \in [d] : X_i \succeq_{\text{FSD}} Y_i.$$

Proof. First, note that for any random vector $\boldsymbol{X}$,

$$F_{X_i}(x_i) = \lim_{\boldsymbol{x}_{-i} \to \infty} F_{\boldsymbol{X}_{-i}, X_i}(\boldsymbol{x}_{-i}, x_i).$$

Thus, when $\boldsymbol{X} \succeq_{\text{FSD}} \boldsymbol{Y}$, then $\forall \boldsymbol{v} \in \mathbb{R}^d$:

$$\begin{aligned}
&\implies F_{\boldsymbol{X}}(\boldsymbol{v}) \leq F_{\boldsymbol{Y}}(\boldsymbol{v}) \\
&\implies \lim_{\boldsymbol{v}_{-i} \to \infty} F_{\boldsymbol{X}_{-i}, X_i}(\boldsymbol{v}_{-i}, v_i) \leq \lim_{\boldsymbol{v}_{-i} \to \infty} F_{\boldsymbol{Y}_{-i}, Y_i}(\boldsymbol{v}_{-i}, v_i) \\
&\implies \forall i \in [d] : F_{X_i}(v_i) \leq F_{Y_i}(v_i) \\
&\implies \forall i \in [d] : X_i \succeq_{\text{FSD}} Y_i.
\end{aligned}$$

□

Interestingly, in the special case of random vectors with independent coordinates, Lemma 4 can be used to show that distributional dominance coincides with strict FSD. In practice, though, objectives are rarely independent and are typically in conflict.

Proposition 5: Strict FSD for independent coordinates

Let $\boldsymbol{X}$ and $\boldsymbol{Y}$ be d-dimensional random vectors. If the coordinates of $\boldsymbol{X}$ and of $\boldsymbol{Y}$ are mutually independent, then

$$\boldsymbol{X} \succ_{\text{d}} \boldsymbol{Y} \iff \boldsymbol{X} \succ_{\text{FSD}} \boldsymbol{Y}.$$

Proof. ($\Rightarrow$) Suppose by Lemma 4 $X_i \succeq_{\text{FSD}} Y_i$ for all i and $X_j \succ_{\text{FSD}} Y_j$ for some j. By independence,

$$F_{\boldsymbol{X}}(\boldsymbol{v}) = \prod_{i=1}^{d} F_{X_i}(v_i) \leq \prod_{i=1}^{d} F_{Y_i}(v_i) = F_{\boldsymbol{Y}}(\boldsymbol{v}) \quad \text{for all } \boldsymbol{v}.$$

Let t satisfy $F_{X_j}(t) < F_{Y_j}(t)$. Choose any finite values v_k for $k \neq j$ and set $v_j = t$. Then

$$\begin{aligned}
F_{\boldsymbol{Y}}(\boldsymbol{v}) - F_{\boldsymbol{X}}(\boldsymbol{v}) &= \Big(\prod_{k \neq j} F_{Y_k}(v_k)\Big) F_{Y_j}(t) - \Big(\prod_{k \neq j} F_{X_k}(v_k)\Big) F_{X_j}(t) \\
&\geq \Big(\prod_{k \neq j} F_{X_k}(v_k)\Big) \big(F_{Y_j}(t) - F_{X_j}(t)\big) \; > \; 0,
\end{aligned}$$

since $F_{Y_k} \geq F_{X_k}$ for all k. Thus there exists a $\boldsymbol{v}$ with $F_{\boldsymbol{X}}(\boldsymbol{v}) < F_{\boldsymbol{Y}}(\boldsymbol{v})$, and hence $\boldsymbol{X} \succ_{\text{FSD}} \boldsymbol{Y}$.

($\Leftarrow$) Assume $\boldsymbol{X} \succ_{\text{FSD}} \boldsymbol{Y}$. Then $F_{\boldsymbol{X}}(\boldsymbol{v}) \leq F_{\boldsymbol{Y}}(\boldsymbol{v})$ for all $\boldsymbol{v}$, with strict inequality at some $\boldsymbol{v}^*$. For each i and $t \in \mathbb{R}$,

$$F_{X_i}(t) = \lim_{v_{-i} \to +\infty} F_{\boldsymbol{X}}(v_{-i}, t) \leq \lim_{v_{-i} \to +\infty} F_{\boldsymbol{Y}}(v_{-i}, t) = F_{Y_i}(t),$$

so $X_i \succeq_{\text{FSD}} Y_i$ for all i. Since the coordinates are independent,

$$F_{\boldsymbol{X}}(\boldsymbol{v}^*) = \prod_{i=1}^{d} F_{X_i}(v_i^*), \qquad F_{\boldsymbol{Y}}(\boldsymbol{v}^*) = \prod_{i=1}^{d} F_{Y_i}(v_i^*).$$

The strict product inequality implies there exists j with $F_{X_j}(v_j^*) < F_{Y_j}(v_j^*)$, hence $X_j \succ_{\text{FSD}} Y_j$. □

Relating expected utility and FSD. Since our aim is to assist decision-makers by identifying a set of policies with non-dominated return distributions, it is essential to clarify how expected utility comparisons relate to distributional dominance. In particular, we establish that if one random vector results in greater expected utility than another for every strictly increasing utility function, then it first-order stochastically dominates the alternative. The argument proceeds via an auxiliary result, Lemma 5, which generalises an earlier theorem of Fishburn [1974].

Lemma 5: Expected utility implies FSD

Let $\boldsymbol{X}$ and $\boldsymbol{Y}$ be d-dimensional random vectors. Then,

$$\forall u \in \mathcal{U}^{\uparrow} : \mathbb{E}[u(\boldsymbol{X})] \geq \mathbb{E}[u(\boldsymbol{Y})] \implies \boldsymbol{X} \succeq_{\text{FSD}} \boldsymbol{Y}.$$

Proof. Let $\boldsymbol{X}$ and $\boldsymbol{Y}$ be distributions such that $\mathbb{E}[u(\boldsymbol{X})] \geq \mathbb{E}[u(\boldsymbol{Y})]$ for all $u \in \mathcal{U}^{\uparrow}$. We will show that this implies $\boldsymbol{X} \succeq_{\text{FSD}} \boldsymbol{Y}$.

1. Violation of FSD.
Assume $\boldsymbol{X}$ does not first-order stochastically dominate $\boldsymbol{Y}$. Then there exists $\boldsymbol{v} \in \mathbb{R}^d$ such that

$$\Delta := F_{\boldsymbol{X}}(\boldsymbol{v}) - F_{\boldsymbol{Y}}(\boldsymbol{v}) > 0.$$

2. Choose the perturbation tolerance.
Set

$$\eta = \frac{\Delta}{2\left|\mathbb{E}[s(\boldsymbol{X})] - \mathbb{E}[s(\boldsymbol{Y})]\right| + 1} > 0,$$

where $s(\boldsymbol{z}) = \sum_{i=1}^{d} z_i$.

3. Define a strictly increasing utility.
Let

$$h(\boldsymbol{z}) = \mathbf{1}\{\boldsymbol{z} \not\preceq_{\text{p}} \boldsymbol{v}\}, \qquad u_\eta(\boldsymbol{z}) = h(\boldsymbol{z}) + \eta\, s(\boldsymbol{z}).$$

For any $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$ with $\boldsymbol{x} \succ_{\mathrm{p}} \boldsymbol{y}$, either $h(\boldsymbol{x}) > h(\boldsymbol{y})$ or h ties but $s(\boldsymbol{x}) > s(\boldsymbol{y})$. Hence $u_\eta(\boldsymbol{x}) > u_\eta(\boldsymbol{y})$, so $u_\eta \in \mathcal{U}$.

4. Evaluate the expectation gap.
Using $\mathbb{E}[h(\boldsymbol{X})] = 1 - F_{\boldsymbol{X}}(\boldsymbol{v})$ and $\mathbb{E}[h(\boldsymbol{Y})] = 1 - F_{\boldsymbol{Y}}(\boldsymbol{v})$,

$$\begin{aligned}\mathbb{E}\big[u_\eta(\boldsymbol{X})\big] - \mathbb{E}\big[u_\eta(\boldsymbol{Y})\big] &= \big[F_{\boldsymbol{Y}}(\boldsymbol{v}) - F_{\boldsymbol{X}}(\boldsymbol{v})\big] + \eta\left(\mathbb{E}[s(\boldsymbol{X})] - \mathbb{E}[s(\boldsymbol{Y})]\right) \\ &= -\Delta + \eta\left(\mathbb{E}[s(\boldsymbol{X})] - \mathbb{E}[s(\boldsymbol{Y})]\right).\end{aligned}$$

By the choice of η, the second term has magnitude less than $\frac{\Delta}{2}$, so the entire expression is less than $-\frac{\Delta}{2} < 0$.

5. Contradiction.
The negative expectation difference contradicts the premise that every $u \in \mathcal{U}$ satisfies $\mathbb{E}[u(\boldsymbol{X})] \geq \mathbb{E}[u(\boldsymbol{Y})]$. Therefore, our assumption is false and $\boldsymbol{X} \succeq_{\mathrm{FSD}} \boldsymbol{Y}$. □

Connecting to distributional dominance. Given Definition 27 and Lemmas 4 and 5, we can immediately show that when a given random vector has *strictly* greater expected utility for all utility functions in $\mathcal{U}$ than a second random vector, this implies distributional dominance. In other words, distributional dominance is a necessary condition for a random vector to yield strictly greater expected utility for all increasing utility functions.

Theorem 7: Increasing utility implies distributional dominance

Let $\boldsymbol{X}$ and $\boldsymbol{Y}$ be d-dimensional random vectors. Then,

$$\forall u \in \mathcal{U} : \mathbb{E}\left[u(\boldsymbol{X})\right] > \mathbb{E}\left[u(\boldsymbol{Y})\right] \implies \boldsymbol{X} \succ_{\mathrm{d}} \boldsymbol{Y}.$$

Proof. **1. Multivariate ⇒ marginal weak FSD.**
By Lemma 5, the premise $\mathbb{E}\left[u(\boldsymbol{X})\right] > \mathbb{E}\left[u(\boldsymbol{Y})\right]$ for every $u \in \mathcal{U}$ implies $\boldsymbol{X} \succeq_{\mathrm{FSD}} \boldsymbol{Y}$. Lemma 4 then yields

$$\forall i \in [d] : \qquad X_i \succeq_{\mathrm{FSD}} Y_i. \tag{1}$$

2. A contradiction if *all* marginals were equal.
Assume for contradiction that every marginal distribution coincides:

$$F_{X_i}(t) = F_{Y_i}(t) \quad \forall t,\ \forall i. \tag{2}$$

Choose any strictly increasing univariate function ϕ, and define the separable utility $u(\boldsymbol{x}) = \sum_{i=1}^{d} \phi(x_i)$. Because $\partial_{x_i} u > 0$ for all i, we have $u \in \mathcal{U}$. Linearity of expectation

together with (2) gives

$$\mathbb{E}\left[u(\boldsymbol{X})\right] = \sum_{i=1}^{d} \mathbb{E}\left[\phi(X_i)\right] = \sum_{i=1}^{d} \mathbb{E}\left[\phi(Y_i)\right] = \mathbb{E}\left[u(\boldsymbol{Y})\right],$$

contradicting the strict inequality assumed for every $u \in \mathcal{U}$.

3. Existence of a strictly dominant marginal.
Hence (2) is false; there exists an index j with $F_{X_j} \neq F_{Y_j}$. Together with the weak inequality in (1) this gives $X_j \succ_{\text{FSD}} Y_j$ strictly. We now have $\boldsymbol{X} \succeq_{\text{FSD}} \boldsymbol{Y}$ and $\exists j : X_j \succ_{\text{FSD}} Y_j$, which is exactly the definition of distributional dominance. □

Using distributional dominance. In practice, verifying that one random vector yields strictly greater expected utility than another for all utility functions is infeasible. In contrast, distributional dominance can be checked efficiently (see Section 4.2), making it desirable for Theorem 7 to hold as an equivalence. However, similar to Proposition 3, counterexamples exist in the general case. To recover a partial converse, we restrict attention to multivariate risk-averse decision-makers, that is, decision-makers with a utility function $u \in \hat{\mathcal{U}}^2$.

Theorem 8: Distributional dominance implies increasing utility

Let $\boldsymbol{X}$ and $\boldsymbol{Y}$ be two-dimensional random vectors. Then $\forall u \in \hat{\mathcal{U}}^2$,

$$\boldsymbol{X} \succ_{\text{d}} \boldsymbol{Y} \implies \mathbb{E}\left[u(\boldsymbol{X})\right] > \mathbb{E}\left[u(\boldsymbol{Y})\right].$$

Proof. By definition, $\boldsymbol{X} \succ_{\text{d}} \boldsymbol{Y} \implies \boldsymbol{X} \succeq_{\text{FSD}} \boldsymbol{Y}$. For this condition, Hayes et al. [2022c] show that,

$$\mathbb{E}\left[u(\boldsymbol{X})\right] - \mathbb{E}\left[u(\boldsymbol{Y})\right] \geq -\int_{-\infty}^{+\infty} \lim_{t\to+\infty} \frac{\partial u(t,z)}{\partial z} \Delta_F(t,z)\mathrm{d}z,$$

where $\Delta_F(t,z) = F_{\boldsymbol{X}}(t,z) - F_{\boldsymbol{Y}}(t,z)$. Without loss of generality, let us assume that $X_2 \succ_{\text{FSD}} Y_2$, i.e. $\exists z \in \mathbb{R} : F_{X_2}(z) < F_{Y_2}(z)$ and write $\Delta_{F_2}(z) := F_{X_2}(z) - F_{Y_2}(z)$. Then,

$$\begin{aligned}
&\mathbb{E}\left[u(\boldsymbol{X})\right] - \mathbb{E}\left[u(\boldsymbol{Y})\right] \\
&\geq -\int_{-\infty}^{+\infty} \lim_{t\to+\infty} \frac{\partial u(t,z)}{\partial z} \Delta_F(t,z)\mathrm{d}z \\
&= -\int_{-\infty}^{+\infty} \left(\lim_{t\to+\infty} \frac{\partial u(t,z)}{\partial z}\right) \Delta_{F_2}(z)\mathrm{d}z \\
&> 0.
\end{aligned}$$

□

4 A General Solution Set

We adopt distributional dominance to define the *distributional undominated set* (DUS). The DUS has two key desiderata: it contains the Pareto front, i.e. the optimal set under scalarised expected returns (SER), and it includes all optimal policies for multivariate risk-averse decision-makers under expected scalarised returns (ESR). This makes it particularly useful when the type of decision-maker is not known in advance, supporting a learn large, prune later strategy in which a broad set is learned first and later pruned into smaller subsets tailored to specific preferences or risk attitudes without retraining.

4.1 Distributional Undominated Set

As the name suggests, the distributional undominated set contains only those policies which are not pairwise distributional dominated. We define this formally in Definition 28.

Definition 28: Distributional undominated set

The distributional undominated set is the set of all policies that are not distributional dominated:

$$\text{DUS} = \left\{ \pi \in \Pi \mid \nexists \pi' \in \Pi, Z^{\pi'} \succ_{\text{d}} Z^{\pi} \right\}. \tag{4.3}$$

From this definition it is clear that all policies which are optimal for multivariate risk-averse decision-makers are in the set. To show that the Pareto front is a subset as well, we first introduce Lemma 6, stating that distributional dominance implies Pareto dominance.

Lemma 6: Distributional dominance implies Pareto dominance

For all policies $\pi, \pi' \in \Pi$: $Z^{\pi} \succ_{\text{d}} Z^{\pi'} \implies \boldsymbol{v}^{\pi} \succ_{\text{p}} \boldsymbol{v}^{\pi'}$.

Proof. **1. From distributional to marginal FSD.**
Distributional dominance means $Z^{\pi} \succeq_{\text{FSD}} Z^{\pi'}$ and $\exists j \in [d]$ with $Z_j^{\pi} \succ_{\text{FSD}} Z_j^{\pi'}$. By Lemma 4, this means that for every coordinate $i \in [d]$,

$$F_{Z_i^{\pi}}(v) \leq F_{Z_i^{\pi'}}(v) \quad \forall v \in \mathbb{R}.$$

2. Expectation of a real random variable via its CDF.
For any real-valued integrable random variable W with CDF F_W,

$$\mathbb{E}[W] = \underbrace{\int_0^{\infty} \left[1 - F_W(v)\right] \mathrm{d}v}_{\text{positive tail}} - \underbrace{\int_{-\infty}^{0} F_W(v)\, \mathrm{d}v}_{\text{negative tail}}. \tag{1}$$

3. Weak Pareto dominance from marginal FSD.
Fix i. Insert $F_{Z_i^\pi}$ and $F_{Z_i^{\pi'}}$ into (1) and subtract:

$$\mathbb{E}\left[Z_i^\pi\right] - \mathbb{E}\left[Z_i^{\pi'}\right] = \int_0^\infty \left[F_{Z_i^{\pi'}}(v) - F_{Z_i^\pi}(v)\right] \mathrm{d}v \; + \; \int_{-\infty}^0 \left[F_{Z_i^{\pi'}}(v) - F_{Z_i^\pi}(v)\right] \mathrm{d}v. \tag{2}$$

Both integrands are ≥ 0 by Step 1, so each integral is ≥ 0; hence $\mathbb{E}\left[Z_i^\pi\right] \geq \mathbb{E}\left[Z_i^{\pi'}\right]$.

4. Strictness and conclusion.
For the special index j with strict FSD, there is a v for which $F_{Z_j^\pi}(v) < F_{Z_j^{\pi'}}(v)$. In (2) this makes at least one integral strictly positive, giving $\mathbb{E}\left[Z_j^\pi\right] > \mathbb{E}\left[Z_j^{\pi'}\right]$, while for all i the difference is ≥ 0. Therefore $\boldsymbol{v}^\pi \succ_{\mathrm{p}} \boldsymbol{v}^{\pi'}$. □

Leveraging Lemma 6, it is a corollary that the Pareto front is a subset of the DUS. We highlight that our dominance results and solution sets are not restricted to MOMDPs but apply to any stochastic multi-objective decision problem with vector-valued outcomes.

Corollary 2: Pareto front is a subset of the DUS

For any family of policies Π, the Pareto front is a subset of the distributional undominated set, i.e.,

$$\mathcal{F} \subseteq \text{DUS}.$$

Proof. Assume there exists a $\pi \in \mathcal{F}$ such that $\pi \notin \text{DUS}$. As $\pi \notin \text{DUS}$, we know that,

$$\exists \pi' \in \Pi, \boldsymbol{Z}^{\pi'} \succ_{\mathrm{d}} \boldsymbol{Z}^{\pi}.$$

Lemma 6 implies then that $\boldsymbol{v}^{\pi'} \succ_{\mathrm{p}} \boldsymbol{v}^{\pi}$. As π is Pareto dominated by π', $\pi \notin \mathcal{F}$, leading to a contradiction. □

4.2 Computing the DUS

To deal with return distributions computationally, we project distributions to multivariate categorical distributions [Bellemare et al., 2023; Hayes et al., 2022c]. This ensures that finite memory is used, and, importantly, that computations can be performed efficiently. Concretely, to verify first-order stochastic dominance, we need only compare a finite number of points as the CDF is a multivariate step function with steps at $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n$. Formally, for the categorical distribution $\boldsymbol{X}$ the cumulative distribution at $\boldsymbol{x}$ is computed as follows,

$$F_X(\boldsymbol{x}) = \sum_{\boldsymbol{v}_i \preceq_{\mathrm{p}} \boldsymbol{x}} p(\boldsymbol{v}_i). \tag{4.4}$$

Additionally, discrete distributions enable straightforward computation of marginal distributions, thus having all ingredients to check distributional dominance (see Definition 27). Then, starting from a given set of policies, the DUS can be computed using a modified version of the Pareto Prune (PPRUNE) algorithm [Roijers and Whiteson, 2017] that checks for distributional dominance rather than Pareto dominance. We refer to the resulting pruning algorithm as DPRUNE.

5 A Solution Set for Expected Utility Maximisers

Because the DUS is a superset of the Pareto front and also contains all policies that are optimal for multivariate risk-averse expected utility maximisers, it can become prohibitively large for practical decision support applications. When considering SER, DPRUNE reduces the set to the Pareto front. Here, we propose an analogous pruning method for expected utility maximisers and introduce the corresponding solution concept.

5.1 Convex Mixture of Distributions

To obtain a more compact solution set for expected utility maximisers, we now extend distributional dominance to the case where both sides of the comparison are *convex mixtures of distributions*. Intuitively, this allows us to capture situations where no individual distribution dominates another, yet combinations of distributions yield stochastically superior outcomes. This extension provides a principled way to further reduce the DUS, aligning it with the preferences of risk-averse decision-makers without discarding optimal policies under ESR.

The idea of dominance through convex mixtures has been studied in the univariate case, where it was shown that a mixture distribution can be constructed that first-order stochastically dominates another distribution if and only if, for any decision-maker, there exists at least one component distribution in the mixture that is preferred to the dominated one [Fishburn, 1974; Bawa et al., 1985]. Extensions to multivariate distributions have also been explored [Denuit et al., 2013]. Building on this foundation, we show that convex distributional dominance implies greater expected utility for multivariate risk-averse decision-makers when considering bivariate distributions.

Theorem 9: Mixture dominance improves at least one component

Let $\{\boldsymbol{X}_1, \dots, \boldsymbol{X}_n\}$ and $\{\boldsymbol{Y}_1, \dots, \boldsymbol{Y}_n\}$ be collections of *two-dimensional* random vectors, each component having a finite expectation. Fix weights $\lambda = (\lambda_1, \dots, \lambda_n) \in \Delta^n$ and form the *mixtures*

$$\boldsymbol{X}^{(\lambda)} := \boldsymbol{X}_I, \qquad \boldsymbol{Y}^{(\lambda)} := \boldsymbol{Y}_I, \quad \text{where } \Pr[I = i] = \lambda_i.$$

If

$$\boldsymbol{X}^{(\lambda)} \succ_{\mathrm{d}} \boldsymbol{Y}^{(\lambda)},$$

then for all $u \in \hat{\mathcal{U}}^2$,

$$\exists\, i \in [n] : \qquad \mathbb{E}\big[u(\boldsymbol{X}_i)\big] > \mathbb{E}\big[u(\boldsymbol{Y}_i)\big].$$

Proof. **1. Expected utility gap for the mixtures.**
Distributional dominance of the mixtures and Theorem 8 give

$$\mathbb{E}\big[u(\boldsymbol{X}^{(\lambda)})\big] > \mathbb{E}\big[u(\boldsymbol{Y}^{(\lambda)})\big] \qquad \forall u \in \hat{\mathcal{U}}^2. \tag{1}$$

2. Law of total expectation.
By construction of the mixtures,

$$\mathbb{E}\big[u(\boldsymbol{X}^{(\lambda)})\big] = \sum_{i=1}^{n} \lambda_i\, \mathbb{E}\big[u(\boldsymbol{X}_i)\big], \qquad \mathbb{E}\big[u(\boldsymbol{Y}^{(\lambda)})\big] = \sum_{i=1}^{n} \lambda_i\, \mathbb{E}\big[u(\boldsymbol{Y}_i)\big]. \tag{2}$$

3. Weighted inequality.
Insert (2) into (1):

$$\sum_{i=1}^{n} \lambda_i\, \mathbb{E}\big[u(\boldsymbol{X}_i)\big] > \sum_{i=1}^{n} \lambda_i\, \mathbb{E}\big[u(\boldsymbol{Y}_i)\big]. \tag{3}$$

4. Contradiction argument.
Assume, for contradiction, that $\mathbb{E}[u(\boldsymbol{X}_i)] \leq \mathbb{E}[u(\boldsymbol{Y}_i)]$ for every i. Multiplying each inequality by $\lambda_i \geq 0$ and summing contradicts (3). Hence

$$\exists\, i \in [n] : \qquad \mathbb{E}\big[u(\boldsymbol{X}_i)\big] > \mathbb{E}\big[u(\boldsymbol{Y}_i)\big]. \qquad \square$$

Notice that this result may be used to further prune the set of relevant return distributions. In particular, if we select a single $\boldsymbol{Y}$ and construct a convex mixture of distributions $\boldsymbol{X}_i$ such that $\boldsymbol{X}^{(\lambda)} \succ_{\mathrm{d}} \boldsymbol{Y}$, then for every multivariate risk-averse decision-maker there exists a distribution in the mixture that is preferred over $\boldsymbol{Y}$. We may therefore remove $\boldsymbol{Y}$ from the set of distributions, since no decision-maker will ever prefer it.

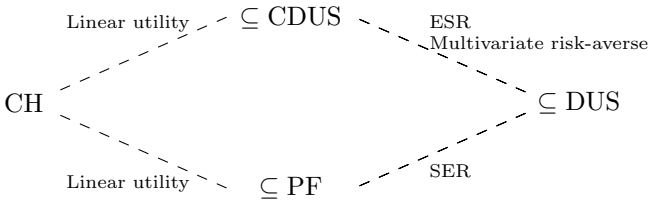


Figure 4.1: A taxonomy of solution sets in multi-objective decision-making.

5.2 Convex Distributional Undominated Set

We define a final solution set, called the convex distributional undominated set (CDUS), that contains only those policies which are undominated by a mixture of distributions. We define the CDUS formally below.

Definition 29: Convex distributional undominated set

The CDUS is the set of all policies that are not distributional dominated by a convex mixture:

$$\text{CDUS} = \left\{ \pi \in \Pi \mid \nexists \lambda \in \Delta^{|\Pi|} : \sum_{i=1}^{|\Pi|} \lambda_i Z^{\pi_i} \succ_{\text{d}} Z^{\pi} \right\}.$$

Given the myriad of solution sets in multi-objective decision-making, it is useful to define a complete taxonomy between them. From Corollary 2, we know that the Pareto front is a subset of the DUS. Additionally, it follows from Definition 29 that the CDUS is also a subset of the DUS. Earlier work has shown that the convex hull is a subset of the Pareto front [Roijers and Whiteson, 2017] and we show that this is also true for the CDUS.

Corollary 3: Convex hull is a subset of the CDUS

For any family of policies Π,

$$\text{CH}(\Pi) \subseteq \text{CDUS}.$$

Proof. The proof follows the same structure as Corollary 2. □

Completing the taxonomy. The final missing piece concerns the relation between the CDUS and the Pareto front. However, in Proposition 6 we present counterexamples for any subset relation between them. The landscape of solution sets for multi-objective decision-making can then be summarised as shown in Fig. 4.1.

Proposition 6: No subset relation between $\mathcal{F}$ and CDUS

For any family of policies Π, in general neither

$$\mathcal{F} \subseteq \text{CDUS} \quad \text{nor} \quad \text{CDUS} \subseteq \mathcal{F}.$$

Proof. Consider four policies with the following return distributions:

$$\begin{aligned} \boldsymbol{Z}^{\pi_1} &= \{P(5,1) = 1\}, \\ \boldsymbol{Z}^{\pi_2} &= \{P(1,5) = 1\}, \\ \boldsymbol{Z}^{\pi_3} &= \{P(1,3) = \tfrac{1}{2},\ P(3,1) = \tfrac{1}{2}\}, \\ \boldsymbol{Z}^{\pi_4} &= \{P(1,10) = 0.49,\ P(1,0) = 0.51\}. \end{aligned}$$

Their value vectors are

$$\boldsymbol{v}^{\pi_1} = (5,1), \quad \boldsymbol{v}^{\pi_2} = (1,5), \quad \boldsymbol{v}^{\pi_3} = (2,2), \quad \boldsymbol{v}^{\pi_4} = (1,4.9).$$

1. $\pi_3 \in \mathcal{F} \setminus$ **CDUS.**
Relative to $\{\pi_1, \pi_2, \pi_3, \pi_4\}$, $\boldsymbol{v}^{\pi_3}$ is not Pareto dominated by any other value vector, so $\pi_3 \in \mathcal{F}$. Moreover,

$$\tfrac{1}{2}\boldsymbol{Z}^{\pi_1} + \tfrac{1}{2}\boldsymbol{Z}^{\pi_2} \succ_{\text{d}} \boldsymbol{Z}^{\pi_3},$$

so $\pi_3 \notin$ CDUS.

2. $\pi_4 \in$ **CDUS** $\setminus \mathcal{F}$**.**
We have $\boldsymbol{v}^{\pi_2} = (1,5) \succ_{\text{p}} (1,4.9) = \boldsymbol{v}^{\pi_4}$, hence $\pi_4 \notin \mathcal{F}$. For distributional dominance, note that any convex mixture of $\boldsymbol{Z}^{\pi_1}, \boldsymbol{Z}^{\pi_2}, \boldsymbol{Z}^{\pi_3}$ has second-coordinate support contained in $(-\infty, 5]$, whereas $\boldsymbol{Z}^{\pi_4}$ places positive mass at 10. Therefore, for the second marginal CDFs, at $v = 7$ the mixture satisfies $F_{\text{mix},2}(7) = 1$ while $F_{Z^{\pi_4},2}(7) < 1$, so the marginal FSD condition fails and no such mixture distributional dominates $\boldsymbol{Z}^{\pi_4}$. Hence $\pi_4 \in$ CDUS. □

5.3 Pruning to the CDUS

To prune a set of distributions to its CDUS, we must check for each distribution whether it is dominated by a mixture of the other distributions. Fortunately, this verification is feasible by restating the problem using linear programming. Concretely, we extend an algorithm that checks whether a univariate distribution is convex first-order stochastic dominated to our setting [Bawa et al., 1985]. We show the resulting linear program in Algorithm 7 and refer to the overall pruning process as CDPrune.

For notational simplicity, we define the size of the set of distributions allowed in the mixture as n. Then the linear program takes in total $n + 1$ distributions as input, where

Algorithm 7: The dominance check for CDPrune.

Input: A set of return distributions $\mathcal{Z}$ allowed in the mixture, and a return distribution Z to check

Output: Whether Z is convex dominated

1 Solve the following optimisation problem:

$$\text{Maximise } \delta = \sum_{j=1}^{h} \sum_{k=1}^{d} l_{j,k} \tag{4.5}$$

Subject to:

$$\sum_{i=1}^{n} \lambda_i F_{Z_i}(\boldsymbol{v}_j) + s_j = F_Z(\boldsymbol{v}_j) \quad j = 1, \dots, h \tag{4.6}$$

$$\sum_{i=1}^{n} \lambda_i F_{Z_{i,k}}(v_{j,k}) + l_{j,k} = F_{Z_k}(v_{j,k})$$
$$j = 1, \dots, h \quad k = 1, \dots, d \tag{4.7}$$

$$\sum_{i=1}^{n} \lambda_i = 1$$
$$\lambda_i \geq 0 \quad i = 1, \dots, n$$
$$s_j \geq 0 \quad j = 1, \dots, h \quad \text{where } s_j \text{ is a slack variable} \tag{4.8}$$

return *True if $\delta > 0$, else False*

the final distribution is the distribution to check. As these distributions are discrete, the CDFs are multivariate step functions that step at a finite number of points. Let D_i be the set of points at which the CDF of distribution i steps. Then $D = \bigcup_{i=1}^{n+1} D_i$ is the union of all such points. We denote $h = |D|$.

The linear program maximises δ, which is the sum of slack variables that make up the difference between the CDFs of the marginal mixture distributions and the marginals of the distribution to check (Eq. (4.5)). If this procedure leads to a δ greater than zero, this implies that the conditions for distributional dominance are met and the distribution is dominated by the mixture. Note that we may omit an additional constraint on the l slack variables to be greater than or equal to zero, as this is implied by the constraint on the s slack variables (Eq. (4.8)).

Approximating dominance without joint CDFs. When no exact formulation of the joint CDFs is available, we propose an alternative linear program that operates solely on the marginal distributions. In this case, it is necessary to change the first constraint in

Eq. (4.6) to

$$\sum_{i=1}^{n} \lambda_i \prod_{k=1}^{d} F_{\mathcal{Z}_{i,k}}(v_{j,k}) + s_j = \prod_{k=1}^{d} F_{Z_k}(v_{j,k}), \tag{4.9}$$

while the second constraint in Eq. (4.7) is removed altogether. By maximising the sum of s slack variables, the resulting linear program essentially checks for strict first-order stochastic dominance between random vectors with independent variables. As shown in Proposition 5, this is equivalent to distributional dominance for independent variables and otherwise may serve as an approximation.

Computational efficiency. The linear program in Algorithm 7 scales poorly with both the number of input distributions and the dimensionality of their categorical projections. It also requires access to the joint CDF and the corresponding marginal CDFs. In practice, efficiency can be improved by evaluating marginal CDFs lazily and caching results for reuse. Further speed-ups may be obtained by exploiting structure in the order of dominance checks, for instance by bootstrapping the solution of new linear programs from previously solved, similar ones [Roijers et al., 2018]. Finally, since all dominance checks are independent, they can be executed in parallel.

6 Distributional Multi-Objective Q-Learning

We conclude by bridging our theoretical results to an algorithmic design that learns a DUS in a tabular MOMDP. Pareto Q-learning (PQL) is a classical method for approximating the Pareto front in multi-objective reinforcement learning [Van Moffaert and Nowé, 2014]. Its framework extends naturally to the distributional setting, which we leverage to learn the DUS. Our resulting algorithm, *Distributional Multi-Objective Q-learning* (DIMOQ), is presented in Algorithm 8.

6.1 Overview

Training proceeds via repeated interaction under an ε-greedy policy. The algorithm learns (i) the immediate reward distributions $\boldsymbol{r}(s, a, s')$ from empirical samples and (ii) the non-dominated future-return set $ND(s, a, s')$ using a distributional variant of the PQL update (see Eq. (4.10)). In this update, dominance pruning in the successor state employs DPrune, the distributional analogue of PPrune, to remove dominated distributions. The full procedure is presented in Algorithm 8. Importantly, DIMOQ never explicitly tracks Q-values, instead only composing them whenever they are needed for the update or action selection. After training, the DUS is obtained by pruning the Q-set at the initial state for each action.

Algorithm 8: DIMOQ algorithm.

Input: State space s, action space $\mathcal{A}$, discount factor γ
Output: The DUS

Initialise all $\boldsymbol{Q}(s, a)$ as empty sets
Initialise all $\boldsymbol{r}(s, a, s')$ as Dirac delta distributions
Estimate p from random walks
for *each episode* **do**
    Initialise state s
    repeat
        Take an action $a \sim \pi(a \mid s)$
        Observe next state $s' \in \mathcal{S}$ and reward $\boldsymbol{r} \in \mathbb{R}^d$
        $ND(s, a, s') \leftarrow \text{DPRUNE}\left(\bigcup_{a' \in \mathcal{A}} \boldsymbol{Q}(s', a')\right)$
        Update reward distribution $\boldsymbol{r}(s, a, s')$ with $\boldsymbol{r}$
        $s \leftarrow s'$
    until *s is terminal*
return $DP\text{RUNE}\left(\bigcup_{a \in \mathcal{A}} \boldsymbol{Q}(s_0, a)\right)$

6.2 Dealing with Stochasticity

The original PQL update is formulated for deterministic transitions. In stochastic MOMDPs, a Bellman-style backup must aggregate over all possible successors. We therefore use the following distributional update, which makes the roles of each component explicit:

$$Q(s,a) \leftarrow \underbrace{\bigoplus_{s'} \hat{p}\big(s' \mid s, a\big)}_{\text{mixture over successors}} \underbrace{\Big[\underbrace{\boldsymbol{r}\big(s,a,s'\big)}_{\text{immediate reward distribution at } s'} + \gamma \underbrace{ND(s,a,s')}_{\text{non-dominated future returns from } s'} \Big]}_{\text{one-step shift \& discount at a fixed successor } s'}. \tag{4.10}$$

For each successor state s', the bracketed term forms all one-step return distributions reachable by first collecting the immediate reward distribution and then appending the discounted, non-dominated set of future returns. The set-valued mixture operator $\bigoplus_{s'} \hat{p}(s' \mid s, a)$ then takes all probability mixtures across successors, weighting each s' according to its transition probability. Intuitively, this is the distributional analogue of taking an expectation over next states while preserving the full multi-objective return structure.

In a learning setting the true transition kernel p is unknown. We estimate $\hat{p}$ via an initial exploration phase (random walks), using empirical frequencies for $\hat{p}(s' \mid s, a)$. During training we keep this estimate fixed. Freezing the mixture weights stabilises the backup and prevents a combinatorial growth of distinct mixtures that would otherwise arise from drift in the estimated transition probabilities.

6.3 Action Selection

Learning the DUS requires an action scoring and selection rule for sets of return distributions, requiring the usage of set metrics. In PQL, common choices include the hypervolume indicator [Guerreiro et al., 2021] and Chebyshev scalarisation [Van Moffaert et al., 2013]. In DIMOQ, these extend directly by first mapping each return distribution to its expectation and then evaluating the resulting set with the chosen metric. As a simple and efficient baseline, we propose linear utility scoring: fix weights $w \in \Delta^d$, define $u_w(\boldsymbol{v}) = w^\top \boldsymbol{v}$, and score a Q-set by its *mean expected utility*, $\frac{1}{|\mathcal{Q}|} \sum_{Z \in \mathcal{Q}} \mathbb{E}[u_w(Z)]$, selecting the action with the highest score. Linear scalarisation is inexpensive and can be replaced by a more faithful surrogate if additional information about the decision-maker is available.

6.4 Limiting the Set Sizes

The closure operations in Eq. (4.10) (one-step shifting, discounting, and mixing over successors) cause rapid growth of the Q-sets, so we control complexity with two complementary mechanisms within a single update cycle. First, we limit the precision of the distributions that are learned, which was already demonstrated to be successful in multi-objective dynamic programming [Mandow et al., 2022]. Second, we enforce a hard cap K on each Q-set: whenever $|\boldsymbol{Q}(s, a)| > K$, we perform agglomerative clustering into exactly K clusters using pairwise distances between distributions and replace each cluster by a random member. In our experiments, we compute the Jensen-Shannon distance between the flattened distributions. Alternatively, one could use the cost of optimal transport between pairs of distributions.

7 Case Study

To evaluate the contributions of this chapter, we present a case study where DIMOQ learns the DUS, which is then pruned to the CDUS, Pareto front, and convex hull using DPrune and CDPrune. Finally, we consider two concrete decision-makers to show how the proposed solution sets better support their needs.

Table 4.1: Configuration of the generated MOMDPs. Time steps refer to the maximum time horizon after which the episode is terminated.

Name	States	Actions	Next states	Time steps	Set limit
Small	5	2	$[1, 2]$	3	10
Medium	10	3	$[1, 2]$	5	15
Large	15	4	$[1, 2]$	7	20

Table 4.2: Runtime for DIMOQ on randomly generated MOMDPs.

Name	Mean	SD	Min	Max
Small	00:01:21	00:00:25	00:00:58	00:02:01
Medium	01:49:11	00:47:07	00:17:41	02:31:18
Large	17:01:25	06:02:35	09:46:06	27:55:55

7.1 Obtaining the Solution Sets

We evaluate DIMOQ (Algorithm 8) and CDPrune (Algorithm 7) on randomly generated MOMDPs of different sizes shown in Table 4.1. For each size category, we repeat the experiment with seeds one through five and perform $50,000$ random walks to estimate T followed by $2,000$ training episodes. All experiments considered two objectives, used a discount factor of 1 and limited the precision of distributions to three decimals. Finally, the experiments were run on a single core of an Intel Xeon Gold 6148 processor, with a maximum RAM requirement of 2GB.

Runtime analysis. The runtimes of DIMOQ in Table 4.2 grow substantially with the size of the MOMDP and also show considerable variance across seeds. These differences cannot be explained by transition complexity alone, but are likely due to interactions between the transition and reward functions. In particular, when transitions generate many undominated returns, each iteration must process a large number of combinations. Scaling thus becomes a key limitation of DIMOQ in larger state and action spaces. A natural direction for future work is to incorporate function approximation, analogous to the progression from Q-learning to DQN. Moreover, MOMDPs inspired by real-world scenarios are likely to exhibit additional structure, making them an important target for further study.

Subset analysis. In Table 4.3 we report the average size of the DUS together with the percentage of policies that also belong to the CDUS, Pareto front, and convex hull. Similar to the runtimes, we observe that larger MOMDPs produce larger solution sets. At the same time, they also allow a greater proportion of policies to be pruned into the smaller sets, which is beneficial for decision support.

Table 4.3: The relative sizes of the pruned subsets.

Name	DUS	CDUS	PF	CH
Small	13.0 ± 10.73	95.71% ± 8.57	39.88% ± 16.45	36.07% ± 20.12
Medium	372.2 ± 211.88	61.27% ± 12.16	6.28% ± 6.70	2.87% ± 3.48
Large	639.0 ± 221.71	53.00% ± 5.68	3.43% ± 2.01	1.33% ± 0.82

Although the CDUS is often substantially smaller than the DUS, the Pareto front and convex hull are much smaller still. Intuitively, this is because when maximising both objectives, Pareto-optimal policies can only occur in the upper-right region of the objective space, while policies in the DUS and CDUS may also lie in Pareto-dominated regions. Recall, however, from Example 19 that such policies may still be optimal under ESR.

7.2 Decision Support

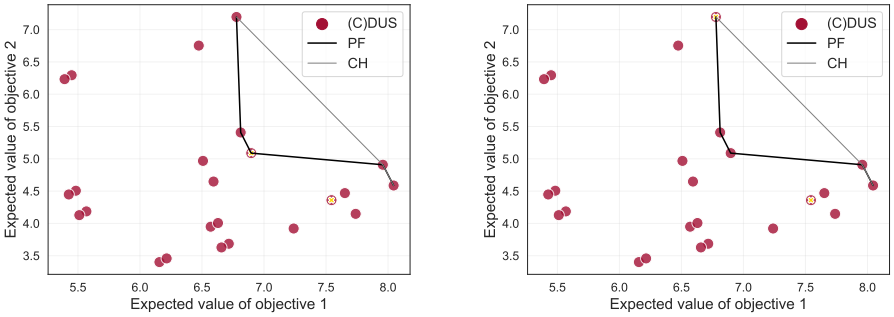
We use two representative utility functions that may be encountered in a real-world decision support setting to show why the extra policies in the DUS add value beyond the Pareto front. Both require attention to both objectives and favour balanced outcomes. We refer to the corresponding decision-makers as DM1 and DM2, with utilities

$$u_{\text{DM1}}(v_1, v_2) = v_1 v_2, \qquad u_{\text{DM2}}(v_1, v_2) = \min\{v_1, v_2\}. \tag{4.11}$$

The multiplicative utility u_{DM1} (also used in Example 19) rewards balance by penalising dispersion across objectives. The Leontief utility u_{DM2} reflects perfect complementarity and is standard in economics and game-theoretic models [Codenotti and Varadarajan, 2007]; improvements in a single objective do not increase u_{DM2} unless the other objective improves as well. Because both utilities value joint performance, they can select policies in the DUS that do not lie on the Pareto front, which motivates distributional solution sets for decision support.

In Fig. 4.2 we plot the expected returns of the policies learned by DIMOQ in a sample MOMDP. Conventional approaches in multi-objective reinforcement learning and planning focus on either the Pareto front (black line) or the convex hull (grey line). Under this practice, only policies in these sets are shown to decision-makers and all remaining policies are discarded a priori as suboptimal.

If a decision support system presents only policies with Pareto-optimal expected values, the policy marked with a yellow cross on the black line in Fig. 4.2a is optimal for DM1, yielding an expected utility of 34.87, while the policy marked in Fig. 4.2b is optimal for DM2, yielding 5.37.



(a) DM1. (b) DM2.

Figure 4.2: A case study comparing optimal policies in the Pareto front and distributional undominated set for two decision-makers.

Under a distributional approach, both decision-makers prefer the same policy, located in the Pareto-dominated region and indicated with a yellow cross. This policy yields an expected utility of 49.59 for DM1 and 6.49 for DM2, and is strictly preferred to all policies on the Pareto front or convex hull. Including distributional information in the solution set can therefore deliver substantially higher value to decision-makers.

Finally, although the CDUS is not guaranteed to contain the Pareto front in general, it did so in all of our experiments, as also visible in Fig. 4.2. Specifying conditions under which this inclusion is guaranteed is an interesting direction for future work.

8 Conclusion

In Chapter 3 we developed a decomposition-based method to learn the Pareto front in MOMDPs. This is valuable, yet it implicitly assumes that the Pareto front is the relevant target for the decision-maker. In this chapter we asked a broader question, namely what set of policies ought to be shown to an expected utility maximising decision-maker.

We introduced the *distributional undominated set* (DUS), which contains the Pareto front and all policies that are optimal for multivariate risk-averse decision-makers. We then proposed the *convex distributional undominated set* (CDUS), tailored to expected utility, and established a taxonomy linking the DUS, CDUS, Pareto front, and convex hull.

To enable practical use, we developed a learning procedure for the DUS, called DIMOQ, and pruning operators for the CDUS. In a case study on randomly generated MOMDPs, we first evaluated the performance of these algorithms and then analysed two concrete

decision-makers. The results show that our solution sets can be effectively pruned to their chosen targets and provide clear utility benefits over conventional ones, highlighting the value of distributional approaches for decision support.

Follow-up challenges. This chapter lays a foundation for distributional MORL, and several open challenges remain.

- **Generalising Theorems 8 and 9.** The current links between distributional dominance and expected utility are proven for two-dimensional returns. Extending these results to higher dimensions remains open.
- **Beyond risk aversion.** Theorems 8 and 9 are not only limited by their two-dimensionality but also by the fact that they only consider risk-averse decision-makers. Future work could investigate whether the stronger FSD notion in Section 2.3 enables broader guarantees.
- **Deep distributional MORL.** The DIMOQ algorithm introduced in Algorithm 8 represents an initial step towards learning the DUS but remains computationally demanding and confined to small, tabular MOMDPs. Incorporating function approximation offers a promising way to scale to larger domains while simultaneously alleviating the resource requirements of the tabular approach.

5 The Real World Contains Multiple Agents

This chapter is based on *Bridging the gap between single and multi objective games* [Röpke et al., 2023a] and *On nash equilibria in normal-form games with vectorial payoffs* [Röpke et al., 2022]. The two works have been merged, and all proofs are integrated into the main text. Our code is available at `https://github.com/wilrop/pure-strategy-equivalence` and `https://github.com/wilrop/moqups`.

1 Introduction

Up to this point, we have studied how to balance trade-offs in single-agent settings. Real-world problems, however, rarely involve a single decision-maker. Instead, multiple agents interact, pursuing their own goals while adapting to the behaviour of others. These interactions may be cooperative, competitive, or somewhere in between. In this chapter we focus on the competitive case, adopting a *game-theoretic* perspective in which agents strategically account for the actions of their opponents.

In earlier chapters (Chapters 3 and 4), we considered settings where the utility functions of agents were unknown and therefore learned sets of candidate solutions. In multi-agent problems this uncertainty becomes even more challenging: agents may not share utilities, and the joint action space grows exponentially with the number of agents, quickly leading to prohibitively large solution sets [Rădulescu et al., 2020a]. To make progress, this

chapter adopts the alternative assumption that utility functions are known. Under this assumption, the central solution concept becomes the *Nash equilibrium*, a joint strategy from which no agent can profitably deviate unilaterally [Nash, 1951]. We then address the following challenge introduced in Chapter 1:

> **Challenge 3:** *How can equilibrium solutions with multiple objectives be characterised, and how do they relate to classical game-theoretic models?*

Although one might expect that known utility functions reduce a multi-objective game to a single-objective one, prior work has revealed counterintuitive behaviour, including the possible non-existence of Nash equilibria under non-linear utility functions [Rădulescu et al., 2020b]. This raises the question: What is particular about multi-objective games?

Our results show that a multi-objective game with known utility functions is equivalent to a *continuous game* [Stein et al., 2008], in which each player selects actions from an infinite strategy space. Continuous games naturally arise in domains such as firms setting continuous prices to maximise profit [Judd et al., 2012]. Multi-objective normal-form games (MONFGs) [Blackwell, 1954], on the other hand, produce vector-valued payoffs and have been applied in settings ranging from economic regulation [Sinha et al., 2013] to household energy scheduling [Lu et al., 2022].

We introduce a new equivalence relation, called *pure-strategy equivalence*, which rigorously connects these two models. This relation preserves core game-theoretic structure, including Nash equilibria, and allows results and algorithms to transfer directly between the models. Because continuous games are extensively studied, this equivalence enables rapid theoretical and computational progress for MONFGs. Conversely, the compact representation of MONFGs provides algorithmic advantages that can be carried back to continuous games. We further identify conditions on MONFGs and utility functions under which equilibria exist in specific forms or can be computed efficiently.

Just as Chapter 3 reduced Pareto front learning to a sequence of single-objective problems, here we establish a bridge between multi-objective games and well-studied single-objective models. This connection encourages cross-fertilisation between communities, yielding new insights and filling long-standing gaps in both models.

Contributions. This chapter makes the following contributions:

1. **Pure-strategy equivalence.** We introduce the notion of pure-strategy equivalence, a formal mapping between continuous games and MONFGs. The mapping establishes a bijection from pure strategies in a continuous game to mixed strategies in an MONFG, while exactly preserving utilities.

2. **Preservation of Nash equilibria.** We prove that Nash equilibria are invariant under pure-strategy equivalence, creating a strong theoretical bridge between the two models.

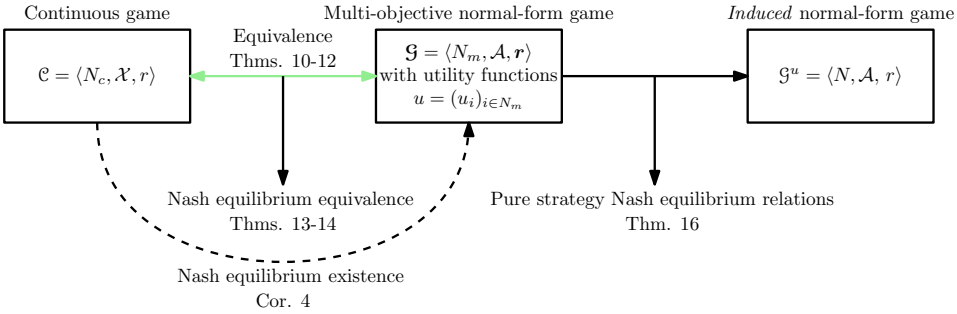


Figure 5.1: A visual overview of the core theoretical contributions in Chapter 5.

3. **Special cases.** We identify classes of MONFGs and utility functions for which Nash equilibria exhibit specific structural properties and admit efficient computation.

4. **Algorithmic methods.** We construct explicit mappings realising pure-strategy equivalence, adapt fictitious play from continuous games to compute equilibria in two-player MONFGs, and design an algorithm that provably recovers all pure-strategy Nash equilibria when utilities are quasiconvex.

Figure 5.1 summarises how the chapter's results link MONFGs with continuous and induced normal-form games, and highlights how pure-strategy equivalence supports the equilibrium and algorithmic results that follow.

Related work. Our work is related to other equivalence notions in game theory. The first notable example of such an equivalence notion is strategic equivalence [Maschler et al., 2013]. An advantage of strategic equivalence is that Nash equilibria are preserved, thus being a useful construct for computing Nash equilibria in games. For example, the Nash equilibria of an otherwise intractable game might be computed by constructing a strategically equivalent zero-sum game for which efficient solving methods do exist [Heyman, 2019].

Pure-strategy equivalence, as defined in Section 2.2, is most closely related to the concept of a game isomorphism that defines two games to be isomorphic when there exists a mapping from one to the other [Gabarró et al., 2011]. Two variants of a game isomorphism are defined, namely a strong and a weak isomorphism, with a strong isomorphism preserving all Nash equilibria and a weak isomorphism preserving only the pure-strategy Nash equilibria.

2 Equivalence Relation

Our main contribution establishes an equivalence relation between continuous and multi-objective games. With this goal in mind, we first introduce a special game, called an *identity game*, which plays a crucial role in bridging the two models. Next, we prove that a bijective mapping from pure strategies in the continuous game to mixed strategies in the MONFG always exists. To complete the full equivalence, we introduce a novel concept for MONFGs, which we call hierarchical strategies. Lastly, we show that Nash equilibria are preserved in the process, allowing us to guarantee a Nash equilibrium in hierarchical strategies in MONFGs. For all definitions of terms used in this section, we refer to Chapter 2 Sections 3 and 4.

2.1 Identity Game

An identity game returns, as the name suggests, a payoff vector equal to the strategy it received as input. We will use such games in Section 2.2 to prove that an equivalent MONFG can be constructed for every continuous game and vice versa. The following result shows that an identity game can always be constructed for finite players and actions.

Lemma 7: Identity game

For any finite set of players N and finite pure strategy sets $(A_i)_{i\in N}$, there exists a profile of payoff functions $(\boldsymbol{r}_i)_{i\in N}$ such that for every mixed joint strategy π one has $\boldsymbol{r}_i(\pi) = \pi$ for all $i \in N$.

Proof. Let $m_i := |A_i|$ and write a mixed strategy of player i as the probability vector

$$\pi_i = \big(\pi_i(a_{i,1}), \dots, \pi_i(a_{i,m_i})\big) \in \Delta(A_i).$$

Define the mixed joint strategy π as the concatenation of all players' mixed strategies, so its length is

$$|\pi| = \sum_{i\in N} m_i.$$

Let $\mathcal{A} := \times_{i\in N} A_i$ be the set of pure action profiles. For $a \in \mathcal{A}$, let π_a denote the joint strategy vector obtained by placing, for each i, a 1 at the coordinate corresponding to a_i and 0 elsewhere (that is, the concatenation of one-hot vectors for each player).

1. Construction.
For every $a \in \mathcal{A}$ and every $i \in N$, define the pure-profile payoff vector by

$$\boldsymbol{r}_i(a) := \pi_a \in \mathbb{R}^{|\pi|}.$$

Thus all players share the same pure-profile payoff vectors, and each such vector records, componentwise, which pure action was played by each player.

2. Verification.
Given a mixed joint strategy π, the expected payoff to player i is

$$\boldsymbol{r}_i(\pi) = \sum_{a \in \mathcal{A}} \boldsymbol{r}_i(a) \prod_{j \in N} \pi_j(a_j).$$

Consider a coordinate indexed by player $j \in N$ and an action $b \in A_j$, that is, the entry corresponding to player j choosing b. By construction,

$$\big[\boldsymbol{r}_i(a)\big]_{(j,b)} = \mathbf{1}\{a_j = b\}.$$

Therefore, the (j, b)-th coordinate of $\boldsymbol{r}_i(\pi)$ equals

$$\begin{aligned}
\big[\boldsymbol{r}_i(\pi)\big]_{(j,b)} &= \sum_{a \in \mathcal{A}} \mathbf{1}\{a_j = b\} \prod_{t \in N} \pi_t(a_t) \\
&= \sum_{a_{-j} \in \times_{t \neq j} A_t} \left(\prod_{t \neq j} \pi_t(a_t) \right) \pi_j(b) \\
&= \pi_j(b) \sum_{a_{-j}} \prod_{t \neq j} \pi_t(a_t) \\
&= \pi_j(b),
\end{aligned}$$

since the product distribution over a_{-j} sums to 1. As this holds for every pair (j, b), it follows that

$$\boldsymbol{r}_i(\pi) = \pi \qquad \text{for all } i \in N. \qquad \square$$

Having established that an identity game can always be constructed, it is helpful to see one in action. The next example makes the structure explicit by showing that the input policy is exactly the payoff the player receives.

Example 21: Identity game

Consider the identity game in Fig. 5.2. Assume that player one plays the mixed strategy $\pi_1 = (\frac{1}{2}, \frac{1}{2})$ and player two plays the mixed strategy $\pi_2 = (\frac{1}{3}, \frac{2}{3})$. This leads to a joint strategy $\pi = (\pi_1, \pi_2) = (\frac{1}{2}, \frac{1}{2}, \frac{1}{3}, \frac{2}{3})$. According to Lemma 7, the expected payoff vector should then also be $(\frac{1}{2}, \frac{1}{2}, \frac{1}{3}, \frac{2}{3})$. We verify this:

$$
\begin{aligned}
\boldsymbol{r}_i(\pi) &= \sum_{a \in \mathcal{A}} \boldsymbol{r}_i(a) \prod_{j=1}^{n} \pi_j(a_j) \\
&= (1,0,1,0) \cdot \frac{1}{6} + (1,0,0,1) \cdot \frac{1}{3} + (0,1,1,0) \cdot \frac{1}{6} + (0,1,0,1) \cdot \frac{1}{3} \\
&= \left(\frac{1}{6}, 0, \frac{1}{6}, 0\right) + \left(\frac{1}{3}, 0, 0, \frac{1}{3}\right) + \left(0, \frac{1}{6}, \frac{1}{6}, 0\right) + \left(0, \frac{1}{3}, 0, \frac{1}{3}\right) \\
&= \left(\frac{1}{2}, \frac{1}{2}, \frac{1}{3}, \frac{2}{3}\right) \\
&= \pi.
\end{aligned}
$$

	A	B
A	$(1,\ 0,\ 1,\ 0)$, $(1,\ 0,\ 1,\ 0)$	$(1,\ 0,\ 0,\ 1)$, $(1,\ 0,\ 0,\ 1)$
B	$(0,\ 1,\ 1,\ 0)$, $(0,\ 1,\ 1,\ 0)$	$(0,\ 1,\ 0,\ 1)$, $(0,\ 1,\ 0,\ 1)$

Figure 5.2: The identity game for a 2-player 2-action setting.

2.2 Pure-Strategy Equivalence

We introduce a novel equivalence notion between continuous games and MONFGs, called Pure-Strategy Equivalence (PSE). Informally, two games are pure-strategy equivalent whenever the pure strategies from the continuous game can be bijectively mapped to mixed strategies in the MONFG while keeping the corresponding utilities equal. We formally define this below and subsequently show that such a mapping always exists, allowing us to construct a pure-strategy equivalent MONFG for every continuous game.

Definition 30: Pure-strategy equivalence

A continuous game $\mathcal{C} = \langle N_c, \mathcal{X}, r \rangle$ is pure-strategy equivalent to a finite multi-objective normal-form game $\mathcal{G} = \langle N_m, \mathcal{A}, \boldsymbol{r} \rangle$ with utility functions $u = (u_i)_{i \in N_m}$ when there exists a tuple of functions (ϱ, φ) such that:

- $\varrho : N_c \to N_m$ is a bijective function called the *player bijection*;
- $\varphi = \varphi_1 \times \cdots \times \varphi_n$ where $\forall i \in N_c, \varphi_i : X_i \to \Delta\left(A_{\varrho(i)}\right)$ is a continuous bijective function with a continuous inverse, called the *strategy bijection*;
- $\forall i \in N_c, \forall x \in \mathcal{X}, r_i(x) = u_{\varrho(i)}\left(\boldsymbol{r}_{\varrho(i)}(\varphi(x))\right)$.

Theorem 10: MONFG pure-strategy equivalence guarantee

For every multi-objective normal-form game with continuous utility functions, there exists a pure-strategy equivalent continuous game.

Proof. Let $\mathcal{G} = \langle N_m, \mathcal{A}, \boldsymbol{r} \rangle$ be a multi-objective normal-form game with continuous utility functions $u = (u_i)_{i \in N_m}$. We will construct a continuous game $\mathcal{C} = \langle N_c, \mathcal{X}, r \rangle$ that is pure-strategy equivalent to $\mathcal{G}$.

1. Player bijection.
Set $N_c = N_m$. The player bijection is therefore $\varrho(i) = i$ for all i.

2. Strategy bijection.
In $\mathcal{G}$, the set of mixed strategies for player i is the simplex $\Delta(A_i)$. Define the continuous game strategy set for player i as

$$X_i := \Delta(A_i).$$

This set is a non-empty compact metric space, satisfying the definition of a continuous game strategy set. The strategy bijection is thus $\varphi_i = \mathbb{1}$ for all i.

3. Payoff functions.
Define each continuous game payoff function as

$$r_i := u_i \circ \boldsymbol{r}_i.$$

Since both u_i and $\boldsymbol{r}_i$ are continuous, their composition r_i is also continuous.

4. Conclusion.
We have constructed a continuous game $\mathcal{C}$ with the same players, identical mixed-strategy sets, and payoff functions preserving the original utilities. Therefore, $\mathcal{C}$ is pure-strategy equivalent to $\mathcal{G}$. □

We now show a converse to Theorem 10, namely that every continuous game with convex strategy sets can be mapped to a pure-strategy equivalent MONFG.

Theorem 11: Continuous game pure-strategy equivalence guarantee

For every continuous game whose strategy spaces are convex subsets of a Euclidean space, there exists a pure-strategy equivalent multi-objective normal-form game.

Proof. Let $\mathcal{C} = \langle N_c, \mathcal{X}, r \rangle$ be a continuous game in which each strategy set X_i is a compact, convex, non-empty subset of a Euclidean space. We will construct a finite multi-objective normal-form game $\mathcal{G} = \langle N_m, \mathcal{A}, \boldsymbol{r} \rangle$ with utility functions $u = (u_i)_{i \in N_m}$ that is pure-strategy equivalent to $\mathcal{C}$.

1. Player bijection.
Set $N_m = N_c$. The player bijection is $\varrho(i) = i$ for all i. From this point on, we refer simply to N for the common player set.

2. Strategy bijection when X_i has non-empty interior.
A known topological result states that every compact convex subset of $\mathbb{R}^k$ with non-empty interior is homeomorphic to the probability k-simplex [Bredon, 1993, Theorem 16.4]. Thus, for each $i \in N$ with $\text{int}(X_i) \neq \varnothing$, there exists a homeomorphism

$$f_i : X_i \to \Delta^{k_i},$$

where Δ^{k_i} is the simplex of mixed strategies over $k_i + 1$ pure actions. The vertices of Δ^{k_i} correspond to the pure actions of player i in $\mathcal{G}$, i.e., $\Delta^{k_i} = \Delta(A_i)$. Define the joint mapping

$$f(x) := f_1(x_1) \times \cdots \times f_n(x_n),$$

and take $\varphi = f$. Since each f_i is a homeomorphism, φ is a continuous bijection with a continuous inverse.

3. Case of empty interior.
If some X_i has empty interior in $\mathbb{R}^k$, then it has non-empty interior relative to its affine span. In this case X_i is homeomorphic to a k-simplex, where $k = \dim \text{aff}(X_i)$. The above construction applies identically using this relative interior.

4. Payoff and utility functions in $\mathcal{G}$.
Let the payoff functions $\boldsymbol{r}$ of $\mathcal{G}$ be those of the identity game from Lemma 7. Define each utility function in $\mathcal{G}$ by

$$u_i := r_i \circ f^{-1}.$$

Given any $x \in \mathcal{X}$ and corresponding $\pi = f(x)$, we have

$$\begin{aligned} u_i\big(\boldsymbol{r}_i(\varphi(x))\big) &= u_i\big(\boldsymbol{r}_i(f(x))\big) \\ &= u_i(\pi) \\ &= r_i\big(f^{-1}(\pi)\big) \\ &= r_i(x), \end{aligned}$$

and hence, for the full profile,

$$u_i\big(\boldsymbol{r}_i(\varphi(x))\big) = r_i(x) \qquad \forall i \in N.$$

This shows the payoffs are preserved under the mapping.

5. Conclusion.
We have constructed a finite MONFG $\mathcal{G}$ with players N, strategies in bijection with those of $\mathcal{C}$, and utility functions that reproduce the payoffs of $\mathcal{C}$ via the identity-game mapping. Thus $\mathcal{C}$ and $\mathcal{G}$ are pure-strategy equivalent. □

The proof proceeds by representing each convex strategy set via a homeomorphism to a probability simplex and then reusing the original utilities through composition. Concretely, the identity game in Lemma 7 returns the mixed strategy itself as the payoff. For each player i, a homeomorphism $f_i : X_i \to \Delta^{k_i}$ maps continuous strategies to mixed strategies over finitely many pure actions. Defining utilities in the MONFG by $u_i := r_i \circ f_i^{-1}$ ensures that when the identity game outputs a mixed strategy, evaluating u_i reproduces r_i at the preimage in X_i. Hence payoffs are preserved under the bijection between profiles, which yields pure-strategy equivalence.

Non-convex strategy sets. Theorem 11 extends immediately to games whose strategy sets are non-convex but nevertheless homeomorphic to a simplex. When this is not the case, a natural workaround is to approximate the original game by convexifying each set, for example by replacing X_i with its convex hull. One can then construct a PSE MONFG for the modified game and interpret the result as an approximately PSE representation of the original. We illustrate this approach in Section 6.3 and show that it can still capture the essential strategic structure.

Uniqueness of PSE. Taken together, Theorems 10 and 11 establish the bidirectional existence of pure-strategy equivalent games, but not their uniqueness. Different choices of homeomorphisms and payoff specifications can yield multiple PSE games, and the constructions given in the proofs are not the only possibilities. These variations may have computational implications: some PSE representations can be considerably easier to solve than others, consistent with findings on strategically equivalent reductions in related settings [Heyman, 2019].

Proposition 7: Non-uniqueness of pure-strategy equivalent games

A continuous game may have multiple pure-strategy equivalent multi-objective normal-form games and vice versa.

	A	B
A	$(0,0);(0,0)$	$(0,3);(0,3)$
B	$(3,0);(3,0)$	$(3,3);(3,3)$

Figure 5.3: The MONFG used in the proof of Proposition 7.

Proof. Let $\mathcal{C}$ be the continuous game with $X_1 = X_2 = [0,3]$ and $r_i(x_1, x_2) = x_1 + x_2$ for $i \in \{1,2\}$.

1. One MONFG pure-strategy equivalent to $\mathcal{C}$.
Consider the 2×2 MONFG in Fig. 5.3 with pure actions $\{A, B\}$ per player and vector

payoffs

$$\boldsymbol{r}_i(A,A) = (0,0), \quad \boldsymbol{r}_i(A,B) = (0,3), \quad \boldsymbol{r}_i(B,A) = (3,0), \quad \boldsymbol{r}_i(B,B) = (3,3).$$

Define the strategy bijection $\varphi_i : [0,3] \to \Delta(\{A,B\})$ by

$$\varphi_i(x) := \left(1 - \tfrac{x}{3},\ \tfrac{x}{3}\right),$$

interpreted in the coordinate order (A,B). Let $u_i(z_1,z_2) = z_1 + z_2$. For $x = (x_1,x_2)$ and $\pi = \varphi(x)$, the expected vector payoff is $(3 \cdot \frac{x_1}{3},\ 3 \cdot \frac{x_2}{3}) = (x_1,x_2)$, hence

$$u_i(\mathbb{E}\,[\boldsymbol{r}_i(\pi)]) = x_1 + x_2 = r_i(x).$$

Thus this MONFG is pure-strategy equivalent to $\mathcal{C}$.

2. A second, distinct MONFG also equivalent to $\mathcal{C}$.
Scale the above payoffs by 2, i.e., $\boldsymbol{r}'_i = 2\,\boldsymbol{r}_i$, and define $u'_i(z_1,z_2) = \frac{1}{2}(z_1+z_2)$. With the same φ, we obtain

$$u'_i\big(\mathbb{E}\left[\boldsymbol{r}'_i(\pi)\right]\big) = \tfrac{1}{2} \cdot 2\left(\tfrac{x_1}{3} \cdot 3 + \tfrac{x_2}{3} \cdot 3\right) = x_1 + x_2 = r_i(x),$$

so $\mathcal{C}$ is also pure-strategy equivalent to this *distinct* MONFG.

3. Non-uniqueness in the opposite direction.
Fix the MONFG of Fig. 5.3. It is pure-strategy equivalent to the above $\mathcal{C}$ via $\varphi_i(x) = \left(1 - \frac{x}{3},\ \frac{x}{3}\right)$ and also to the continuous game $\mathcal{C}'$ with $X'_1 = X'_2 = [0,1]$ and $r'_i(x_1,x_2) = 3x_1 + 3x_2$, via $\varphi'_i(x) = (1-x,\ x)$. In both cases,

$$u_i(\mathbb{E}\,[\boldsymbol{r}_i(\varphi(\cdot))]) = r_i(\cdot), \qquad u_i\big(\mathbb{E}\left[\boldsymbol{r}_i(\varphi'(\cdot))\right]\big) = r'_i(\cdot),$$

so the same MONFG admits multiple pure-strategy equivalent continuous games.

4. Conclusion.
A continuous game may have multiple pure-strategy equivalent MONFGs, and an MONFG may have multiple pure-strategy equivalent continuous games. □

2.3 Mixed Strategy Equivalence

Motivated by the pure-strategy equivalence results, a natural question is how to represent the mixed strategies of a continuous game within an equivalent MONFG. To this end, we introduce a strategy concept for MONFGs that permits "mixing over mixed strategies", which we call *hierarchical strategies*.

Definition 31: Hierarchical strategy

For player i with mixed-strategy simplex $\Delta(A_i)$, a *hierarchical strategy* is a Borel probability measure on $\Delta(A_i)$; that is,

$$\mu_i \in \mathcal{B}\big(\Delta(A_i)\big).$$

While mainly used as an instrument in our analysis, hierarchical strategies may also have an applied use in multi-objective games. Concretely, hierarchical strategies are appropriate to consider when agents have to decide on a strategy which is then executed for a given period without the possibility for downstream adjustments. For example, agents joining an auction may have to commit a single mixed strategy to an automated auctioneer a priori, after which auctions are held for a number of rounds without further input from the agents.

Expected utility. We can naturally extend the definition of expected utility to hierarchical strategies. Write $\boldsymbol{r}_i(\pi_1, \dots, \pi_n)$ for player i's payoff under the mixed profile $(\pi_1, \dots, \pi_n)$ in the MONFG, and u_i for the player's utility function on payoffs. Given a profile of hierarchical strategies $(\mu_1, \dots, \mu_n)$, we define player i's expected utility by integrating u_i over the product measure:

$$u_i(\mu_1, \dots, \mu_n) \;=\; \int_{\Delta(A_1)\times\cdots\times\Delta(A_n)} u_i(\boldsymbol{r}_i(\pi_1, \dots, \pi_n))\; \mathrm{d}\big(\mu_1 \otimes \cdots \otimes \mu_n\big)(\pi_1, \dots, \pi_n). \tag{5.1}$$

This mirrors the standard construction for mixed strategies in continuous games, now lifted to probability measures over mixed strategies. Note that we overload the notation of utility for actions, mixed strategies and hierarchical strategies by simply using $u_i(\cdot)$ in all cases, as the argument type, and hence the required computation, is always clear from context.

Collapsing the hierarchical strategy. It may be tempting to "push" the outer randomisation μ_i down to a single mixed strategy $\bar{\pi}_i := \int \pi_i \, \mathrm{d}\mu_i(\pi_i)$ and thereby recover an equivalent ordinary mixed strategy profile. In general this is *not* valid. The obstruction is the non-linearity of u_i: utility is applied *after* payoffs are realised from a mixed profile, so

$$\int u_i(\boldsymbol{r}_i(\pi_1, \dots, \pi_n))\; \mathrm{d}(\mu_1 \otimes \cdots \otimes \mu_n) \;\neq\; u_i(\boldsymbol{r}_i(\bar{\pi}_1, \dots, \bar{\pi}_n))$$

whenever u_i is non-linear. Thus, the probabilities in a hierarchical strategy cannot, in general, be redistributed to yield an equivalent ordinary mixed strategy. We demonstrate this more concretely in Example 22.

	A	B
A	$(3,1);(3,1)$	$(1,3);(1,3)$
B	$(1,3);(1,3)$	$(3,1);(3,1)$

Figure 5.4: The game used in Example 22.

Example 22: Hierarchical strategy

Consider the game in Fig. 5.4 and utility functions

$$u_1(x,y) = u_2(x,y) = x^2 + y^2. \tag{5.2}$$

For the row player, we define two strategies, $\pi_{1,1} = (1,0)$ and $\pi_{1,2} = (0,1)$. For simplicity, we assume a deterministic strategy $\pi_2 = (1,0)$ for the column player where they always play action A. For both players, the joint strategy $(\pi_{1,1}, \pi_2)$ leads to an expected payoff vector of $(3,1)$ and $(\pi_{1,2}, \pi_2)$ to an expected payoff vector of $(1,3)$. Both joint strategies individually result in a utility of 10.

Consider now the following hierarchical strategy for the row player,

$$\mu_1 = \left(P(\pi_{1,1}) = \frac{1}{2}, P(\pi_{1,2}) = \frac{1}{2} \right).$$

This hierarchical strategy denotes the fact that they will play $\pi_{1,1}$ with 50% probability and $\pi_{1,2}$ with 50% probability. As both strategies result in a utility of 10, the expected utility of μ_1 is also 10. However, distributing the probabilities in μ_1 to form a mixed strategy, $\pi_{1,3} = (\frac{1}{2}, \frac{1}{2})$, results in an expected payoff vector of $(2,2)$ with a utility of 8. This demonstrates that a hierarchical strategy cannot be distributed to form an equivalent mixed strategy.

Formalising the equivalence. We now define mixed strategy equivalence between a continuous game and an MONFG. Informally, this equivalence notion generalises pure-strategy equivalence to relate mixed strategies from the continuous game with hierarchical strategies in the MONFG.

Definition 32: Mixed strategy equivalence

Let $\mathcal{C} = \langle N_c, \mathcal{X}, r\rangle$ be a continuous game and $\mathcal{G} = \langle N_m, \mathcal{A}, \boldsymbol{r}\rangle$ a finite multi-objective normal-form game with utility functions $u = (u_i)_{i\in N}$. $\mathcal{C}$ is mixed strategy equivalent to $\mathcal{G}$ if they are pure-strategy equivalent with (ϱ, φ) and there exists a function ψ such that,

- $\psi = \psi_1 \times \cdots \times \psi_n$ where $\forall i \in N_c, \psi_i : \mathcal{B}(X_i) \to \mathcal{B}(\Delta(A_i))$ is a bijective function;
- $\forall i \in N_c, \forall \mu \in \mathcal{B}(\mathcal{X}), r_i(\mu) = u_{\varrho(i)}(\psi(\mu))$.

We remark that, by definition, mixed strategy equivalence implies pure-strategy equivalence. Moreover, recall that the definition of mixed strategies in continuous games is similar to the definition of the set of hierarchical strategies in MONFGs. This is no coincidence and allows us to show that whenever a continuous game is pure-strategy equivalent to an MONFG, it also implies mixed strategy equivalence. To prove this result, we need the following folklore result, for which we provide a proof for completeness.

Lemma 8: Pushforward bijection

Let X, Y be topological spaces with Borel σ-algebras Σ_X, Σ_Y and sets of Borel probability measures $\mathcal{B}(X), \mathcal{B}(Y)$ respectively. If $\varphi : X \to Y$ is a homeomorphism, then the map

$$\psi : \mathcal{B}(X) \to \mathcal{B}(Y) : \mu \mapsto \varphi_*(\mu)$$

is a bijection, where $\varphi_*(\mu)$ denotes the *pushforward measure* of μ under φ, defined by $\varphi_*(\mu)(B) = \mu(\varphi^{-1}(B))$ for all $B \in \Sigma_Y$.

Proof. Let us first remark that

$$\{\varphi^{-1}(B) : B \subseteq Y \text{ open}\} = \{A : A \subseteq X \text{ open}\} \tag{1}$$

because φ is a homeomorphism. Therefore, the set above generates Σ_X, and a Borel measure on X is uniquely determined via the values it takes on that set.
To see that ψ is injective, let $\mu, \mu' \in \mathcal{B}(X)$. Then

$$\begin{aligned} & \psi(\mu) = \psi(\mu') \\ \implies & \varphi_*(\mu) = \varphi_*(\mu') \\ \implies & \mu\left(\varphi^{-1}(B)\right) = \mu'\left(\varphi^{-1}(B)\right) \quad \forall B \subseteq Y \text{ open} \\ \implies & \mu = \mu'. \end{aligned}$$

The last step uses again that (1) is a generating set for Σ_X, which implies that any two Borel measures that take the same value on this set, must be equal.

To see that ψ is also surjective, let $\mu_Y \in \mathcal{B}(Y)$ and we simply define $\mu_X \in \mathcal{B}(X)$ on the generating set (1) again:

$$\mu_X(\varphi^{-1}(B)) = \mu_Y(B), \quad \forall B \subseteq Y \text{ open.}$$

By construction, $\psi(\mu_X) = \mu_Y$ since for all open $B \subseteq Y$,

$$\psi(\mu_X)(B) = \mu_X(\varphi^{-1}(B)) = \mu_Y(B),$$

so since the measures take the same values on a generating set for Σ_Y, they take the same values on Σ_Y. □

Theorem 12: Mixed strategy equivalence guarantee

If a continuous game is pure-strategy equivalent to a multi-objective normal-form game, they are also mixed strategy equivalent.

Proof. Let $\mathcal{C} = \langle N_c, \mathcal{X}, v\rangle$ and $\mathcal{G} = \langle N_m, \mathcal{A}, \boldsymbol{r}\rangle$ with utility functions $u = (u_i)_{i\in N}$ be pure-strategy equivalent. For notational simplicity, we assume the player bijection $\varrho : N_c \to N_m$ to be implicitly applied in any mapping between $\mathcal{C}$ and $\mathcal{G}$ and refer to the set of players simply as N. Let $\varphi = \varphi_1 \times \cdots \times \varphi_n$ be the homeomorphism from pure strategies $x \in \mathcal{X}$ to mixed strategies $\pi \in \Delta(\mathcal{A})$ such that, for all i and all x,

$$u_i(\mathbb{E}[\boldsymbol{r}_i(\varphi(x))]) = r_i(x). \tag{1}$$

1. Pushforward on mixed strategies.
For each player i, Lemma 8 gives a bijection

$$\psi_i : \mathcal{B}(X_i) \to \mathcal{B}(\Delta(A_i)), \qquad \psi_i(\mu_i) = \varphi_{i*}(\mu_i).$$

Set $\psi := \psi_1 \times \cdots \times \psi_n$. Given $\mu = (\mu_1, \dots, \mu_n) \in \prod_i \mathcal{B}(X_i)$, the image $\psi(\mu)$ is a profile of Borel probability measures over the mixed-strategy simplices.

2. Equality of expected utilities via change of variables.
Using (1) and the definition of pushforward,

$$\begin{aligned}
\mathbb{E}[r_i(\mu)] &= \int_{X_1\times\cdots\times X_n} r_i(x_1, \cdots, x_n)\mathrm{d}\big(\mu_1 \otimes \cdots \otimes \mu_n\big)(x_1, \dots, x_n) \\
&= \int_{X_1\times\cdots\times X_n} u_i\left(\mathbb{E}\left[\boldsymbol{r}_i\left(\varphi\left(x_1, \cdots, x_n\right)\right)\right]\right) \mathrm{d}\big(\mu_1 \otimes \cdots \otimes \mu_n\big)(x_1, \dots, x_n) \\
&= \int_{\mathcal{B}(\Delta(A_1))\times\cdots\times\mathcal{B}(\Delta(A_n))} u_i\left(\mathbb{E}\left[\boldsymbol{r}_i\left(\pi_1, \cdots, \pi_n\right)\right]\right) \mathrm{d}\big(\psi_1(\mu_1) \otimes \cdots \otimes \psi_n(\mu_n)\big)(\pi_1, \dots, \pi_n) \\
&= \mathbb{E}\left[u_i(\psi(\mu))\right].
\end{aligned}$$

Hence the expected utility from μ in $\mathcal{C}$ equals that from $\psi(\mu)$ in $\mathcal{G}$ for every player i.

3. Conclusion.
Since ψ is a bijection and preserves expected utilities, $\mathcal{C}$ and $\mathcal{G}$ are mixed-strategy equivalent. □

3 Constructing Equivalent Games

The constructive nature of the proofs for Theorems 10 and 11 provides explicit recipes for building pure-strategy-equivalent pairs of continuous games and MONFGs. In this section, we package those recipes into practical algorithms and discuss their implications for algorithmic work in multi-objective planning and reinforcement learning.

3.1 From MONFGs to Continuous Games

The argument underlying Theorem 10 yields a direct procedure for constructing a pure-strategy-equivalent continuous game from any given MONFG. We summarise the steps in Algorithm 9. A key feature of this construction is that the strategy sets coincide across the two models. Consequently, the strategy bijection is the identity map. From an algorithmic standpoint, this eliminates any translation overhead between representations since strategies found in one model can be evaluated, compared, and refined in the other without conversion costs.

Two immediate consequences follow. First, observe that we may also perform the construction in single-player settings, such as those studied in multi-objective planning and reinforcement learning. This suggests that algorithms designed for continuous action spaces can be used whenever the utility function of the agent is known a priori. Second, if the resulting utility functions are (twice) continuously differentiable, the constructed game falls into the class of differentiable games. This enables the use of efficient gradient-based methods for equilibrium computation [Letcher et al., 2019].

Algorithm 9: Continuous game construction from an MONFG.

Input: An MONFG $\mathcal{G} = \langle N_m, \mathcal{A}, \boldsymbol{r} \rangle$ and utility functions $u = (u_i)_{i \in N}$
Output: A continuous game $\mathcal{C} = \langle N_c, \mathcal{X}, r \rangle$

$N_c \leftarrow N_m$
$\mathcal{X} \leftarrow \Delta(A_1) \times \cdots \times \Delta(A_n)$
$r \leftarrow (u_1 \circ \boldsymbol{r}_1, \ldots, u_n \circ \boldsymbol{r}_n)$
$\mathcal{C} \leftarrow \langle N_c, \mathcal{X}, r \rangle$
return $\mathcal{C}$

3.2 From Continuous Games to MONFGs

The construction defined in Theorem 11 also establishes a computational approach to transform a continuous game to an MONFG, formalised in Algorithm 10. However, contrary to the other direction, the strategy bijection does appear here. As such, the algorithm requires the strategy bijection to be explicitly defined. A potential drawback is that this function is not guaranteed to be efficiently computable, hence rendering the procedure intractable for some applications.

Algorithm 10: MONFG construction from a continuous game.

Input: A continuous game $\mathcal{C} = \langle N_c, \mathcal{X}, r \rangle$ and homeomorphisms $\varphi_i : X_i \to \Delta(A_i)$
Output: An MONFG $\mathcal{G} = \langle N_m, \mathcal{A}, \boldsymbol{r} \rangle$ and utility functions $u = (u_i)_{i \in N}$

$N_m \leftarrow N_c$
$\mathcal{A} \leftarrow [k_1 + 1] \times \cdots \times [k_n + 1]$
$\boldsymbol{r} \leftarrow \textsc{IdentityPayoffs}(N_m, \mathcal{A})$
$\mathcal{G} \leftarrow \langle N_m, \mathcal{A}, \boldsymbol{r} \rangle$
$u \leftarrow (r_1 \circ \varphi_1^{-1}, \dots, r_n \circ \varphi_n^{-1})$
return $\mathcal{G}, u$

Note that $[k_i + 1] = \{1, \dots, k_i + 1\}$ where k_i is the number of vertices in the simplex homeomorphic to player i's strategy set. The function IdentityPayoffs returns the payoffs of the identity game for a given player base and joint action set (see Lemma 7).

3.3 Constructing the Strategy Bijections

To construct an MONFG from a continuous game using Algorithm 10, it is necessary to provide a strategy bijection $\varphi_i : X_i \to \Delta(A_i)$ for every player i. We provide a straightforward, yet not necessarily efficient, approach for obtaining such functions. Afterwards, we present a short discussion on other techniques that may be better suited for this task.

Standard technique. When no obvious homeomorphism is available, it is possible to first construct a map from any player i's strategy space X_i to the closed unit ball $B \subset \mathbb{R}^{k_i}$. We can subsequently create a homeomorphism from $B \subset \mathbb{R}^{k_i}$ to a probability simplex Δ^{k_i}. By composing the two, we obtain a homeomorphism from X_i to the probability simplex Δ^{k_i}.

Let C be a compact convex subset in $\mathbb{R}^d$ with non-empty interior and define ∂C as its boundary. Let $f : \partial C \to S^{d-1}$ be defined by

$$f(x) = \frac{x}{\|x\|}. \tag{5.3}$$

This maps the boundary points of C to the $(d-1)$-sphere. Intuitively, for a 2-dimensional convex compact set C, f maps the boundary ∂C to a circle. The map f is a homeomorphism, so has an inverse f^{-1}, and the map $h : B^d \to C$ defined by

$$h(x) = \begin{cases} \|x\| f^{-1}\left(\frac{x}{\|x\|}\right) & x \neq 0, \\ 0 & x = 0, \end{cases} \tag{5.4}$$

is also a homeomorphism (see e.g. Bredon [1993]). Note that the construction assumes the origin is in the interior of C, which is always possible to accomplish by translation.

To complete the full construction, we also specify the inverse functions $f^{-1} : S^{d-1} \to \partial C$ and $h^{-1} : C \to B^d$. First, $x = f^{-1}(y)$ can be obtained by noticing that x is the place where the ray through y intersects ∂C. Let $p_A(y) := \inf\{\lambda > 0 : y \in \lambda A\}$ be a Minkowski functional [Rudin, 1991]. Informally, a Minkowski functional p_A returns for an input point y the smallest positive number by which it is possible to scale A such that y is in the resulting space. We define

$$f^{-1}(y) = \frac{y}{p_C(y)}. \tag{5.5}$$

Finally, $h^{-1} : C \to B^d$ can be constructed by first computing where the ray through y intersects ∂C and rescaling:

$$h^{-1}(y) = \begin{cases} p_C(y)\frac{y}{\|y\|} & y \neq 0, \\ 0 & y = 0, \end{cases}. \tag{5.6}$$

As stated earlier, we may use these functions to go from any compact convex subset of a Euclidean space C to the closed unit ball and we can apply the same procedure to map from the probability simplex to the unit ball. By composing the two, a full homeomorphism is obtained between C and the probability simplex.

Practical alternatives. While the proposed approach is straightforward to explain, it may be difficult to implement in practice. This is because computing the Minkowski functional $p_{\partial C}$ requires searching over a continuous range. One possible solution is to utilise a binary search algorithm that locates a λ with a desired precision. However, as this approach may be computationally expensive, we suggest exploring more efficient techniques whenever possible. Another option is to cache values of λ and reuse them when feasible to avoid the need for repeated binary searches.

When leveraging pure-strategy equivalence to solve continuous games, it may be beneficial to employ algorithms that necessitate only a limited number of function evaluations to avoid costly computations. Finally, rather than employing an exact solution for the strategy bijections, it may be useful to *learn* such bijections. There is exciting work in learning neural bijective functions that may be used for this purpose [Ardizzone et al., 2019; Behrmann et al., 2019].

4 Mapping of Nash Equilibria

Our next theoretical contributions consider Nash equilibria in pure-strategy equivalent games. Theorem 13 first specifies that pure-strategy Nash equilibria in continuous games correspond to mixed strategy Nash equilibria in pure-strategy equivalent MONFGs. Intuitively, this is clear as utilities for pure strategies in the continuous game were already guaranteed to be equal to the utilities for mixed strategies in the MONFG. Therefore, if a joint strategy cannot be improved upon unilaterally in either game, the mapped joint strategy in the related game can also not be improved upon.

Theorem 13: Nash equilibrium equivalence

A pure strategy is a Nash equilibrium in a continuous game if and only if it is a mixed strategy Nash equilibrium in a pure-strategy equivalent multi-objective normal-form game.

Proof. Let $\mathcal{C} = \langle N_c, \mathcal{X}, r\rangle$ be a continuous game and $\mathcal{G} = \langle N_m, \mathcal{A}, \boldsymbol{r}\rangle$ a pure-strategy equivalent MONFG with utilities $u = (u_i)_{i\in N}$. For notational simplicity we assume the player bijection $\varrho : N_c \to N_m$ to be implicitly applied in any mapping between $\mathcal{C}$ and $\mathcal{G}$ and refer to the set of players simply as N. Let $\varphi = \varphi_1 \times \cdots \times \varphi_n$ be the homeomorphism mapping pure strategies $x \in \mathcal{X}$ to mixed strategies $\pi \in \prod_i \Delta(A_i)$ such that, for all i and all x,

$$u_i\big(\mathbb{E}\left[\boldsymbol{r}_i\big(\varphi(x)\big)\right]\big) = r_i(x). \tag{1}$$

1. From pure NE in $\mathcal{C}$ to mixed NE in $\mathcal{G}$.
Assume $x^* \in \mathcal{X}$ is a pure-strategy Nash equilibrium in $\mathcal{C}$:

$$\forall i \in N,\ \forall x_i \in X_i :\quad r_i(x_i^*, x_{-i}^*) \ \geq\ r_i(x_i, x_{-i}^*).$$

Fix i and let $\pi_i \in \Delta(A_i)$ be any mixed deviation in $\mathcal{G}$. By bijectivity of φ_i, there exists $x_i \in X_i$ with $\pi_i = \varphi_i(x_i)$. Using (1),

$$u_i\big(\mathbb{E}\left[\boldsymbol{r}_i\big(\varphi(x^*)\big)\right]\big) = r_i(x^*) \ \geq\ r_i(x_i, x_{-i}^*) = u_i\big(\mathbb{E}\left[\boldsymbol{r}_i\big(\pi_i, \varphi_{-i}(x_{-i}^*)\big)\right]\big)\,.$$

Hence no mixed deviation in $\mathcal{G}$ is profitable, so $\varphi(x^*)$ is a mixed-strategy Nash equilibrium of $\mathcal{G}$.

2. From mixed NE in $\mathcal{G}$ to pure NE in $\mathcal{C}$.
Assume $\pi^* \in \Delta(\mathcal{A})$ is a mixed-strategy Nash equilibrium in $\mathcal{G}$:

$$\forall i \in N,\ \forall \pi_i \in \Delta(A_i) :\quad u_i\big(\mathbb{E}\left[\boldsymbol{r}_i(\pi_i^*, \pi_{-i}^*)\right]\big) \ \geq\ u_i\big(\mathbb{E}\left[\boldsymbol{r}_i(\pi_i, \pi_{-i}^*)\right]\big)\,.$$

Let $x^* = \varphi^{-1}(\pi^*)$. For any pure deviation $x_i \in X_i$, set $\pi_i = \varphi_i(x_i)$ and use (1):

$$r_i(x^*) = u_i(\mathbb{E}\left[\boldsymbol{r}_i(\pi^*)\right]) \ \geq\ u_i\big(\mathbb{E}\left[\boldsymbol{r}_i(\pi_i, \pi_{-i}^*)\right]\big) = r_i(x_i, x_{-i}^*).$$

Thus no pure deviation in $\mathcal{C}$ is profitable, and x^* is a pure-strategy Nash equilibrium of $\mathcal{C}$.

3. Conclusion.
A pure strategy is a Nash equilibrium in $\mathcal{C}$ if and only if its image under φ is a mixed-strategy Nash equilibrium in the pure-strategy equivalent $\mathcal{G}$. □

We now define the notion of a hierarchical Nash equilibrium, allowing us to mirror the earlier argument. In particular, every mixed-strategy Nash equilibrium of a continuous game must correspond to a hierarchical-strategy Nash equilibrium in any pure-strategy equivalent MONFG.

Definition 33: Hierarchical Nash equilibria in MONFGs

A hierarchical strategy profile μ^* is a hierarchical Nash equilibrium if

$$u_i(\mu_i^*, \mu_{-i}^*) \geq u_i(\mu_i, \mu_{-i}^*),$$

for all players i and alternative hierarchical strategies $\mu_i \in \mathcal{B}\big(\Delta(A_i)\big)$.

Theorem 14: Mixed to hierarchical Nash equilibrium

A mixed strategy is a Nash equilibrium in a continuous game if and only if it is a hierarchical Nash equilibrium in a pure-strategy equivalent multi-objective normal-form game.

Proof. Let $\mathcal{C} = \langle N_c, \mathcal{X}, r\rangle$ and $\mathcal{G} = \langle N_m, \mathcal{A}, \boldsymbol{r}\rangle$ with utility functions $u = (u_i)_{i\in N}$ be pure-strategy equivalent. By Theorem 12, for each player i there is a bijection

$$\psi_i : \mathcal{B}(X_i) \longrightarrow \mathcal{B}\big(\Delta(A_i)\big), \qquad \psi_i(\mu_i) = \varphi_{i*}(\mu_i),$$

and for all profiles $\mu \in \prod_i \mathcal{B}(X_i)$,

$$r_i(\mu) = u_i\big(\psi(\mu)\big), \qquad \text{where } \psi := \psi_1 \times \cdots \times \psi_n. \tag{1}$$

1. From mixed NE in $\mathcal{C}$ to hierarchical NE in $\mathcal{G}$.
Assume μ^* is a mixed-strategy Nash equilibrium in $\mathcal{C}$:

$$\forall i \in N,\ \forall \mu_i \in \mathcal{B}(X_i) : \quad r_i(\mu_i^*, \mu_{-i}^*) \geq r_i(\mu_i, \mu_{-i}^*).$$

Apply (1) to obtain

$$u_i\big(\psi(\mu^*)\big) \geq u_i\big(\psi_i(\mu_i),\ \psi_{-i}(\mu_{-i}^*)\big) \quad \forall i \in N,\ \forall \mu_i \in \mathcal{B}(X_i).$$

Since ψ_i is bijective, the right-hand side ranges over all hierarchical deviations in $\mathcal{B}(\Delta(A_i))$. Hence $\psi(\mu^*)$ is a hierarchical Nash equilibrium in $\mathcal{G}$.

2. From hierarchical NE in $\mathcal{G}$ to mixed NE in $\mathcal{C}$.
Assume $\nu^* \in \prod_i \mathcal{B}(\Delta(A_i))$ is a hierarchical Nash equilibrium in $\mathcal{G}$:

$$\forall i \in N,\ \forall \nu_i \in \mathcal{B}(\Delta(A_i)) :\quad u_i(\nu_i^*, \nu_{-i}^*) \geq u_i(\nu_i, \nu_{-i}^*).$$

Let $\mu^* = \psi^{-1}(\nu^*)$. Using (1) and bijectivity of ψ_i,

$$r_i(\mu^*) = u_i(\nu^*) \geq u_i\big(\nu_i,\ \nu_{-i}^*\big) = r_i\Big(\psi_i^{-1}(\nu_i),\ \mu_{-i}^*\Big) \quad \forall i \in N,\ \forall \nu_i,$$

which shows that μ^* is a mixed-strategy Nash equilibrium in $\mathcal{C}$.

3. Conclusion.
A mixed strategy is a Nash equilibrium in $\mathcal{C}$ if and only if it is a hierarchical Nash equilibrium in the pure-strategy equivalent $\mathcal{G}$. □

These properties are significant because they allow both algorithmic techniques for computing Nash equilibria and theoretical guarantees to be transferred between the two game models. As a direct consequence of Theorems 13 and 14, we can state two general existence results for Nash equilibria in MONFGs, presented below.

Corollary 4: Equilibrium existence in finite MONFGs

Let $\mathcal{G} = \langle N, \mathcal{A}, \boldsymbol{r} \rangle$ be a finite MONFG with continuous utility functions $u = (u_i)_{i \in N}$.

1. There exists a hierarchical Nash equilibrium.
2. If each u_i is quasiconcave, then there exists a mixed-strategy Nash equilibrium.

Proof. **1. Existence of a hierarchical Nash equilibrium.**
Construct the continuous game $\mathcal{C}$ from $\mathcal{G}$ as in Algorithm 9: take $X_i = \Delta(A_i)$ and $r_i := u_i \circ \boldsymbol{r}_i$. Each X_i is compact and convex, and r_i is continuous. By Glicksberg [1952], $\mathcal{C}$ admits a mixed-strategy Nash equilibrium μ^*. By Theorem 14, $\psi(\mu^*)$ is a hierarchical Nash equilibrium of $\mathcal{G}$.

2. Existence of a mixed-strategy Nash equilibrium under quasiconcavity.
If each u_i is quasiconcave, then $r_i(\cdot, x_{-i}) = u_i \circ \boldsymbol{r}_i(\cdot, x_{-i})$ is quasiconcave in x_i because $\boldsymbol{r}_i$ is affine in x_i. By the Debreu-Glicksberg-Fan theorem (see, e.g. Fudenberg and Tirole [1991]), $\mathcal{C}$ admits a pure-strategy Nash equilibrium $x^* \in \prod_i \Delta(A_i)$. Theorem 13 allows us to interpret each $x_i^* \in \Delta(A_i)$ as a mixed strategy in $\mathcal{G}$, yielding a mixed-strategy Nash equilibrium of $\mathcal{G}$. □

5 Special Cases

One drawback of pure-strategy equivalence is that it requires working with a continuous strategy space when computing Nash equilibria. In contrast, in single-objective games, tasks such as computing a best response can often be reduced to a simple enumeration of actions followed by selecting the one with the highest expected utility. In this section we show that, under certain conditions, some of this simplicity can be recovered, or conversely, that such simplification is not possible.

To this end, we introduce the *induced normal-form game* $\mathcal{G}^u$ associated with an MONFG $\mathcal{G} = \langle N, \mathcal{A}, \boldsymbol{r} \rangle$ and utility functions $u = (u_i)_{i \in N}$. The induced game is defined as the single-objective game obtained by scalarising the vector payoffs $\boldsymbol{r}$ a priori with the utilities u. Concretely, the payoff function for each player i is given by

$$r_i(a) = u_i\big(\boldsymbol{r}_i(a)\big),$$

so that $\mathcal{G}^u = \langle N, \mathcal{A}, r \rangle$. In line with the terminology established in earlier chapters, the original MONFG is considered under the SER criterion, whereas the induced NFG is defined under the ESR criterion. For computational purposes, we restrict attention from this point onward to pure and mixed strategies over $\mathcal{A}$. Hierarchical strategies are not considered further.

5.1 Nash Sets Under Different Games

The following theorem is easy to demonstrate by a variety of examples:

Theorem 15: Nash sets under SER vs ESR

There exists a finite MONFG $\mathcal{G} = \langle N, \mathcal{A}, \boldsymbol{r} \rangle$ with utility functions $u = (u_i)_{i \in N}$ such that for its set of Nash equilibria $\mathcal{N}^{\mathcal{G}}$ and the set of Nash equilibria in the induced NFG $\mathcal{N}^{\mathcal{G}^u}$, we have:

1. $|\mathcal{N}^{\mathcal{G}}| \neq |\mathcal{N}^{\mathcal{G}^u}|$;
2. $\mathcal{N}^{\mathcal{G}} \cap \mathcal{N}^{\mathcal{G}^u} = \varnothing$.

Proof. **1. MONFG and utility construction.**
Consider the 2×2 MONFG in Fig. 5.5a (same vector payoffs for both players). Let the utility functions be

$$u_1(p_1, p_2) = u_2(p_1, p_2) = 0.1\, p_1 + \max\{0, p_1\} \max\{0, p_2\}.$$

Applying u cellwise yields the induced NFG shown in Fig. 5.5b.

	A	B
A	$(1,\ 0);\ (1,\ 0)$	$(0,\ 1);\ (0,\ 1)$
B	$(0,\ 1);\ (0,\ 1)$	$(-10,\ 0);\ (-10,\ 0)$

(a) MONFG vector payoffs (SER).

	A	B
A	0.1; 0.1	0; 0
B	0; 0	-1; -1

(b) Induced NFG utilities (ESR).

Figure 5.5: An MONFG and its induced single-objective game.

2. Nash equilibria in $\mathcal{G}$ and $\mathcal{G}^u$.
Induced NFG $\mathcal{G}^u$ (ESR). From Fig. 5.5b, best responses are A against both A and B, so the unique Nash equilibrium is (A, A).

MONFG $\mathcal{G}$ (SER). Let x denote the probability with which the row player plays A (and hence $(1 - x)$ the probability of playing B). If the column player plays A deterministically, the row player's best response is obtained by maximising the following function:

$$f(x) = 0.1\,x + \max\{0, x\}\,\max\{0, 1 - x\} \;=\; 1.1x - x^2, \qquad 0 \leq x \leq 1.$$

Since f is strictly concave, it obtains a unique maximum at $x^* = \frac{11}{20}$. As such (A, A) is not a Nash equilibrium under SER. Moreover, at x^*, the column player's best response is to play A with probability 1, since this maximises their expected payoff against the row player's mixture. Consequently, $(x^*, 1)$ forms a Nash equilibrium, and by symmetry, $(1, x^*)$ is also a Nash equilibrium.

3. The two properties.
The induced NFG $\mathcal{G}^u$ has the unique equilibrium (A, A); the MONFG $\mathcal{G}$ has at least the two mixed equilibria above and (A, A) is not an equilibrium. Hence

$$\left|\mathcal{N}^{\mathcal{G}}\right| \neq \left|\mathcal{N}^{\mathcal{G}^u}\right| \qquad \text{and} \qquad \mathcal{N}^{\mathcal{G}} \cap \mathcal{N}^{\mathcal{G}^u} = \varnothing. \qquad \square$$

Intuitively, this result demonstrates that in general, we cannot solve the much simpler $\mathcal{G}^u$ and reuse the Nash equilibria in $\mathcal{G}$. The main obstacle for this approach is the expectation over returns. One type of strategy that does not require this is a pure strategy, which deterministically plays a specific joint action and therefore is much more straightforward to analyse. We contribute the following result for this special case:

Theorem 16: Relating pure-strategy Nash equilibria in $\mathcal{G}$ and $\mathcal{G}^u$

Let $\mathcal{G} = \langle N, \mathcal{A}, \boldsymbol{r} \rangle$ be a finite MONFG with utility functions $u = (u_i)_{i \in N}$ and let $\mathcal{G}^u = \langle N, \mathcal{A}, r \rangle$ denote its induced single-objective NFG. Then:

1. If $a^* \in \mathcal{A}$ is a pure-strategy Nash equilibrium of $\mathcal{G}$, then a^* is a pure-strategy Nash equilibrium of $\mathcal{G}^u$.
2. The converse fails in general: there exist games where a pure-strategy Nash equilibrium of $\mathcal{G}^u$ is not a pure-strategy Nash equilibrium of $\mathcal{G}$.
3. If each u_i is quasiconvex, then every pure-strategy Nash equilibrium of $\mathcal{G}^u$ is also a pure-strategy Nash equilibrium of $\mathcal{G}$.

Proof. **1. Induced NFG and evaluation at pure profiles.**
By definition, $\mathcal{G}^u$ evaluates each action profile $a \in \mathcal{A}$ via the scalar payoff $u_i\big(\boldsymbol{r}_i(a)\big)$. In $\mathcal{G}$ under SER, for any (possibly mixed) deviation the payoff is $u_i\big(\mathbb{E}\left[\boldsymbol{r}_i(\cdot)\right]\big)$. At a *pure* profile a, the expectation is trivial, so

$$u_i\big(\boldsymbol{r}_i(a)\big) \;=\; u_i\big(\mathbb{E}\left[\boldsymbol{r}_i(a)\right]\big) \qquad \forall i \in N. \tag{1}$$

2. From $\mathcal{G}$ (SER) to $\mathcal{G}^u$ (ESR).
Suppose a^* is a pure-strategy Nash equilibrium of $\mathcal{G}$ (SER). Then, for every player i and every pure deviation $a_i \in A_i$,

$$u_i\big(\mathbb{E}\left[\boldsymbol{r}_i(a_i^*, a_{-i}^*)\right]\big) \;\geq\; u_i\big(\mathbb{E}\left[\boldsymbol{r}_i(a_i, a_{-i}^*)\right]\big).$$

By (1), this is exactly

$$u_i\big(\boldsymbol{r}_i(a_i^*, a_{-i}^*)\big) \;\geq\; u_i\big(\boldsymbol{r}_i(a_i, a_{-i}^*)\big),$$

which is the best-response condition in $\mathcal{G}^u$. Hence a^* is a pure-strategy Nash equilibrium of $\mathcal{G}^u$.

3. Counterexample for $\mathcal{G}^u$ to $\mathcal{G}$ in general.
Theorem 15 provides a concrete 2×2 construction where (A, A) is a pure-strategy Nash equilibrium in the induced NFG (ESR) but is not a pure-strategy Nash equilibrium in the original MONFG (SER).

4. Quasiconvex sufficiency for $\mathcal{G}^u$ to $\mathcal{G}$.
Assume now that each u_i is quasiconvex and that a^* is a pure-strategy Nash equilibrium of $\mathcal{G}^u$. Then for every player i and every pure $a_i \in A_i$,

$$u_i\big(\boldsymbol{r}_i(a_i^*, a_{-i}^*)\big) \;\geq\; u_i\big(\boldsymbol{r}_i(a_i, a_{-i}^*)\big).$$

Let $\pi_i \in \Delta(A_i)$ be any mixed deviation for player i in the SER game. Then

$$\mathbb{E}\left[\boldsymbol{r}_i(\pi_i, a^*_{-i})\right] = \sum_j \pi_i(a_i^j)\, \boldsymbol{r}_i\big(a_i^j, a^*_{-i}\big).$$

By quasiconvexity of u_i,

$$u_i\Big(\ \mathbb{E}\left[\boldsymbol{r}_i(\pi_i, a^*_{-i})\right]\ \Big) \ \leq\ \max_j u_i\big(\boldsymbol{r}_i(a_i^j, a^*_{-i})\big) \ \leq\ u_i\big(\boldsymbol{r}_i(a_i^*, a^*_{-i})\big) = u_i\Big(\ \mathbb{E}\left[\boldsymbol{r}_i(a_i^*, a^*_{-i})\right]\ \Big),$$

where the last equality uses (1). Hence no mixed (and therefore no pure) deviation is profitable in the SER game, so a^* is a pure-strategy Nash equilibrium of $\mathcal{G}$.

5. Conclusion.
Step 1 holds always; Step 2 shows the converse fails without further assumptions; Step 3 establishes quasiconvexity of u_i as a sufficient condition for equivalence of pure-strategy equilibria between $\mathcal{G}^u$ and $\mathcal{G}$. □

5.2 Blended Games

Additionally, we consider *blended* games, which are games where a subset of players is playing the MONFG $\mathcal{G}$ with utility functions u (SER players) and the remaining players are playing the induced NFG $\mathcal{G}^u$ (ESR players). We define this below and subsequently show in Corollary 5 that pure-strategy Nash equilibria in the blended game follow the same properties as in Theorem 16.

Definition 34: Blended game

Let $\mathcal{G} = \langle N, \mathcal{A}, \boldsymbol{r}\rangle$ be a finite MONFG with utility functions $u = (u_i)_{i\in N}$. A blended game $\mathcal{B}^u = \langle N, \mathcal{A}, \boldsymbol{r}, \mathcal{I}, u\rangle$ is a game where the utility for a mixed strategy $\pi \in \Delta(\mathcal{A})$ is defined as follows:

$$u_i(\pi) \ =\ \begin{cases} u_i\big(\mathbb{E}_{a\sim\pi}\left[\boldsymbol{r}_i(a)\right]\big), & \text{if } i \in \mathcal{I} \quad \text{(SER player)},\\ \mathbb{E}_{a\sim\pi}\left[u_i\big(\boldsymbol{r}_i(a)\big)\right], & \text{if } i \notin \mathcal{I} \quad \text{(ESR player)}. \end{cases}$$

Corollary 5: Pure-strategy Nash equilibria in blended games

Let $\mathcal{G} = \langle N, \mathcal{A}, \boldsymbol{r} \rangle$ be a finite MONFG with utility functions $u = (u_i)_{i \in N}$, $\mathcal{G}^u$ its induced NFG and $\mathcal{B}^u$ a blended game. Then,

1. If a is a pure-strategy Nash equilibrium in $\mathcal{G}$, then it is also a pure-strategy Nash equilibrium in any blended game.
2. If each u_i is quasiconvex and a is a pure-strategy Nash equilibrium in $\mathcal{G}^u$, then it is also a pure-strategy Nash equilibrium in any blended game.

Proof. If a is a Nash equilibrium in $\mathcal{G}$, it is also a Nash equilibrium in $\mathcal{G}^u$ by Theorem 16. By the definition of a blended game, this immediately guarantees that a is also a Nash equilibrium in the blended game. The result for quasiconvex utility functions follows similarly. □

5.3 Algorithmic Implications

We give a simple procedure to compute all pure-strategy Nash equilibria (PSNE) of a finite n-player MONFG with quasiconvex utility functions. The construction operates via the induced NFG $\mathcal{G}^u$ and, by Theorem 16 and Corollary 5, yields PSNE for SER and for any blended setting as well. We show the algorithm in Algorithm 11.

Algorithm 11: Computing all PSNE in an MONFG

Input: An MONFG $\mathcal{G} = \langle N, \mathcal{A}, \boldsymbol{r} \rangle$ and quasiconvex utilities $u = (u_i)_{i \in N}$
Output: The set $\mathcal{P}$ of all pure-strategy Nash equilibria

$\mathcal{G}^u \leftarrow \langle N, \mathcal{A}, u \circ \boldsymbol{r} \rangle$
$\mathcal{P} \leftarrow \varnothing$
for $a \in \mathcal{A}$ **do**
    if *a is a PSNE of $\mathcal{G}^u$* **then**
        $\mathcal{P} \leftarrow \mathcal{P} \cup \{a\}$
return $\mathcal{P}$

Complexity and practicalities. The reduction preserves the game size; checking whether a pure profile is a Nash equilibrium in $\mathcal{G}^u$ amounts to a constant-time comparison per player-deviation once payoffs are tabulated. The dependence on the number of objectives d arises only in evaluating $u_i(\boldsymbol{r}_i(a))$ for each profile a; this adds a modest scalarisation cost, so the problem is essentially as hard as solving a single-objective NFG of the same action size. The naive enumeration in Algorithm 11 can be replaced by faster

routines for all-PSNE computation in NFGs (e.g. regret-based filtering [Corley, 2020]). Moreover, since all checks can be evaluated independently, parallel evaluation of profiles can further reduce runtime.

6 Empirical Results

We provide empirical evidence for the provided theorems and show that they can also be applied to compute approximate equilibria when the strategy sets do not satisfy the necessary conditions for pure-strategy equivalence. We adapt the well-known fictitious play algorithm from single-objective games to multi-objective games and use it to compute pure-strategy Nash equilibria in continuous games. The results empirically demonstrate the applicability of our contributions and may serve as a useful template for future applications.

6.1 Multi-Objective Fictitious Play

In Algorithm 12, we show an extension for fictitious play to multi-objective games. For simplicity, we consider a two-player variant but this can trivially be extended to n-player games. In each iteration of the algorithm, players calculate the empirical strategy of their opponent based on their history of play and compute a best response to this strategy. Players subsequently sample an action from their new strategy and update their histories.

Comparison to single-objective fictitious play. The fictitious play algorithm shown above appears identical to the original fictitious play algorithm [Robinson, 1951]. The exception, however, lies in the best response computation steps. In single-objective games, this can be done efficiently by selecting the action with the highest expected returns, i.e.

$$BR(A_i, \pi_{-i}, r_i) = \arg\max_{a_i \in A_i} r_i(a_i, \pi_{-i}). \tag{5.7}$$

In multi-objective games, this approach can only be guaranteed to return a correct best response when employing a quasiconvex utility function. In general MONFGs, the best response can be a mixed strategy and thus requires executing an optimisation subroutine to find the strategy generating the maximum utility. As a best response needs to be a global maximum, this requires the use of a global optimisation algorithm. Under specific utility functions or when approximate best responses suffice, a local optimiser could also be used.

Benefits and limitations. Recent work has studied an adaptation of fictitious play to continuous games [Ganzfried, 2021]. In their algorithm, a growing array of past strategies is kept to later compute a best response to, which imposes a significant memory

Algorithm 12: Two-Player Multi-Objective Fictitious Play

Input: A two-player MONFG $G = \langle N, \mathcal{A}, \boldsymbol{r} \rangle$, utility functions u, and horizon T
Output: Mixed strategy profile $\pi = (\pi_1, \pi_2)$

for $i \in \{1, 2\}$ **do**
  Initialise action counts $h_i(a_i) \leftarrow 0$ for all $a_i \in A_i$
for $t = 1$ **to** T **do**
  for $i \in \{1, 2\}$ **do**
    Let $j = 3 - i$
    Define opponent's empirical strategy: $\tilde{\pi}_j(a_j) \leftarrow \frac{h_j(a_j)}{t}$ for all $a_j \in A_j$
    Compute best response:

$$\pi_i \leftarrow \textsc{BestResponse}(\boldsymbol{r}_i, u_i, \tilde{\pi}_j)$$

    Sample action: $a_i \sim \pi_i$
  Play joint action (a_1, a_2)
  Update counts: $h_1(a_1) \leftarrow h_1(a_1) + 1$, $h_2(a_2) \leftarrow h_2(a_2) + 1$
return $\pi = \left(\frac{h_1}{T}, \frac{h_2}{T}\right)$

requirement. A key advantage of our approach is that it only requires an array of fixed length, i.e., one entry per action, where a counter is incremented each time an action is played. The empirical mixed strategy of the opponent is then calculated by taking the relative frequency of each action. A limitation of this approach is that it can only learn pure-strategy equilibria from the continuous game.

6.2 Polynomial Game

Polynomial games are a subset of continuous games, where utility functions are guaranteed to be polynomial functions of the player strategies [Stein et al., 2008, 2011]. We demonstrate that such games can also be represented as an MONFG and may be solved without employing any continuous game or polynomial game specific machinery. We cover a simple example as described by Parrilo [2006].

Consider a zero-sum game where both players select a strategy from the interval $[-1, 1]$. The utility function for player one is defined as,

$$r_1(a, b) = 2ab^2 - a^2 - b, \tag{5.8}$$

with a the strategy selected by player one and b the strategy of player two. As the game is zero-sum, player two's utility is given by $r_2(a, b) = -r_1(a, b)$. The utility functions

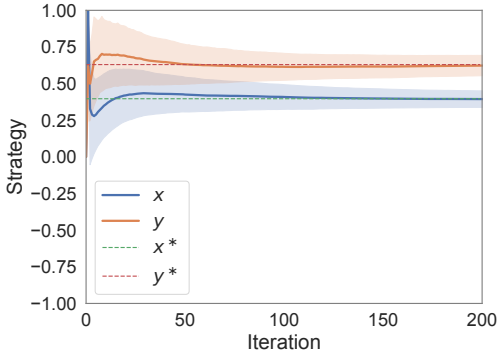


Figure 5.6: Learning curves for the polynomial game.

used in the game guarantee the existence of a unique Nash equilibrium in pure strategies where $a^* = 0.397$ and $b^* = 0.630$.

As the strategy sets are line segments, and thus are 1-simplices, a pure-strategy equivalent multi-objective game is guaranteed to exist. To complete the transformation from the polynomial game to a multi-objective game, a strategy bijection $\varphi_i : X_i \to \Delta(A_i)$ is required for each player i. The strategy bijection for both players is given by,

$$\varphi_i(x_i) = \left(\frac{x_i - x_{i,\min}}{x_{i,\max} - x_{i,\min}}, 1 - \frac{x_i - x_{i,\min}}{x_{i,\max} - x_{i,\min}} \right), \tag{5.9}$$

where $x_{i,\min} = -1$ and $x_{i,\max} = 1$ for both players. The inverse strategy bijection is given by,

$$\varphi_i^{-1}(\pi_i) = x_{i,\min} + \pi_{i,0} \cdot \left(x_{i,\max} - x_{i,\min} \right). \tag{5.10}$$

The final multi-objective game thus has two players with two actions each and corresponding identity game payoffs. Furthermore, the utility functions for both players are $u_i = r_i \circ \varphi_i^{-1}$. Because the original utility functions r_1 and r_2 guarantee a pure-strategy Nash equilibrium in the continuous game, fictitious play is well suited to find the mixed strategy equilibrium in the MONFG.

We execute the fictitious play algorithm for 200 iterations on the constructed multi-objective game and repeat this for 1000 trials. Figure 5.6 shows the learned strategies over time, with the shaded area denoting the standard deviation at that time. We illustrate the Nash equilibrium $(0.397, 0.630)$ with dotted lines. It is clear that our algorithm learns the equilibrium after approximately 100 iterations and is able to keep improving its strategies closer to the exact equilibrium over time.

6.3 Bertrand Price Game

Theorem 11 states that a pure-strategy equivalent MONFG is only guaranteed to exist for continuous games whose strategy spaces are convex subsets of a Euclidean space. We demonstrate that pure-strategy equivalence can still be applied when this condition is not met by approximating the continuous game. We illustrate this using the Bertrand price game characterised by Judd et al. [2012].

Bertrand price games have been extensively studied as an economic model for determining prices in competitive settings [Varian, 2014]. In this example, we consider two firms, x and y, which respectively produce a different good for price p_x and p_y. There are three types of customers, which have a distinct demand for both goods. The first type of customer has linear demand curves $d_{x,1}$ and $d_{y,1}$ and only wants the good from firm x,

$$d_{x,1}(p_x, p_y) = a - p_x \qquad d_{y,1}(p_x, p_y) = 0, \tag{5.11}$$

with a signifying all factors, other than price, which influence the demand. The demand function for the third type of customer is defined analogously for the good of firm y,

$$d_{x,3}(p_x, p_y) = 0 \qquad d_{y,3}(p_x, p_y) = a - p_y. \tag{5.12}$$

Finally, the second type of customer has a demand for both goods,

$$d_{x,2}(p_x, p_y) = n \cdot p_x^{-\sigma} \left(p_x^{1-\sigma} + p_y^{1-\sigma}\right)^{(\gamma-\sigma)/(-1+\sigma)} \tag{5.13}$$

$$d_{y,2}(p_x, p_y) = n \cdot p_y^{-\sigma} \left(p_y^{1-\sigma} + p_x^{1-\sigma}\right)^{(\gamma-\sigma)/(-1+\sigma)}. \tag{5.14}$$

with n the number of type two customers, σ the elasticity of substitution between x and y and γ the elasticity of demand for the composite good. The total demand for each good, respectively d_x and d_y, is given by summing the individual demands for each type. Finally, let m be the unit cost of production for each firm, then the profit for both firms is defined as,

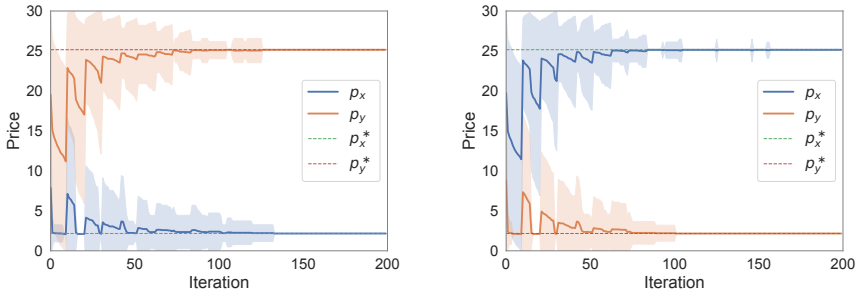
$$r_x(p_x, p_y) = (p_x - m) \cdot d_x(p_x, p_y) \tag{5.15}$$

$$r_y(p_x, p_y) = \left(p_y - m\right) \cdot d_y(p_x, p_y). \tag{5.16}$$

The range of possible prices considered in the game is in the open interval $(0, +\infty)$. As such, strategy spaces in the continuous game are non-compact, thus violating a necessary condition for pure-strategy equivalence. We can resolve this, however, by making compact convex approximations of the strategy spaces and using these instead. We do this by constraining prices to be in a closed interval $[p_{\min}, p_{\max}]$, which ensures that strategy sets are 1-simplices as in the previous example. Because of this approximation, we may reuse the same strategy bijection as defined in Eqs. (5.9) and (5.10). Note

Table 5.1: Nash equilibrium strategies and profits; highest profit for each firm highlighted.

p_x	p_y	r_x	r_y
2.168	25.157	**724.337**	608.981
25.157	2.168	608.981	**724.337**



(a) Learning curves for the equilibrium at $(2.168, 25.157)$.

(b) Learning curves for the equilibrium at $(25.157, 2.168)$.

Figure 5.7: Learning curves for the Bertrand price game in the interval $[1, 30]$.

that approximating the continuous game by altering strategy sets may remove existing equilibria from reach or introduce new ones. In this particular example, as we are both lower and upper bounding the strategy sets, it is possible that an equilibrium falls outside of the bounds and a new equilibrium is created in the MONFG which is not an equilibrium in the original game.

For the following experiments, we define $\sigma = 3, \gamma = 2, n = 2700, m = 1$ and $a = 50$. With these parameters, the price game has two distinct Nash equilibria, shown in Table 5.1.

Suitable Approximation

To give both equilibria a chance of being selected, we set $p_{\min} = 1$ and $p_{\max} = 30$. Every execution of the fictitious play algorithm is run for 200 iterations and results are averaged over 1000 trials as in the previous section.

In Fig. 5.7, we show the trajectories leading to the two equilibria. In earlier episodes, the trajectories are non-smooth and show high standard deviation, as the best response computation from a limited history of play leads agents to change strategies rapidly. However, once beliefs converge after approximately 150 episodes, a Nash equilibrium is

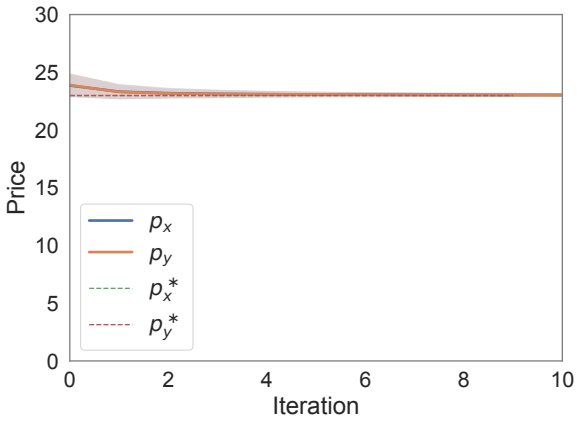


Figure 5.8: Learning curves for the Bertrand price game in the interval $[4, 30]$.

consistently played. We also find that the learning trajectories for both equilibria are similar, showing that the individual trajectory that is followed is mostly determined by randomisation early on in the learning process. These results demonstrate that multi-objective algorithms can be applied even to approximations of continuous games, given that these approximations sufficiently capture the original game.

Unsuitable Approximation

Next, we consider what happens when an unsuitable approximation of the strategy sets is used. Specifically, we raise the minimum price to 4, which renders both equilibria from Table 5.1 impossible. Intuitively, these equilibria had one player who opted for a mass-market strategy with lower prices and another which opted for a niche strategy with higher prices. By raising the minimum price, we render this mass-market strategy impossible. We show the resulting learning trajectories for this experiment in Fig. 5.8.

Across all trials, players quickly converge to the joint strategy $(22.987, 22.987)$, yielding a profit of 673.38 for each player. This outcome is a Nash equilibrium of the MONFG, although it is not an equilibrium of the original continuous game, which highlights the limitations of the approximation technique. Nevertheless, the induced equilibrium in the MONFG produces higher social welfare. It yields a total profit of 1346.754, compared with 1333.318 for the equilibria reported in Table 5.1, and also secures a higher minimum profit across players. Hence, even when the approximation is imperfect, the resulting equilibria may still provide valuable solutions, for instance from a mechanism design perspective.

7 Conclusion

This chapter addressed the central question of how equilibrium solutions with multiple objectives can be characterised and how they relate to classical game-theoretic models. To this end, we introduced pure-strategy equivalence, a novel equivalence relation between continuous games and multi-objective normal-form games (MONFGs). We proved that whenever the strategy spaces of a continuous game are convex subsets of Euclidean space, one can construct a pure-strategy equivalent MONFG, and conversely. Crucially, this equivalence preserves Nash equilibria, thereby establishing a rigorous bridge between multi-objective and classical models.

This bridge has both theoretical and algorithmic implications. Theoretically, it extends the well-established practice of reducing more complex game models to normal-form games [Maschler et al., 2013] into the multi-objective setting, thereby opening the door to equivalence results for a wider class of games with infinitely many strategies. Algorithmically, it provides a pathway for solving continuous games by exploiting the discrete, tabular structure of MONFGs, which can be more amenable to computation. Because the equivalence is not unique, a promising direction lies in identifying representations that are strategically faithful yet computationally efficient.

We also explored important special cases that reveal how equilibria in MONFGs can sometimes be computed through simpler induced normal-form games rather than through the full MONFG, which is often computationally intractable. Our experiments confirmed the practical relevance of these insights. By adapting fictitious play for continuous games to the MONFG setting, we demonstrated that equilibria can be computed effectively in polynomial and Bertrand price games, thus showing the concrete value of our theoretical framework.

Follow-up challenges. While we have taken significant steps towards characterising equilibria with multiple objectives through pure-strategy equivalence, several open challenges remain.

- **Interdependent utility functions.** Our existence guarantees rely on quasiconcavity of utility functions, which is a sufficient but not a necessary condition. The broader difficulty lies in the interdependence of players' utilities, which makes general equilibrium existence results elusive. Future work could explore measures of functional agreement and their effect on equilibrium existence [Rosen, 1965; Mannion and Radulescu, 2023].
- **Representation learning for MONFGs.** Computing equilibria in MONFGs is computationally demanding. Recent advances such as the NfgTransformer [Liu et al., 2024] show that deep learning can amortise this cost by learning structured representations of games. Through task-specific decoders, they demonstrate the ability to approximate Nash equilibria or estimate best responses. Whether such

approaches can be adapted effectively to the multi-objective setting remains an open and promising direction.

- **Learning strategy bijectors.** The forward mapping from an MONFG to a continuous game is straightforward, but the inverse requires explicit bijections that are difficult to construct by hand. Learning such mappings automatically would greatly enhance the practical applicability of pure-strategy equivalence.

6

Conclusion

1 Summary of Contributions

Artificial agents that operate in the real world must contend with uncertainty, multiple and often conflicting objectives, and interaction with other agents. If such systems are to be deployed responsibly, these challenges must be addressed with methods that are both effective and principled. This thesis develops a coherent path towards that goal by *connecting single-objective theory and algorithms to multi-objective decision-making.* We organised the work around the following guiding question:

How can we develop principled methods for multi-objective decision-making and compute solutions that balance trade-offs in line with decision-makers' preferences by leveraging advances from single-objective problems?

The thesis answers this question across three parts, each combining formal analysis with algorithms, and each making the single-objective → multi-objective connection explicit.

From single-objective RL to Pareto fronts (Chapter 3). We study decision-makers whose utility functions are monotonically increasing and computed from the expected returns of a policy, for whom the *Pareto front* is the central solution concept. We show that learning the policies on the Pareto front can be reduced to solving a sequence of constrained single-objective problems. This reduction yields *Iterated Pareto Referent Optimisation* (IPRO), an anytime method that repeatedly invokes single-objective

solvers. IPRO converges to an ε-approximation of the Pareto front in finitely many iterations and to the exact front in the limit, with explicit quality bounds at each step. We also obtain complexity guarantees in terms of the number of subproblems required for a given accuracy. Empirically, IPRO matches or outperforms prior work while requiring less domain knowledge and transferring readily to settings such as planning and multi-objective pathfinding. A key advantage of IPRO is its modularity, so advances in single-objective RL can be plugged in to immediately improve multi-objective performance. Moreover, when the single-objective solvers used within IPRO come with guarantees such as optimality or sample-complexity bounds, these guarantees carry over through our reduction and yield corresponding end-to-end guarantees for the learned Pareto front.

Beyond expectations: distributional solution sets (Chapter 4). For many applications, the expected return is an inadequate proxy for policy quality. Building on distributional RL, we introduce solution sets that retain *return distributions* rather than only their means. We define the *Distributional Undominated Set* (DUS) and its convex variant (CDUS), and prove that the CDUS coincides with the set of policies that maximise expected utility for multivariate risk-averse decision-makers. We then clarify how these sets relate to each other: (i) the DUS contains both the Pareto front and CDUS; (ii) the Pareto front and CDUS are generally incomparable; and (iii) both sets contain the convex coverage set, the solution concept sufficient for all linear utility functions. We provide pruning procedures that reduce an arbitrary collection of distributions to a (C)DUS and extend Q-learning to handle distributional information and recover a DUS in practice. In decision support settings, we propose a *learn large, prune later* strategy: first recover a larger candidate set, such as a DUS, then prune it, for example, to a Pareto front or CDUS, as preference information becomes available. Empirically, the pruned subsets are much smaller while preserving decision quality, which makes them easier to use. Additionally, by preserving distributional information, the learned sets remain flexible and can be tailored to individual preferences that are unknown at training time.

Bridging multi- and single-objective games (Chapter 5). We study strategic interaction with multiple objectives and establish a formal equivalence between a broad class of multi-objective games with fixed utilities and a family of continuous single-objective games. The mapping interprets each mixed strategy in the multi-objective game as a pure strategy in the continuous single-objective counterpart while preserving payoffs, which we call *pure-strategy equivalence*. We extend the correspondence to mixed strategies on the single-objective side by introducing *hierarchical strategies* in the multi-objective model, yielding a full equilibrium mapping. This equivalence explains several known pathologies of multi-objective games and, more importantly, allows existence results and computational techniques for Nash equilibria to transfer. Under standard compactness and continuity assumptions, we obtain general existence guarantees. The

equivalence works in both directions: theoretical results from single-objective games carry over to the multi-objective setting, while structural features of the multi-objective formulation make certain continuous games easier to analyse and compute. We illustrate this on two well-studied continuous games, where the equivalence combined with simple fictitious play converges to a Nash equilibrium in practice. More broadly, this chapter shows how multi-objective multi-agent problems can be reframed so that mature single-objective theory and algorithms become directly applicable.

Take-away message. A unifying theme runs through all three parts: carefully constructed links to single-objective methods reveal that multi-objective decision-making is not an isolated endeavour but can be connected to well-understood single-objective principles. In the single-agent case, this yields principled Pareto front learning and distributional solution sets that better reflect decision-makers' preferences. In the multi-agent case, it provides intuitions for the observed limitations and delivers equivalences that carry existence theorems and algorithms across modelling choices. Together, these results show how multi-objective RL can *bootstrap* from the mature single-objective literature to achieve both theoretical clarity and practical effectiveness.

2 The Road Ahead

This thesis examined how to train agents that balance multiple objectives by integrating ideas and tools from single-objective methods. While the results clarify several foundations and deliver practical algorithms, important open questions remain. We group them into *conceptual*, *theoretical*, and *algorithmic* challenges, and highlight two directions in each.

Conceptual Challenges

A common language. Adjacent communities, including reinforcement learning, operations research, economics, decision analysis, and control, often study closely related problems but use different terminology and notation. Even basic notions differ: what one field calls "objectives" may appear as *criteria* [Gábor et al., 1998; Greco et al., 2016] or *attributes* [Dyer, 2005], and "utility" is defined and used differently across traditions [see Keeney and Raiffa, 1993, p. 150]. These discrepancies hinder the transfer of results, obscure equivalences, and encourage reinvention. The problem is arguably worse in theoretical work, where not only the terms but also the underlying assumptions, modelling frameworks, and even notation differ. Establishing a shared vocabulary for the same underlying objects, while still recognising the real and important differences between these fields, would make cross-pollination far more routine.

The problem with reward functions. Standard RL assumes rewards are well-defined and stationary. In practice, they are often underspecified or non-stationary, and in some settings not available at all. In such cases, the agent's optimisation target must be inferred from demonstrations or feedback [Christiano et al., 2017; Wirth et al., 2017], negotiated among stakeholders [Conitzer et al., 2024], or specified at inference time through goal- or preference conditioning [Hartikainen et al., 2020; Touati and Ollivier, 2021]. Crucially, this challenge is not limited to the scalar case: multi-objective formulations still require the underlying objectives and their semantics to be known in advance. When these are misspecified, agents can optimise the wrong proxy and exhibit harmful behaviour, including reward hacking and other failure modes [Muslimani et al., 2025]. The challenge, then, is to rely less on fixed reward specifications and instead develop methods that *infer*, *elicit*, and *adapt* objectives over time while remaining robust to misspecification. We view this primarily as a conceptual agenda, since it remains unclear how best to address it. The most promising approaches are likely to be domain-dependent and draw on a wide range of ideas from related fields.

Theoretical Challenges

A taxonomy of solution sets. Different decision-makers warrant different solution concepts. Even within this thesis we analysed several, most notably the Pareto front, the convex coverage set, and the distributional undominated set with its convex variant, because each captures a distinct notion of optimality. A compelling theoretical agenda is to carve out *well-structured preference classes* and to characterise the corresponding solution sets. This means mapping preference models to solution concepts and analysing their structural properties (convexity, smoothness, dimensionality), order relations (inclusions, separations, equivalences), and stability (sensitivity to dynamics, noise, and elicitation error). Such structure is useful since it yields smaller, tailored sets that are easier to work with in decision support scenarios.

Preferences, environments, and optimal policies. Optimality of policies is shaped jointly by the environment's dynamics and the decision-maker's preferences. A natural perspective is to treat the preference vector as a parameter and analyse how value functions and optimal policies vary with it. For example, if a policy is optimal for a given preference vector and the environment satisfies certain regularity conditions, can we bound the loss for neighbouring preferences or provide continuity and sensitivity results for the optimal policy map? By treating a preference class as inducing a family of MDPs, the problem aligns with policy transfer across MDPs [Gleave et al., 2021; Fu et al., 2023], which offers a starting point for further analysis.

Algorithmic Challenges

The last mile: interactive deployment. Multi-objective reinforcement learning rarely concludes with the delivery of a single policy. In practice, agents must incorporate online preference elicitation and provide decision support rather than full automation. Tight integration of learning with *interactive* choice models is still underdeveloped for sequential problems, with some notable exceptions [Zintgraf et al., 2018; Shao et al., 2023]. This aspect is crucial, however, since the explicit goal in MORL is to serve decision-makers in accomplishing their desired balance across objectives.

Computational efficiency and reuse across preferences. Many MORL methods are computationally demanding, and our own contributions are no exception. IPRO (Chapter 3) requires numerous environment interactions to construct a Pareto front, while DIMOQ (Chapter 4) incurs substantial memory and computational cost. These demands are justified by their respective advantages, IPRO being an anytime algorithm and DIMOQ yielding higher-utility solutions for decision-makers, but further efficiency gains would greatly enhance their practical appeal. Progress will depend on reusing experience and computation across nearby preferences, for instance through successor features for rapid transfer [Barreto et al., 2017] or amortised policies and value functions conditioned on preferences [Schaul et al., 2015]. Scalable approximations to solution sets and distributed training offer additional avenues for reducing computational overhead.

3 Closing Remarks

Single-objective methods provide more than inspiration for multi-objective reinforcement learning: they offer a practical pathway to principled and effective algorithms. By making the connections explicit, we can carry theory, algorithms, and intuitions across problem classes, and then refine them to accommodate heterogeneous preferences. The agenda ahead combines conceptual, theoretical, and algorithmic challenges. Pursuing these strands together promises multi-objective systems that are both principled and usable in the settings where trade-offs matter most. Ultimately, this is what will enable agents to truly *think in trade-offs*.

Bibliography

Abdolmaleki, A., S. Huang, L. Hasenclever, M. Neunert, F. Song, M. Zambelli, M. Martins, N. Heess, R. Hadsell, and M. Riedmiller
2020. A distributional view on multi-objective policy optimization. In *Proceedings of the 37th International Conference on Machine Learning*, H. D. III and A. Singh, eds., volume 119, Pp. 11–22. PMLR. Cited on page 89.

Abel, D., W. Dabney, A. Harutyunyan, M. K. Ho, M. L. Littman, D. Precup, and S. Singh
2021. On the expressivity of markov reward. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, Virtual*, M. Ranzato, A. Beygelzimer, Y. N. Dauphin, P. Liang, and J. W. Vaughan, eds., Pp. 7799–7812. Cited on page 17.

Abels, A., D. M. Roijers, T. Lenaerts, A. Nowé, and D. Steckelmacher
2019. Dynamic Weights in Multi-Objective Deep Reinforcement Learning. In *Proceedings of the 36th International Conference on Machine Learning*, K. Chaudhuri and R. Salakhutdinov, eds., volume 97, Pp. 11–20, Long Beach, California, USA. PMLR. Cited on pages 53, 93, and 96.

Achiam, J., D. Held, A. Tamar, and P. Abbeel
2017. Constrained policy optimization. In *Proceedings of the 34th International Conference on Machine Learning*, D. Precup and Y. W. Teh, eds., volume 70 of *Proceedings of Machine Learning Research*, Pp. 22–31. PMLR. Cited on page 91.

Adam, L., R. Horčík, T. Kasl, and T. Kroupa
2021. Double Oracle Algorithm for Computing Equilibria in Continuous Games. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(6):5070–5077. Cited on page 55.

Albrecht, S. V., F. Christianos, and L. Schäfer
2024. *Multi-Agent Reinforcement Learning: Foundations and Modern Approaches*. MIT Press. Cited on page 18.

Alegre, L. N., D. M. Roijers, A. Nowé, A. L. C. Bazzan, and B. C. da Silva
2023. Sample-efficient multi-objective learning via generalized policy improvement prioritization. In *Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. Cited on pages 18, 46, 53, 62, 63, and 95.

Altman, E.
1999. *Constrained Markov Decision Processes*, 1 edition. Boca Raton: Routledge. Cited on pages 68, 90, and 91.

Amodei, D., C. Olah, J. Steinhardt, P. F. Christiano, J. Schulman, and D. Mané
2016. Concrete problems in AI safety. *CoRR*, abs/1606.06565. Cited on page 19.

André, F. J. and L. Riesgo
2007. A non-interactive elicitation method for non-linear multiattribute utility functions: Theory and application to agricultural economics. *European Journal of Operational Research*, 181(2):793–807. Cited on page 47.

Andrychowicz, M., B. Baker, M. Chociej, R. Józefowicz, B. McGrew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray, J. Schneider, S. Sidor, J. Tobin, P. Welinder, L. Weng, and W. Zaremba
2020. Learning dexterous in-hand manipulation. *Int. J. Robotics Res.*, 39(1). Cited on page 43.

Anil, R., S. Borgeaud, Y. Wu, J.-B. Alayrac, J. Yu, R. Soricut, J. Schalkwyk, A. M. Dai, A. Hauth, K. Millican, D. Silver, S. Petrov, M. Johnson, I. Antonoglou, J. Schrittwieser, A. Glaese, J. Chen, E. Pitler, T. P. Lillicrap, A. Lazaridou, O. Firat, J. Molloy, M. Isard, P. R. Barham, T. Hennigan, B. Lee, F. Viola, M. Reynolds, Y. Xu, R. Doherty, E. Collins, C. Meyer, E. Rutherford, E. Moreira, K. Ayoub, M. Goel, G. Tucker, E. Piqueras, M. Krikun, I. Barr, N. Savinov, I. Danihelka, B. Roelofs, A. White, A. Andreassen, T. von Glehn, L. Yagati, M. Kazemi, L. Gonzalez, M. Khalman, J. Sygnowski, and e. al.
2023. Gemini: A family of highly capable multimodal models. *CoRR*, abs/2312.11805. Cited on page 15.

Ardizzone, L., J. Kruse, C. Rother, and U. Köthe
2019. Analyzing inverse problems with invertible neural networks. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net. Cited on page 143.

Aumann, R. J.
1974. Subjectivity and correlation in randomized strategies. *Journal of Mathematical Economics*, 1(1):67–96. Cited on pages 19 and 59.

Babichenko, Y., S. Barman, and R. Peretz
2014. Simple approximate equilibria in large games. In *ACM Conference on Economics and Computation, EC '14, Stanford, CA, USA, June 8-12, 2014*, M. Babaioff, V. Conitzer, and D. A. Easley, eds., Pp. 753–770. ACM. Cited on page 56.

Barreto, A., W. Dabney, R. Munos, J. J. Hunt, T. Schaul, D. Silver, and H. van Hasselt
2017. Successor features for transfer in reinforcement learning. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, eds., Pp. 4055–4065. Cited on pages 98 and 165.

Basaklar, T., S. Gumussoy, and U. Ogras
2023. PD-MORL: Preference-driven multi-objective reinforcement learning algorithm. In *The Eleventh International Conference on Learning Representations*. Cited on page 53.

Bawa, V. S., J. N. Bodurtha, M. R. Rao, and H. L. Suri
1985. On determination of stochastic dominance optimal sets. *The Journal of Finance*, 40(2):417–431. Cited on pages 100, 114, and 117.

Behrmann, J., W. Grathwohl, R. T. Q. Chen, D. Duvenaud, and J.-H. Jacobsen
2019. Invertible residual networks. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, K. Chaudhuri and R. Salakhutdinov, eds., volume 97 of *Proceedings of Machine Learning Research*, Pp. 573–582. PMLR. Cited on page 143.

Bellemare, M. G., W. Dabney, and R. Munos
2017. A distributional perspective on reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, Pp. 449–458, Sydney, NSW, Australia. JMLR.org. Cited on page 42.

Bellemare, M. G., W. Dabney, and M. Rowland
2023. *Distributional Reinforcement Learning*. MIT Press. Cited on pages 100, 101, and 113.

Bellemare, M. G., Y. Naddaf, J. Veness, and M. Bowling
2013. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279. Cited on page 41.

Bellman, R. E.
1957. *Dynamic Programming*. Princeton University Press. Cited on page 38.

Bentley, J. L., H. T. Kung, M. Schkolnick, and C. D. Thompson
1978. On the average number of maxima in a set of vectors and applications. *Journal of The Acm*, 25(4):536–543. Cited on page 50.

Bertsekas, D.
2012. *Dynamic Programming and Optimal Control: Volume I*, volume 4. Athena Scientific. Cited on page 34.

Bertsekas, D.
2019. *Reinforcement Learning and Optimal Control*, volume 1. Athena Scientific. Cited on page 34.

Billingsley, P.
1995. *Probability and Measure*, Wiley Series in Probability and Mathematical Statistics, 3rd edition. New York: John Wiley & Sons. Cited on page 28.

Biswas, P., Z. Osika, I. Tamassia, A. Whorra, J. Zatarain-Salazar, J. Kwakkel, F. A. Oliehoek, and P. K. Murukannaiah
2025. Exploring equity of climate policies using multi-agent multi-objective reinforcement learning. Cited on page 25.

Blackwell, D.
1954. An analog of the minimax theorem for vector payoffs. *Pacific Journal of Mathematics*, 6(1):1–8. Cited on pages 57 and 128.

Bleichrodt, H., J. van Rijn, and M. Johannesson
1999. Probability Weighting and Utility Curvature in QALY-Based Decision Making. *Journal of Mathematical Psychology*, 43(2):238–260. Cited on page 47.

Bottou, L.
2010. Large-scale machine learning with stochastic gradient descent. In *19th International Conference on Computational Statistics, COMPSTAT 2010, Paris, France, August 22-27, 2010 - Keynote, Invited and Contributed Papers*, Y. Lechevallier and G. Saporta, eds., Pp. 177–186. Physica-Verlag. Cited on page 41.

Bowling, M., J. D. Martin, D. Abel, and W. Dabney
2023. Settling the reward hypothesis. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, eds., volume 202 of *Proceedings of Machine Learning Research*, Pp. 3003–3020. PMLR. Cited on page 17.

Bradbury, J., R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, and Q. Zhang
2018. JAX: Composable transformations of Python+NumPy programs. Cited on page 15.

Bredon, G. E.
1993. General Topology. In *Topology and Geometry*, G. E. Bredon, ed., Pp. 1–62. New York, NY: Springer New York. Cited on pages 134 and 143.

Cai, X.-Q., P. Zhang, L. Zhao, J. Bian, M. Sugiyama, and A. J. Llorens
2023. Distributional pareto-optimal multi-objective reinforcement learning. In *Thirty-Seventh Conference on Neural Information Processing Systems*. Cited on page 105.

Castelletti, A., F. Pianosi, and M. Restelli
2013. A multiobjective reinforcement learning approach to water resources systems operation: Pareto frontier approximation in a single run. *Water Resources Research*, 49(6):3476–3486. Cited on page 61.

Cesa-Bianchi, N. and G. Lugosi
2006. *Prediction, Learning, and Games*. Cambridge: Cambridge University Press. Cited on page 56.

Chatterjee, K., R. Majumdar, and T. A. Henzinger
2006. Markov decision processes with multiple objectives. In *STACS 2006*, B. Durand and W. Thomas, eds., Pp. 325–336, Berlin, Heidelberg. Springer Berlin Heidelberg. Cited on pages 67, 77, 91, and 92.

Christiano, P. F., J. Leike, T. Brown, M. Martic, S. Legg, and D. Amodei
2017. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds., volume 30. Curran Associates, Inc. Cited on page 164.

Codenotti, B. and K. Varadarajan
2007. Computation of market equilibria by convex programming. In *Algorithmic Game Theory*, N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani, eds., Pp. 135–158. Cambridge: Cambridge University Press. Cited on page 123.

Conitzer, V., R. Freedman, J. Heitzig, W. H. Holliday, B. M. Jacobs, N. Lambert, M. Mossé, E. Pacuit, S. Russell, H. Schoelkopf, E. Tewolde, and W. S. Zwicker
2024. Position: Social choice should guide AI alignment in dealing with diverse human feedback. In *Forty-First International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net. Cited on page 164.

Corley, H. W.
1985. Games with vector payoffs. *Journal of Optimization Theory and Applications*, 47(4):491–498. Cited on page 58.

Corley, H. W.
2020. A Regret-Based Algorithm for Computing All Pure Nash Equilibria for Noncooperative Games in Normal Form. *Theoretical Economics Letters*, 10(06):1253–1259. Cited on page 152.

Corrente, S., S. Greco, and A. Ishizaka
2016. Combining analytical hierarchy process and Choquet integral within non-additive robust ordinal regression. *Omega*, 61:2–18. Cited on page 48.

Daskalakis, C., P. W. Goldberg, and C. H. Papadimitriou
2006. The complexity of computing a nash equilibrium. In *Proceedings of the Thirty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '06, Pp. 71–78, New York, NY, USA. Association for Computing Machinery. Cited on page 56.

Degrave, J., F. Felici, J. Buchli, M. Neunert, B. D. Tracey, F. Carpanese, T. Ewalds, R. Hafner, A. Abdolmaleki, D. de Las Casas, C. Donner, L. Fritz, C. Galperti, A. Huber, J. Keeling, M. Tsimpoukelli, J. Kay, A. Merle, J.-M. Moret, S. Noury, F. Pesamosca, D. Pfau, O. Sauter, C. Sommariva, S. Coda, B. Duval, A. Fasoli, P. Kohli, K. Kavukcuoglu, D. Hassabis, and M. A. Riedmiller
2022. Magnetic control of tokamak plasmas through deep reinforcement learning. *Nat.*, 602(7897):414–419. Cited on page 17.

Delgrange, F.
2024. *Activating Formal Verification of Deep Reinforcement Learning Policies by Model Checking Bisimilar Latent Space Models*. PhD thesis, VUBPRESS, Brussels University Press / Vrije Universiteit Brussel, University of Antwerp. Cited on page 16.

Delgrange, F., Ann Nowe, and G. Perez
2023. Wasserstein auto-encoded MDPs: Formal verification of efficiently distilled RL policies with many-sided guarantees. In *The Eleventh International Conference on Learning Representations*. Cited on page 36.

Delgrange, F., J.-P. Katoen, T. Quatmann, and M. Randour
2020. Simple strategies in multi-objective MDPs. In *Tools and Algorithms for the Construction and Analysis of Systems*, A. Biere and D. Parker, eds., Pp. 346–364, Cham. Springer International Publishing. Cited on page 92.

Deng, Z. and M. Liu
2018. An integrated generation-compensation optimization strategy for enhanced short-term voltage security of large-scale power systems using multi-objective reinforcement learning method. In *2018 International Conference on Power System Technology (POWERCON)*, Pp. 4099–4106. Cited on page 25.

Denuit, M., L. Eeckhoudt, I. Tsetlin, and R. L. Winkler
2013. Multivariate Concave and Convex Stochastic Dominance. In *Risk Measures and Attitudes*, F. Biagini, A. Richter, and H. Schlesinger, eds., Pp. 11–32. London: Springer London. Cited on pages 100, 104, and 114.

Dickhaut, J. and T. Kaplan
1993. A program for finding nash equilibria. In *Economic and Financial Modeling with Mathematica®*, H. R. Varian, ed., Pp. 148–166. New York, NY: Springer New York. Cited on page 56.

Ding, D., X. Wei, Z. Yang, Z. Wang, and M. Jovanovic
2021. Provably efficient safe exploration via primal-dual policy optimization. In *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics*, A. Banerjee and K. Fukumizu, eds., volume 130 of *Proceedings of Machine Learning Research*, Pp. 3304–3312. PMLR. Cited on page 91.

Dolgov, D. and E. Durfee
2005. Stationary deterministic policies for constrained MDPs with multiple rewards, costs, and discount factors. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, IJCAI'05, Pp. 1326–1331, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc. Cited on page 91.

Duan, Z., W. Huang, D. Zhang, Y. Du, J. Wang, Y. Yang, and X. Deng
2023. Is nash equilibrium approximator learnable? In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '23,

Pp. 233–241, Richland, SC. International Foundation for Autonomous Agents and Multiagent Systems. Cited on page 56.

Dyer, J. S.
2005. Maut — Multiattribute Utility Theory. In *Multiple Criteria Decision Analysis: State of the Art Surveys*, J. Figueira, S. Greco, and M. Ehrogott, eds., Pp. 265–292. New York, NY: Springer New York. Cited on pages 45 and 163.

Dyer, J. S. and R. K. Sarin
1979. Measurable multiattribute value functions. *Operations Research*, 27(4):810–822. Cited on page 47.

Epshteyn, A. and G. DeJong
2006. Qualitative reinforcement learning. In *Machine Learning, Proceedings of the Twenty-Third International Conference (ICML 2006), Pittsburgh, Pennsylvania, USA, June 25-29, 2006*, W. W. Cohen and A. W. Moore, eds., volume 148 of *ACM International Conference Proceeding Series*, Pp. 305–312. ACM. Cited on page 101.

Ethayarajh, K., W. Xu, N. Muennighoff, D. Jurafsky, and D. Kiela
2024. KTO: Model alignment as prospect theoretic optimization. *CoRR*, abs/2402.01306. Cited on page 18.

European Parliament and Council
2024. Regulation (EU) 2024/1689 of the European Parliament and of the Council of 13 June 2024 laying down harmonised rules on artificial intelligence (AI Act). `https://data.europa.eu/eli/reg/2024/1689/oj`. Official Journal of the European Union, L 1689, 12 July 2024. Cited on page 18.

Fedus, W., P. Ramachandran, R. Agarwal, Y. Bengio, H. Larochelle, M. Rowland, and W. Dabney
2020. Revisiting fundamentals of experience replay. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, Pp. 3061–3071. PMLR. Cited on page 42.

Feinberg, E. A.
2000. Constrained discounted markov decision processes and hamiltonian cycles. *Mathematics of Operations Research*, 25(1):130–140. Cited on page 91.

Felten, F., L. N. Alegre, A. Nowe, A. L. C. Bazzan, E. G. Talbi, G. Danoy, and B. C. da Silva
2023. A Toolkit for Reliable Benchmarking and Research in Multi-Objective Reinforcement Learning. In *Proceedings of the 37th Conference on Neural Information Processing Systems (NeurIPS 2023)*. Cited on pages 94 and 95.

Felten, F., E.-G. Talbi, and G. Danoy
2024. Multi-objective reinforcement learning based on decomposition: A taxonomy

and framework. *Journal of Artificial Intelligence Research*, 79:679–723. Cited on pages 53 and 63.

Fishburn, P. C.
1974. Convex stochastic dominance with continuous distribution functions. *Journal of Economic Theory*, 7(2):143–158. Cited on pages 100, 105, 109, and 114.

Fu, H., J. Yao, O. Gottesman, F. Doshi-Velez, and G. Konidaris
2023. Performance bounds for model and policy transfer in hidden-parameter mdps. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net. Cited on page 164.

Fudenberg, D. and J. Tirole
1991. *Game Theory*. Cambridge, MA: MIT Press. Cited on page 146.

Gabarró, J., A. García, and M. Serna
2011. The complexity of game isomorphism. *Theoretical Computer Science*, 412(48):6675–6695. Cited on page 129.

Gábor, Z., Z. Kalmár, and C. Szepesvári
1998. Multi-criteria reinforcement learning. In *ICML*, volume 98, Pp. 197–205. Cited on pages 18 and 163.

Gallici, M., M. Fellows, B. Ellis, B. Pou, I. Masmitja, J. N. Foerster, and M. Martin
2025. Simplifying deep temporal difference learning. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net. Cited on page 42.

Ganzfried, S.
2021. Algorithm for Computing Approximate Nash Equilibrium in Continuous Games with Application to Continuous Blotto. *Games*, 12(2):47. Cited on pages 55, 56, and 152.

Gao, L., S. Biderman, S. Black, L. Golding, T. Hoppe, C. Foster, J. Phang, H. He, A. Thite, N. Nabeshima, S. Presser, and C. Leahy
2021. The pile: An 800GB dataset of diverse text for language modeling. *CoRR*, abs/2101.00027. Cited on page 15.

Geist, M., J. Pérolat, M. Laurière, R. Elie, S. Perrin, O. Bachem, R. Munos, and O. Pietquin
2022. Concave utility reinforcement learning: The mean-field game viewpoint. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '22, Pp. 489–497, Richland, SC. International Foundation for Autonomous Agents and Multiagent Systems. Cited on pages 63 and 90.

Gelada, C., S. Kumar, J. Buckman, O. Nachum, and M. G. Bellemare
2019. DeepMDP: Learning continuous latent space models for representation learning. In *Proceedings of the 36th International Conference on Machine Learning*, K. Chaudhuri and R. Salakhutdinov, eds., volume 97 of *Proceedings of Machine Learning Research*, Pp. 2170–2179. PMLR. Cited on page 36.

Gleave, A., M. Dennis, S. Legg, S. Russell, and J. Leike
2021. Quantifying differences in reward functions. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net. Cited on page 164.

Glicksberg, I. L.
1952. A Further Generalization of the Kakutani Fixed Point Theorem, with Application to Nash Equilibrium Points. *Proceedings of the American Mathematical Society*, 3(1):170–170. Cited on pages 56 and 146.

Grabisch, M. and C. Labreuche
2010. A decade of application of the Choquet and Sugeno integrals in multi-criteria decision aid. *Annals of Operations Research*, 175(1):247–286. Cited on page 47.

Greco, S., M. Ehrgott, and J. R. Figueira, eds.
2016. *Multiple Criteria Decision Analysis: State of the Art Surveys*, volume 233 of *International Series in Operations Research & Management Science*. New York, NY: Springer. Cited on pages 18, 45, and 163.

Guerreiro, A. P., C. M. Fonseca, and L. Paquete
2021. The hypervolume indicator: Computational problems and algorithms. *Acm Computing Surveys*, 54(6). Cited on page 121.

Ha, D. and J. Schmidhuber
2018. World models. Cited on page 36.

Haarnoja, T., A. Zhou, P. Abbeel, and S. Levine
2018. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. In *Proceedings of the 35th International Conference on Machine Learning*, J. Dy and A. Krause, eds., volume 80, Pp. 1861–1870. PMLR. Cited on page 42.

Hafner, D., T. P. Lillicrap, M. Norouzi, and J. Ba
2021. Mastering atari with discrete world models. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net. Cited on page 36.

Hartikainen, K., X. Geng, T. Haarnoja, and S. Levine
2020. Dynamical distance learning for semi-supervised and unsupervised skill discovery. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net. Cited on page 164.

Hayes, C. F., R. Rădulescu, E. Bargiacchi, J. Källström, M. Macfarlane, M. Reymond, T. Verstraeten, L. M. Zintgraf, R. Dazeley, F. Heintz, E. Howley, A. A. Irissappane, P. Mannion, A. Nowé, G. Ramos, M. Restelli, P. Vamplew, and D. M. Roijers
2022a. A practical guide to multi-objective reinforcement learning and planning. *Autonomous Agents and Multi-Agent Systems*, 36(1):26. Cited on pages 18, 47, 52, 61, 92, 94, and 99.

Hayes, C. F., M. Reymond, D. M. Roijers, E. Howley, and P. Mannion
2021. Distributional Monte Carlo Tree Search for Risk-Aware and Multi-Objective Reinforcement Learning. In *Proceedings of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021)*, P. 3, Online. IFAAMAS. Cited on page 101.

Hayes, C. F., D. M. Roijers, E. Howley, and P. Mannion
2022b. Decision-theoretic planning for the expected scalarised returns. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '22, Pp. 1621–1623, Richland, SC. International Foundation for Autonomous Agents and Multiagent Systems. Cited on page 101.

Hayes, C. F., T. Verstraeten, D. M. Roijers, E. Howley, and P. Mannion
2022c. Expected scalarised returns dominance: A new solution concept for multi-objective decision making. *Neural Computing and Applications*. Cited on pages 99, 101, 111, and 113.

Hessel, M., J. Modayil, H. van Hasselt, T. Schaul, G. Ostrovski, W. Dabney, D. Horgan, B. Piot, M. G. Azar, and D. Silver
2018. Rainbow: Combining improvements in deep reinforcement learning. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th Innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, S. A. McIlraith and K. Q. Weinberger, eds., Pp. 3215–3222. AAAI Press. Cited on page 42.

Heyman, J. L.
2019. *On the Computation of Strategically Equivalent Games*. PhD thesis, The Ohio State University. Cited on pages 129 and 135.

Horgan, D., J. Quan, D. Budden, G. Barth-Maron, M. Hessel, H. van Hasselt, and D. Silver
2018. Distributed prioritized experience replay. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net. Cited on page 42.

Howard, R. A.
1960. *Dynamic Programming and Markov Processes*. MIT Press. Cited on page 38.

Hsieh, Y.-G., K. Antonakopoulos, and P. Mertikopoulos
2021. Adaptive learning in continuous games: Optimal regret bounds and convergence to nash equilibrium. In *Proceedings of Thirty Fourth Conference on Learning Theory*, M. Belkin and S. Kpotufe, eds., volume 134 of *Proceedings of Machine Learning Research*, Pp. 2388–2422. PMLR. Cited on pages 55 and 56.

Huang, S., R. F. J. Dossa, A. Raffin, A. Kanervisto, and W. Wang
2022. The 37 implementation details of proximal policy optimization. *The ICLR Blog Track 2023*. Cited on page 43.

Huang, S., Q. Gallouédec, F. Felten, A. Raffin, R. F. J. Dossa, Y. Zhao, R. Sullivan, V. Makoviychuk, D. Makoviichuk, M. H. Danesh, C. Roumégous, J. Weng, C. Chen, M. M. Rahman, J. G. M. Araújo, G. Quan, D. Tan, T. Klein, R. Charakorn, M. Towers, Y. Berthelot, K. Mehta, D. Chakraborty, A. KG, V. Charraut, C. Ye, Z. Liu, L. N. Alegre, A. Nikulin, X. Hu, T. Liu, J. Choi, and B. Yi
2024. Open RL benchmark: Comprehensive tracked experiments for reinforcement learning. *CoRR*, abs/2402.03046. Cited on page 43.

Hughes, E. J.
2005. Evolutionary many-objective optimisation: Many once or one many? In *Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2005, 2-4 September 2005, Edinburgh, UK*, Pp. 222–227. IEEE. Cited on page 50.

Igarashi, A. and D. M. Roijers
2017. Multi-criteria Coalition Formation Games. In *International Conference on Algorithmic Decision Theory*, volume 10576 LNAI, Pp. 197–213. Springer. Cited on page 59.

Ismaili, A.
2018. On existence, mixtures, computation and efficiency in multi-objective games. In *PRIMA 2018: Principles and Practice of Multi-Agent Systems*, T. Miller, N. Oren, Y. Sakurai, I. Noda, B. T. R. Savarimuthu, and T. Cao Son, eds., Pp. 210–225, Cham. Springer International Publishing. Cited on page 58.

Jaakkola, T. S., M. I. Jordan, and S. P. Singh
1994. On the convergence of stochastic iterative dynamic programming algorithms. *Neural Computation*, 6(6):1185–1201. Cited on pages 39 and 40.

Jacquet-Lagrèze, E., R. Meziani, and R. Slowinski
1987. Molp with an interactive assessment of a piecewise linear utility function. *Methodology for Public Decision-Making Interactive Decision Support Systems Queue and Game Theory*, 31(3):350–357. Cited on page 47.

Jouppi, N. P., C. Young, N. Patil, D. A. Patterson, G. Agrawal, R. Bajwa, S. Bates, S. Bhatia, N. Boden, A. Borchers, R. Boyle, P.-l. Cantin, C. Chao, C. Clark, J. Coriell, M. Daley, M. Dau, J. Dean, B. Gelb, T. V. Ghaemmaghami, R. Gottipati, W. Gulland, R. Hagmann, C. R. Ho, D. Hogberg, J. Hu, R. Hundt, D. Hurt, J. Ibarz, A. Jaffey, A. Jaworski, A. Kaplan, H. Khaitan, D. Killebrew, A. Koch, N. Kumar, S. Lacy, J. Laudon, J. Law, D. Le, C. Leary, Z. Liu, K. Lucke, A. Lundin, G. MacKean, A. Maggiore, M. Mahony, K. Miller, R. Nagarajan, R. Narayanaswami, R. Ni, K. Nix, T. Norrie, M. Omernick, N. Penukonda, A. Phelps, J. Ross, M. Ross, A. Salek, E. Samadiani, C. Severn, G. Sizikov, M. Snelham, J. Souter, D. Steinberg, A. Swing, M. Tan, G. Thorson, B. Tian, H. Toma, E. Tuttle, V. Vasudevan, R. Walter, W. Wang, E. Wilcox, and D. H. Yoon
2017. In-datacenter performance analysis of a tensor processing unit. In *Proceedings of*

the 44th Annual International Symposium on Computer Architecture, ISCA 2017, Toronto, on, Canada, June 24-28, 2017, Pp. 1–12. ACM. Cited on page 15.

Judd, K. L., P. Renner, and K. Schmedders
2012. Finding all pure-strategy equilibria in games with continuous strategies. *Quantitative Economics*, 3(2):289–331. Cited on pages 128 and 155.

Kahneman, D. and A. Tversky
1979. Prospect theory: An analysis of decision under risk. *Econometrica : journal of the Econometric Society*, 47(2):263–291. Cited on pages 18 and 45.

Keeney, R. L.
1974. Multiplicative utility functions. *Operations Research*, 22(1):22–34. Cited on page 47.

Keeney, R. L. and H. Raiffa
1993. *Decisions with Multiple Objectives: Preferences and Value Trade-Offs*. Cambridge: Cambridge University Press. Cited on pages 45, 46, 47, and 163.

Keeney, R. L. and E. F. Wood
1977. An illustrative example of the use of multiattribute utility theory for water resource planning. *Water Resources Research*, 13(4):705–712. Cited on page 47.

Kendall, A., J. Hawke, D. Janz, P. Mazur, D. Reda, J. M. Allen, V. D. Lam, A. Bewley, and A. Shah
2019. Learning to drive in a day. In *2019 International Conference on Robotics and Automation (ICRA)*, Pp. 8248–8254. IEEE. Cited on page 15.

Kingma, D. P. and J. Ba
2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, eds. Cited on page 41.

Kirillov, A., E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. B. Girshick
2023. Segment anything. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, Pp. 3992–4003. IEEE. Cited on page 15.

Klenke, A.
2020. *Probability Theory: A Comprehensive Course*, Universitext, 3 edition. Springer Cham. Cited on pages 26 and 83.

Knox, W. B., A. Allievi, H. Banzhaf, F. Schmitt, and P. Stone
2023. Reward (Mis)design for autonomous driving. *Artificial Intelligence*, 316:103829. Cited on page 19.

Komorowski, M., L. A. Celi, O. Badawi, A. C. Gordon, and A. A. Faisal
2018. The Artificial Intelligence Clinician learns optimal treatment strategies for sepsis in intensive care. *Nature Medicine*, 24(11):1716–1720. Cited on page 15.

Kopa, M. and B. Petrová
2018. Strong and Weak Multivariate First-Order Stochastic Dominance. Cited on page 105.

Koppelman, F. S.
1981. Non-linear utility functions in models of travel choice behavior. *Transportation*, 10(2):127–146. Cited on page 47.

Legriel, J., C. Le Guernic, S. Cotton, and O. Maler
2010. Approximating the pareto front of multi-criteria optimization problems. In *Tools and Algorithms for the Construction and Analysis of Systems*, J. Esparza and R. Majumdar, eds., Pp. 69–83, Berlin, Heidelberg. Springer Berlin Heidelberg. Cited on page 63.

Lemke, C. E. and Jr. Howson J. T.
1964. Equilibrium Points of Bimatrix Games. *Journal of the Society for Industrial and Applied Mathematics*, 12(2):413–423. Cited on page 56.

Letcher, A., D. Balduzzi, S. Racanière, J. Martens, J. Foerster, K. Tuyls, and T. Graepel
2019. Differentiable Game Mechanics. *Journal of Machine Learning Research*, 20(84):1–40. Cited on page 141.

Levine, S., C. Finn, T. Darrell, and P. Abbeel
2016. End-to-end training of deep visuomotor policies. *Journal of Machine Learning Research*, 17:39:1–39:40. Cited on page 17.

Levy, H.
2016a. Bivariate FSD (BFSD). In *Stochastic Dominance: Investment Decision Making under Uncertainty*, Pp. 441–465. Cham: Springer International Publishing. Cited on page 100.

Levy, H.
2016b. Stochastic dominance decision rules. In *Stochastic Dominance: Investment Decision Making under Uncertainty*, Pp. 41–124. Cham: Springer International Publishing. Cited on pages 101 and 103.

Leyton-Brown, Yoav, K. a. S.
2008. *Essentials of Game Theory: A Concise, Multidisciplinary Introduction*, 1st edition. Morgan and Claypool Publishers. Cited on pages 18 and 53.

Li, M., X. Pan, C. Liu, and Z. Li
2025. Federated deep reinforcement learning-based urban traffic signal optimal control. *Scientific Reports*, 15(1):11724. Cited on page 43.

Lillicrap, T. P., J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra
2016. Continuous control with deep reinforcement learning. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, Y. Bengio and Y. LeCun, eds. Cited on page 42.

Lim, S. H. and I. Malik
2022. Distributional reinforcement learning for risk-sensitive policies. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, eds. Cited on page 100.

Lipton, R. J., E. Markakis, and A. Mehta
2003. Playing large games using simple strategies. In *Proceedings 4th ACM Conference on Electronic Commerce (EC-2003), San Diego, California, USA, June 9-12, 2003*, D. A. Menascé and N. Nisan, eds., Pp. 36–41. ACM. Cited on page 56.

Liu, R., Y. Pan, L. Xu, L. Song, P. You, Y. Chen, and J. Bian
2025. Efficient discovery of pareto front for multi-objective reinforcement learning. In *The Thirteenth International Conference on Learning Representations*. Cited on pages 18 and 53.

Liu, S., L. Marris, G. Piliouras, I. Gemp, and N. Heess
2024. NfgTransformer: Equivariant representation learning for normal-form games. In *The Twelfth International Conference on Learning Representations*. Cited on pages 56 and 158.

Loshchilov, I. and F. Hutter
2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net. Cited on page 41.

Lozovanu, D., D. Solomon, and A. Zelikovsky
2005. Multiobjective games and determining Pareto-Nash equilibria. *The Bulletin of Academy of Sciences of Moldova, Mathematics*, 3(49):115–122. Cited on page 58.

Lu, C., P. J. Ball, Y. W. Teh, and J. Parker-Holder
2023a. Synthetic experience replay. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, eds. Cited on page 42.

Lu, H., D. Herman, and Y. Yu
2023b. Multi-objective reinforcement learning: Convexity, stationarity and pareto optimality. In *The Eleventh International Conference on Learning Representations*. Cited on pages 53, 62, and 68.

Lu, J., P. Mannion, and K. Mason
2022. A multi-objective multi-agent deep reinforcement learning approach to residential appliance scheduling. *IET Smart Grid*, n/a(n/a). Cited on page 128.

Luo, J., C. Paduraru, O. Voicu, Y. Chervonyi, S. Munns, J. Li, C. Qian, P. Dutta, J. Q. Davis, N. Wu, X. Yang, C.-M. Chang, T. Li, R. Rose, M. Fan, H. Nakhost, T. Liu, B. Kirkman, F. Altamura, L. Cline, P. Tonker, J. Gouker, D. Uden, W. B. Bryan, J. Law, D. Fatiha, N. Satra, J. Rothenberg, M. Carlin, S. Tallapaka, S. Witherspoon, D. Parish, P. Dolan, C. Zhao, and D. J. Mankowitz
2022. Controlling commercial cooling systems using reinforcement learning. *CoRR*, abs/2211.07357. Cited on page 15.

Makoviychuk, V., L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa, and G. State
2021. Isaac gym: High performance GPU based physics simulation for robot learning. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, Virtual*, J. Vanschoren and S.-K. Yeung, eds. Cited on page 41.

Mandow, L., J. L. Perez-de-la-Cruz, and N. Pozas
2022. Multi-objective dynamic programming with limited precision. *Journal of Global Optimization*, 82(3):595–614. Cited on page 121.

Mannion, P. and R. Radulescu
2023. Comparing utility-based and pareto-based solution sets in multi-objective normal form games. In *Proceedings of the Multi-Objective Decision Making Workshop (MODeM 2023)*. Cited on page 158.

Martin, J. D., M. Lyskawinski, X. Li, and B. J. Englot
2020. Stochastically dominant distributional reinforcement learning. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, Pp. 6745–6754. PMLR. Cited on page 101.

Mas-Colell, A., M. D. Whinston, J. R. Green, et al.
1995. *Microeconomic Theory*, volume 1. Oxford university press New York. Cited on pages 47 and 48.

Maschler, M., E. Solan, and S. Zamir
2013. *Game Theory*. Cambridge: Cambridge University Press. Cited on pages 129 and 158.

McKinney, S. M., M. Sieniek, V. Godbole, J. Godwin, N. Antropova, H. Ashrafian, T. Back, M. Chesus, G. S. Corrado, A. Darzi, M. Etemadi, F. Garcia-Vicente, F. J. Gilbert, M. Halling-Brown, D. Hassabis, S. Jansen, A. Karthikesalingam, C. J. Kelly, D. King, J. R. Ledsam, D. Melnick, H. Mostofi, L. Peng, J. J. Reicher, B. Romera-Paredes, R. Sidebottom, M. Suleyman, D. Tse, K. C. Young, J. De Fauw, and S. Shetty
2020. International evaluation of an AI system for breast cancer screening. *Nature*, 577(7788):89–94. Cited on page 15.

Meng, W., Q. Zheng, G. Pan, and Y. Yin
2023. Off-policy proximal policy optimization. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(8):9162–9170. Cited on page 93.

Mertikopoulos, P. and W. H. Sandholm
2016. Learning in games via reinforcement and regularization. *Mathematics of Operations Research*, 41(4):1297–1324. Cited on page 56.

Miettinen, K.
1998. *Nonlinear Multiobjective Optimization*, volume 12 of *International Series in Operations Research & Management Science*. Boston, MA: Springer US. Cited on pages 18, 50, 64, and 84.

Mnih, V., A. P. Badia, L. Mirza, A. Graves, T. Harley, T. P. Lillicrap, D. Silver, and K. Kavukcuoglu
2016. Asynchronous methods for deep reinforcement learning. In *33rd International Conference on Machine Learning, ICML 2016*, M. F. Balcan and K. Q. Weinberger, eds., volume 4, Pp. 2850–2869, New York, New York, USA. PMLR. Cited on pages 43, 44, and 93.

Mnih, V., K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis
2015. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533. Cited on pages 36, 41, and 93.

Moerland, T. M., J. Broekens, A. Plaat, and C. M. Jonker
2022. A Unifying Framework for Reinforcement Learning and Planning. *Frontiers in Artificial Intelligence*, 5:908353. Cited on page 36.

Moerland, T. M., J. Broekens, A. Plaat, and C. M. Jonker
2023. Model-based reinforcement learning: A survey. *Found. Trends Mach. Learn.*, 16(1):1–118. Cited on page 36.

Montenegro, A., M. Mussi, A. M. Metelli, and M. Papini
2024. Learning optimal deterministic policies with stochastic policy gradients. In *Proceedings of the 41st International Conference on Machine Learning*, R. Salakhutdinov, Z. Kolter, K. Heller, A. Weller, N. Oliver, J. Scarlett, and F. Berkenkamp, eds., volume 235 of *Proceedings of Machine Learning Research*, Pp. 36160–36211. PMLR. Cited on pages 35 and 93.

Muslimani, C., K. Johnstonbaugh, S. Chandramouli, S. Booth, W. B. Knox, and M. E. Taylor
2025. Towards improving reward design in RL: A reward alignment metric for RL practitioners. *Reinforcement Learning Journal*. Cited on page 164.

Nash, J.
1951. Non-Cooperative Games. *The Annals of Mathematics*, 54(2):286–286. Cited on pages 19, 55, 56, and 128.

Nikulin, Y., K. Miettinen, and M. M. Mäkelä
2012. A new achievement scalarizing function based on parameterization in multiobjective optimization. *OR Spectrum*, 34(1):69–87. Cited on page 89.

Nota, C. and P. S. Thomas
2020. Is the policy gradient a gradient? In *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '20, Pp. 939–947, Richland, SC. International Foundation for Autonomous Agents and Multiagent Systems. Cited on page 43.

Obermeyer, Z., B. Powers, C. Vogeli, and S. Mullainathan
2019. Dissecting racial bias in an algorithm used to manage the health of populations. *Science*, 366(6464):447–453. Cited on page 19.

OpenAI, J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, R. Avila, I. Babuschkin, S. Balaji, V. Balcom, P. Baltescu, H. Bao, M. Bavarian, J. Belgum, I. Bello, J. Berdine, G. Bernadett-Shapiro, C. Berner, L. Bogdonoff, O. Boiko, M. Boyd, A.-L. Brakman, G. Brockman, T. Brooks, M. Brundage, K. Button, T. Cai, R. Campbell, A. Cann, B. Carey, C. Carlson, R. Carmichael, B. Chan, C. Chang, F. Chantzis, D. Chen, S. Chen, R. Chen, J. Chen, M. Chen, B. Chess, C. Cho, C. Chu, H. W. Chung, D. Cummings, J. Currier, Y. Dai, C. Decareaux, T. Degry, N. Deutsch, D. Deville, A. Dhar, D. Dohan, S. Dowling, S. Dunning, A. Ecoffet, A. Eleti, T. Eloundou, D. Farhi, L. Fedus, N. Felix, S. P. Fishman, J. Forte, I. Fulford, L. Gao, E. Georges, C. Gibson, V. Goel, T. Gogineni, G. Goh, R. Gontijo-Lopes, J. Gordon, M. Grafstein, S. Gray, R. Greene, J. Gross, S. S. Gu, Y. Guo, C. Hallacy, J. Han, J. Harris, Y. He, M. Heaton, J. Heidecke, C. Hesse, A. Hickey, W. Hickey, P. Hoeschele, B. Houghton, K. Hsu, S. Hu, X. Hu, J. Huizinga, S. Jain, S. Jain, J. Jang, A. Jiang, R. Jiang, H. Jin, D. Jin, S. Jomoto, B. Jonn, H. Jun, T. Kaftan, Ł. Kaiser, A. Kamali, I. Kanitscheider, N. S. Keskar, T. Khan, L. Kilpatrick, J. W. Kim, C. Kim, Y. Kim, J. H. Kirchner, J. Kiros, M. Knight, D. Kokotajlo, Ł. Kondraciuk, A. Kondrich, A. Konstantinidis, K. Kosic, G. Krueger, V. Kuo, M. Lampe, I. Lan, T. Lee, J. Leike, J. Leung, D. Levy, C. M. Li, R. Lim, M. Lin, S. Lin, M. Litwin, T. Lopez, R. Lowe, P. Lue, A. Makanju, K. Malfacini, S. Manning, T. Markov, Y. Markovski, B. Martin, K. Mayer, A. Mayne, B. McGrew, S. M. McKinney, C. McLeavey, P. McMillan, J. McNeil, D. Medina, A. Mehta, J. Menick, L. Metz, A. Mishchenko, P. Mishkin, V. Monaco, E. Morikawa, D. Mossing, T. Mu, M. Murati, O. Murk, D. Mély, A. Nair, R. Nakano, R. Nayak, A. Neelakantan, R. Ngo, H. Noh, L. Ouyang, C. O'Keefe, J. Pachocki, A. Paino, J. Palermo, A. Pantuliano, G. Parascandolo, J. Parish, E. Parparita, A. Passos, M. Pavlov, A. Peng, A. Perelman, F. de Avila Belbute Peres, M. Petrov, H. P. de Oliveira

Pinto, Michael, Pokorny, M. Pokrass, V. H. Pong, T. Powell, A. Power, B. Power, E. Proehl, R. Puri, A. Radford, J. Rae, A. Ramesh, C. Raymond, F. Real, K. Rimbach, C. Ross, B. Rotsted, H. Roussez, N. Ryder, M. Saltarelli, T. Sanders, S. Santurkar, G. Sastry, H. Schmidt, D. Schnurr, J. Schulman, D. Selsam, K. Sheppard, T. Sherbakov, J. Shieh, S. Shoker, P. Shyam, S. Sidor, E. Sigler, M. Simens, J. Sitkin, K. Slama, I. Sohl, B. Sokolowsky, Y. Song, N. Staudacher, F. P. Such, N. Summers, I. Sutskever, J. Tang, N. Tezak, M. B. Thompson, P. Tillet, A. Tootoonchian, E. Tseng, P. Tuggle, N. Turley, J. Tworek, J. F. C. Uribe, A. Vallone, A. Vijayvergiya, C. Voss, C. Wainwright, J. J. Wang, A. Wang, B. Wang, J. Ward, J. Wei, CJ. Weinmann, A. Welihinda, P. Welinder, J. Weng, L. Weng, M. Wiethoff, D. Willner, C. Winter, S. Wolrich, H. Wong, L. Workman, S. Wu, J. Wu, M. Wu, K. Xiao, T. Xu, S. Yoo, K. Yu, Q. Yuan, W. Zaremba, R. Zellers, C. Zhang, M. Zhang, S. Zhao, T. Zheng, J. Zhuang, W. Zhuk, and B. Zoph
2024. GPT-4 technical report. Cited on page 15.

Osika, Z., R. Radulescu, J. Z. Salazar, F. A. Oliehoek, and P. K. Murukannaiah
2025. Multi-objective reinforcement learning for water management. In *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2025, Detroit, MI, USA, May 19-23, 2025*, S. Das, A. Nowé, and Y. Vorobeychik, eds., Pp. 2702–2704. International Foundation for Autonomous Agents and Multiagent Systems / ACM. Cited on page 25.

Papadimitriou, C. and M. Yannakakis
2000. On the approximability of trade-offs and optimal access of Web sources. In *Proceedings 41st Annual Symposium on Foundations of Computer Science*, Pp. 86–92. Cited on pages 67, 77, and 91.

Papadimitriou, C. H. and K. Steiglitz
1998. *Combinatorial Optimization: Algorithms and Complexity*. Courier Corporation. Cited on page 68.

Parrilo, P. A.
2006. Polynomial games and sum of squares optimization. In *Proceedings of the 45th IEEE Conference on Decision and Control*, Pp. 2855–2860, San Diego, CA, USA. IEEE. Cited on page 153.

Patil, G., A. Mahajan, and D. Precup
2024. On learning history-based policies for controlling Markov decision processes. In *International Conference on Artificial Intelligence and Statistics, 2-4 May 2024, Palau de Congressos, Valencia, Spain*, S. Dasgupta, S. Mandt, and Y. Li, eds., volume 238 of *Proceedings of Machine Learning Research*, Pp. 3511–3519. PMLR. Cited on page 35.

Polyak, B.
1964. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17. Cited on page 42.

Porter, R., E. Nudelman, and Y. Shoham
2008. Simple search methods for finding a Nash equilibrium. *Games and Economic Behavior*, 63(2):642–662. Cited on page 56.

Puterman, M. L.
1994. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, Wiley Series in Probability and Statistics. Wiley. Cited on page 32.

Qu, S., Y. Ji, and M. Goh
2015. The Robust Weighted Multi-Objective Game. *PLOS ONE*, 10(9):e0138970. Cited on page 58.

Rădulescu, R., P. Mannion, D. M. Roijers, and A. Nowé
2020a. Multi-objective multi-agent decision making: A utility-based analysis and survey. *Autonomous Agents and Multi-Agent Systems*, 34(1):10–10. Cited on pages 19, 45, and 127.

Rădulescu, R., P. Mannion, Y. Zhang, D. M. Roijers, and A. Nowé
2020b. A utility-based analysis of equilibria in multi-objective normal-form games. *The Knowledge Engineering Review*, 35:e32–e32. Cited on pages 22, 59, and 128.

Reymond, M., E. Bargiacchi, and A. Nowé
2022. Pareto conditioned networks. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '22, Pp. 1110–1118, Richland, SC. International Foundation for Autonomous Agents and Multiagent Systems. Cited on pages 53, 62, and 95.

Reymond, M., C. F. Hayes, D. Steckelmacher, D. M. Roijers, and A. Nowé
2023. Actor-critic multi-objective reinforcement learning for non-linear utility functions. *Autonomous Agents and Multi-Agent Systems*, 37(2):23. Cited on pages 63, 93, and 101.

Reymond, M., C. F. Hayes, L. Willem, R. Rădulescu, S. Abrams, D. M. Roijers, E. Howley, P. Mannion, N. Hens, A. Nowé, and P. Libin
2024. Exploring the Pareto front of multi-objective COVID-19 mitigation policies using reinforcement learning. *Expert Systems with Applications*, 249:123686. Cited on page 18.

Richard, S. F.
1975. Multivariate Risk Aversion, Utility Independence and Separable Utility Functions. *Management Science*, 22(1):12–21. Cited on pages 100 and 103.

Robbins, H. and S. Monro
1951. A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3):400–407. Cited on page 39.

Robinson, J.
1951. An Iterative Method of Solving a Game. *Annals of Mathematics*, 54(2):296–301. Cited on pages 56, 57, and 152.

Rodriguez-Soto, M., J. A. R. Aguilar, and M. López-Sánchez
2024. An analytical study of utility functions in multi-objective reinforcement learning. In *The Thirty-Eighth Annual Conference on Neural Information Processing Systems*. Cited on page 52.

Roijers, D., E. Walraven, and M. Spaan
2018. Bootstrapping lps in value iteration for multi-objective and partially observable mdps. *Proceedings of the International Conference on Automated Planning and Scheduling*, 28(1):218–226. Cited on page 119.

Roijers, D. M.
2016. *Multi-Objective Decision-Theoretic Planning*. PhD thesis, University of Amsterdam. Cited on pages 46 and 62.

Roijers, D. M., P. Vamplew, S. Whiteson, and R. Dazeley
2013. A survey of multi-objective sequential decision-making. *Journal of Artificial Intelligence Research*, 48:67–113. Cited on pages 45, 46, and 99.

Roijers, D. M. and S. Whiteson
2017. Multi-objective decision making. In *Synthesis Lectures on Artificial Intelligence and Machine Learning*, volume 34, Pp. 129–129. Morgan and Claypool. Cited on pages 18, 46, 50, 62, 64, 114, and 116.

Roijers, D. M., S. Whiteson, P. Vamplew, and R. Dazeley
2015. Why multi-objective reinforcement learning. In *European Workshop on Reinforcement Learning*, Pp. 1–2. Cited on page 17.

Röpke, W., C. Groenland, R. Rădulescu, A. Nowé, and D. M. Roijers
2023a. Bridging the gap between single and multi objective games. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2023, London, United Kingdom, 29 May 2023 - 2 June 2023*, N. Agmon, B. An, A. Ricci, and W. Yeoh, eds., Pp. 224–232. ACM. Cited on page 127.

Röpke, W., C. F. Hayes, P. Mannion, E. Howley, A. Nowé, and D. M. Roijers
2023b. Distributional multi-objective decision making. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI 2023, 19th-25th August 2023, Macao, SAR, China*, Pp. 5711–5719. ijcai.org. Cited on page 99.

Röpke, W., M. Reymond, P. Mannion, D. M. Roijers, A. Nowé, and R. Rădulescu
2025. Divide and conquer: Provably unveiling the pareto front with multi-objective reinforcement learning. In *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems*, Aamas '25, Pp. 1774–1783, Richland, SC. International Foundation for Autonomous Agents and Multiagent Systems. Cited on page 61.

Röpke, W., D. M. Roijers, A. Nowé, and R. Rădulescu
2022. On nash equilibria in normal-form games with vectorial payoffs. *Auton. Agents Multi Agent Syst.*, 36(2):53. Cited on page 127.

Rosen, J. B.
1965. Existence and uniqueness of equilibrium points for concave n-person games. *Econometrica : journal of the Econometric Society*, 33(3):520–534. Cited on page 158.

Rowland, M., M. Bellemare, W. Dabney, R. Munos, and Y. W. Teh
2018. An analysis of categorical distributional reinforcement learning. In *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, A. Storkey and F. Perez-Cruz, eds., volume 84 of *Proceedings of Machine Learning Research*, Pp. 29–37. PMLR. Cited on page 101.

Royden, H. L. and P. Fitzpatrick
1988. *Real Analysis*, volume 32. Macmillan New York. Cited on pages 26 and 27.

Rudin, W.
1991. *Functional Analysis*, International Series in Pure and Applied Mathematics, 2 edition. McGraw-Hill. Cited on page 143.

Ruelens, F., B. Claessens, S. Quaiyum, B. D. Schutter, R. Babuska, and R. Belmans
2018. Reinforcement learning applied to an electric water heater: From theory to practice. *IEEE Transactions on Smart Grid*, 9(4):3792–3800. Cited on page 15.

Rummery, G. A. and M. Niranjan
1994. *On-Line Q-learning Using Connectionist Systems*, volume 37. University of Cambridge, Department of Engineering Cambridge, UK. Cited on page 40.

Sandholm, T., A. Gilpin, and V. Conitzer
2005. Mixed-integer programming methods for finding nash equilibria. In *Proceedings, the Twentieth National Conference on Artificial Intelligence and the Seventeenth Innovative Applications of Artificial Intelligence Conference, July 9-13, 2005, Pittsburgh, Pennsylvania, USA*, M. M. Veloso and S. Kambhampati, eds., Pp. 495–501. AAAI Press / The MIT Press. Cited on page 56.

Schaul, T., D. Horgan, K. Gregor, and D. Silver
2015. Universal value function approximators. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, F. R. Bach and D. M. Blei, eds., volume 37 of *JMLR Workshop and Conference Proceedings*, Pp. 1312–1320. JMLR.org. Cited on page 165.

Schaul, T., J. Quan, I. Antonoglou, and D. Silver
2016. Prioritized experience replay. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, Y. Bengio and Y. LeCun, eds. Cited on page 42.

Schrittwieser, J., I. Antonoglou, T. Hubert, K. Simonyan, L. Sifre, S. Schmitt, A. Guez, E. Lockhart, D. Hassabis, T. Graepel, T. Lillicrap, and D. Silver
2020. Mastering Atari, Go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609. Cited on page 36.

Schulman, J., S. Levine, P. Abbeel, M. Jordan, and P. Moritz
2015. Trust Region Policy Optimization. In *Proceedings of the 32nd International Conference on Machine Learning*, F. Bach and D. Blei, eds., volume 37, Pp. 1889–1897, Lille, France. PMLR. Cited on page 43.

Schulman, J., P. Moritz, S. Levine, M. I. Jordan, and P. Abbeel
2016. High-dimensional continuous control using generalized advantage estimation. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, Y. Bengio and Y. LeCun, eds. Cited on page 43.

Schulman, J., F. Wolski, P. Dhariwal, A. Radford, and O. Klimov
2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*. Cited on pages 36, 43, 44, and 93.

Shao, H., L. Cohen, A. Blum, Y. Mansour, A. Saha, and M. R. Walter
2023. Eliciting user preferences for personalized multi-objective decision making through comparative feedback. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, eds. Cited on page 165.

Shapley, L. S. and F. D. Rigby
1959. Equilibrium points in games with vector payoffs. *Naval Research Logistics Quarterly*, 6(1):57–61. Cited on pages 19 and 58.

Shoham, Y. and K. Leyton-Brown
2008. *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge: Cambridge University Press. Cited on pages 18 and 53.

Siddique, U., P. Weng, and M. Zimmer
2020. Learning fair policies in multi-objective (Deep) reinforcement learning with average and discounted rewards. In *Proceedings of the 37th International Conference on Machine Learning*, H. D. III and A. Singh, eds., volume 119 of *Proceedings of Machine Learning Research*, Pp. 8905–8915. PMLR. Cited on pages 63 and 92.

Silver, D., A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis
2016. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489. Cited on pages 15 and 17.

Silver, D., G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller
2014. Deterministic Policy Gradient Algorithms. In *Proceedings of the 31st International Conference on Machine Learning*, E. P. Xing and T. Jebara, eds., volume 32, Pp. 387–395, Bejing, China. PMLR. Cited on page 35.

Silver, D., S. Singh, D. Precup, and R. S. Sutton
2021. Reward is enough. *Artificial Intelligence*, 299:103535. Cited on page 17.

Sinha, A., P. Malo, A. Frantsev, and K. Deb
2013. Multi-objective Stackelberg game between a regulating authority and a mining company: A case study in environmental economics. In *2013 IEEE Congress on Evolutionary Computation*, Pp. 478–485, Cancun, Mexico. IEEE. Cited on page 128.

Siskos, Y., E. Grigoroudis, and N. F. Matsatsinis
2005. UTA Methods. In *Multiple Criteria Decision Analysis: State of the Art Surveys*, J. Figueira, S. Greco, and M. Ehrogott, eds., Pp. 297–334. New York, NY: Springer New York. Cited on page 47.

Skalse, J. and A. Abate
2023. On the limitations of Markovian rewards to express multi-objective, risk-sensitive, and modal tasks. In *Uncertainty in Artificial Intelligence, UAI 2023, July 31 - 4 August 2023, Pittsburgh, PA, USA*, R. J. Evans and I. Shpitser, eds., volume 216 of *Proceedings of Machine Learning Research*, Pp. 1974–1984. PMLR. Cited on page 17.

Skinner, B. F.
1953. *Science and Human Behavior*. New York: Macmillan. Cited on page 17.

Smith, S. F., G. J. Barlow, X.-F. Xie, and Z. B. Rubinstein
2013. Smart urban signal networks: Initial application of the SURTRAC adaptive traffic signal control system. In *Proceedings of the Twenty-Third International Conference on Automated Planning and Scheduling, ICAPS 2013, Rome, Italy, June 10-14, 2013*, D. Borrajo, S. Kambhampati, A. Oddi, and S. Fratini, eds. AAAI. Cited on page 18.

Somasundaram, K. K. and J. S. Baras
2009. Achieving symmetric pareto nash equilibria using biased replicator dynamics. In *Proceedings of the IEEE Conference on Decision and Control*, Pp. 7000–7005, Shanghai, China. IEEE. Cited on page 58.

Stein, N. D., A. Ozdaglar, and P. A. Parrilo
2008. Separable and low-rank continuous games. *International Journal of Game Theory*, 37(4):475–504. Cited on pages 54, 128, and 153.

Stein, N. D., P. A. Parrilo, and A. Ozdaglar
2011. Correlated equilibria in continuous games: Characterization and computation. *Games and Economic Behavior*, 71(2):436–455. Cited on page 153.

Subramani, R., M. Williams, M. Heitmann, H. Holm, C. Griffin, and J. M. V. Skalse
2024. On the expressivity of objective-specification formalisms in reinforcement

learning. In *The Twelfth International Conference on Learning Representations*. Cited on pages 17 and 52.

Sun, C., A. Shrivastava, S. Singh, and A. Gupta
2017. Revisiting unreasonable effectiveness of data in deep learning era. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, Pp. 843–852. IEEE Computer Society. Cited on page 15.

Sutton, R. S.
1995. Generalization in reinforcement learning: Successful examples using sparse coarse coding. In *Advances in Neural Information Processing Systems 8, NIPS, Denver, CO, USA, November 27-30, 1995*, D. S. Touretzky, M. Mozer, and M. E. Hasselmo, eds., Pp. 1038–1044. MIT Press. Cited on page 40.

Sutton, R. S. and A. G. Barto
2018. *Reinforcement Learning: An Introduction*, second edition. Cambridge, MA: MIT Press. Cited on pages 17, 32, and 40.

Sutton, R. S., D. A. McAllester, S. Singh, and Y. Mansour
1999. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems 12, [NIPS Conference, Denver, Colorado, USA, November 29 - December 4, 1999]*, S. A. Solla, T. K. Leen, and K.-R. Müller, eds., Pp. 1057–1063. The MIT Press. Cited on page 42.

Szepesvári, C.
2010. *Algorithms for Reinforcement Learning*, Synthesis Lectures on Artificial Intelligence and Machine Learning. Cham: Springer International Publishing. Cited on page 17.

Taboada, H. A., F. Baheranwala, D. W. Coit, and N. Wattanapongsakorn
2007. Practical solutions for multi-objective optimization: An application to system reliability design problems. *Selected Papers Presented at the Fourth International Conference on Quality and Reliability*, 92(3):314–322. Cited on page 100.

Tambe, M.
2012. *Security and Game Theory - Algorithms, Deployed Systems, Lessons Learned*. Cambridge University Press. Cited on page 18.

Thorndike, E. L.
1911. *Animal Intelligence: Experimental Studies*. New York: Macmillan. Cited on page 17.

Touati, A. and Y. Ollivier
2021. Learning one representation to optimize all rewards. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, Virtual*, M. Ranzato, A. Beygelzimer, Y. N. Dauphin, P. Liang, and J. W. Vaughan, eds., Pp. 13–23. Cited on pages 98 and 164.

Tsitsiklis, J. N.
1994. Asynchronous stochastic approximation and q-learning. *Machine Learning*, 16(3):185–202. Cited on page 39.

Vamplew, P., R. Dazeley, A. Berry, R. Issabekov, and E. Dekker
2011. Empirical evaluation methods for multiobjective reinforcement learning algorithms. *Machine Learning*, 84(1):51–80. Cited on page 95.

Vamplew, P., B. J. Smith, J. Källström, G. de Oliveira Ramos, R. Radulescu, D. M. Roijers, C. F. Hayes, F. Heintz, P. Mannion, P. J. K. Libin, R. Dazeley, and C. Foale
2022. Scalar reward is not enough: A response to Silver, Singh, Precup and Sutton (2021). *Auton. Agents Multi Agent Syst.*, 36(2):41. Cited on page 17.

van Hasselt, H., A. Guez, and D. Silver
2016. Deep reinforcement learning with double q-learning. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, D. Schuurmans and M. P. Wellman, eds., Pp. 2094–2100. AAAI Press. Cited on page 42.

Van Moffaert, K., M. M. Drugan, and A. Nowé
2013. Scalarized multi-objective reinforcement learning: Novel design techniques. In *2013 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)*, Pp. 191–199. Cited on pages 63 and 121.

Van Moffaert, K. and A. Nowé
2014. Multi-objective reinforcement learning using sets of pareto dominating policies. *Journal of Machine Learning Research*, 15(107):3663–3692. Cited on pages 18, 100, and 119.

Varian, H. R.
2014. *Intermediate Microeconomics: A Modern Approach: Ninth Edition*. WW Norton \& Company. Cited on pages 46, 47, 48, and 155.

Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin
2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, eds., Pp. 5998–6008. Cited on page 15.

von Neumann, J. and O. Morgenstern
1944. *Theory of Games and Economic Behavior*, 1 edition. Princeton, NJ: Princeton University Press. Cited on page 45.

Wang, R., K. Frans, P. Abbeel, S. Levine, and A. A. Efros
2024. Prioritized generative replay. Cited on page 42.

Wang, S. Y.
1993. Existence of a pareto equilibrium. *Journal of Optimization Theory and Applications*, 79(2):373–384. Cited on page 58.

Wang, Y.
2025. Game-theoretic understandings of multi-agent systems with multiple objectives. Cited on page 58.

Wang, Z., T. Schaul, M. Hessel, H. van Hasselt, M. Lanctot, and N. de Freitas
2016. Dueling network architectures for deep reinforcement learning. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, M.-F. Balcan and K. Q. Weinberger, eds., volume 48 of *JMLR Workshop and Conference Proceedings*, Pp. 1995–2003. JMLR.org. Cited on page 42.

Watkins, C. J. C. H.
1989. Learning From Delayed Rewards. Cited on page 39.

Watkins, C. J. C. H. and P. Dayan
1992. Q-learning. *Machine Learning*, 8(3-4):279–292. Cited on pages 39 and 40.

White, D. J.
1982. Multi-objective infinite-horizon discounted Markov decision processes. *Journal of Mathematical Analysis and Applications*, 89(2):639–647. Cited on page 92.

Wierzbicki, A. P.
1982. A mathematical basis for satisficing decision making. *Mathematical Modelling*, 3(5):391–405. Cited on page 67.

Williams, R. J.
1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3-4):229–256. Cited on page 42.

Wilson, R.
1971. Computing Equilibria of N-Person Games. *SIAM Journal on Applied Mathematics*, 21(1):80–87. Cited on page 56.

Wiltzer, H., J. Farebrother, A. Gretton, Y. Tang, A. Barreto, W. Dabney, M. G. Bellemare, and M. Rowland
2024. A distributional analogue to the successor representation. Cited on page 101.

Wirth, C., R. Akrour, G. Neumann, and J. Fürnkranz
2017. A Survey of Preference-Based Reinforcement Learning Methods. *Journal of Machine Learning Research*, 18(136):1–46. Cited on page 164.

Xu, H., Y. Gao, F. Yu, and T. Darrell
2017. End-to-end learning of driving models from large-scale video datasets. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, Pp. 3530–3538. IEEE Computer Society. Cited on page 15.

Xu, J., Y. Tian, P. Ma, D. Rus, S. Sueda, and W. Matusik
2020. Prediction-Guided Multi-Objective Reinforcement Learning for Continuous Robot Control. In *Proceedings of the 37th International Conference on Machine Learning*, H. D. III and A. Singh, eds., volume 119, Pp. 10607–10616. PMLR. Cited on pages 46, 53, and 62.

Yang, R., X. Sun, and K. Narasimhan
2019. A generalized algorithm for multi-objective reinforcement learning and policy adaptation. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc. Cited on pages 46, 53, 62, 63, and 95.

Yu, J. and G.-Z. Yuan
1998. The study of Pareto equilibria for multiobjective games by fixed point and Ky Fan minimax inequality methods. *Computers & Mathematics with Applications*, 35(9):17–24. Cited on page 58.

Zahavy, T., B. O'Donoghue, G. Desjardins, and S. Singh
2021. Reward is enough for convex MDPs. In *Advances in Neural Information Processing Systems*, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, eds. Cited on pages 63 and 90.

Zeleny, M.
1975. Games with multiple payoffs. *International Journal of Game Theory*, 4(4):179–191. Cited on page 58.

Zhang, J., A. Koppel, A. S. Bedi, C. Szepesvari, and M. Wang
2020. Variational policy gradient method for reinforcement learning with general utilities. In *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, eds., volume 33, Pp. 4572–4583. Curran Associates, Inc. Cited on pages 63 and 90.

Zhang, Q. and H. Li
2007. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6):712–731. Cited on page 62.

Zhou, Z., S. Kearnes, L. Li, R. N. Zare, and P. Riley
2019. Optimization of Molecules via Deep Reinforcement Learning. *Scientific Reports*, 9(1). Cited on page 18.

Ziegler, D. M., N. Stiennon, J. Wu, T. B. Brown, A. Radford, D. Amodei, P. F. Christiano, and G. Irving
2019. Fine-tuning language models from human preferences. *CoRR*, abs/1909.08593. Cited on page 43.

Zintgraf, L. M., T. V. Kanters, D. M. Roijers, F. A. Oliehoek, and P. Beau
2015. Quality Assessment of MORL Algorithms: A Utility-Based Approach. *Proc Belgian-Dutch Conf on Machine Learning*. Cited on page 94.

Zintgraf, L. M., D. M. Roijers, S. Linders, C. M. Jonker, and A. Nowé
2018. Ordered Preference Elicitation Strategies for Supporting Multi-Objective Decision Making. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, Pp. 1477–1485, Stockholm, Sweden. International Foundation for Autonomous Agents and Multiagent Systems. Cited on pages 18, 25, and 165.

Author's Publications

Felten, F., U. Ucak, H. Azmani, G. Peng, W. Röpke, H. Baier, P. Mannion, D. M. Roijers, J. K. Terry, E.-G. Talbi, G. Danoy, A. Nowé, and R. Rădulescu
2024. MOMAland: A set of benchmarks for multi-objective multi-agent reinforcement learning. *CoRR*, abs/2407.16312.

Michailidis, D., W. Röpke, S. Ghebreab, D. M. Roijers, and F. P. Santos
2023. Fairness in transport network design - a multi-objective reinforcement learning approach. In *Proceedings of the Adaptive and Learning Agents Workshop (ALA 2023)*.

Michailidis, D., W. Röpke, D. M. Roijers, S. Ghebreab, and F. P. Santos
2024. Scalable multi-objective reinforcement learning with fairness guarantees using lorenz dominance. *CoRR*, abs/2411.18195.

Roijers, D. M., W. Röpke, A. Nowe, and R. Rădulescu
2021. On following pareto-optimal policies in multi-objective planning and reinforcement learning. In *Proceedings of the Multi-Objective Decision Making Workshop (MODeM) 2021*.

Röpke, W.
2023. Reinforcement learning in multi-objective multi-agent systems. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2023, London, United Kingdom, 29 May 2023 - 2 June 2023*, N. Agmon, B. An, A. Ricci, and W. Yeoh, eds., Pp. 2999–3001. ACM.

Röpke, W., R. Avalos, R. Rădulescu, A. Nowé, D. M. Roijers, and F. Delgrange
2025a. Integrating RL and planning through optimal transport world models. In *Proceedings of the Adaptive and Learning Agents Workshop (ALA 2025)*.

Röpke, W., C. Groenland, R. Rădulescu, A. Nowé, and D. M. Roijers
2023a. Bridging the gap between single and multi objective games. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2023, London, United Kingdom, 29 May 2023 - 2 June 2023*, N. Agmon, B. An, A. Ricci, and W. Yeoh, eds., Pp. 224–232. ACM.

Röpke, W., C. F. Hayes, P. Mannion, E. Howley, A. Nowé, and D. M. Roijers
2023b. Distributional multi-objective decision making. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI 2023, 19th-25th August 2023, Macao, SAR, China*, Pp. 5711–5719. ijcai.org.

Röpke, W., S. Pollaci, B. Vandenbogaerde, J. Li, and Y. Coppens
2022a. Multi-objective scheduling for agricultural interventions. In *Bnaic / Benelearn*.

Röpke, W., R. Rădulescu, K. Efthymiadis, and A. Nowé
2019a. DuStt - A speech-to-text engine for dutch. In *Proceedings of the 31st Benelux Conference on Artificial Intelligence (BNAIC 2019) and the 28th Belgian Dutch Conference on Machine Learning (Benelearn 2019), Brussels, Belgium, November 6-8, 2019*, K. Beuls, B. Bogaerts, G. Bontempi, P. Geurts, N. Harley, B. Lebichot, T. Lenaerts, G. Louppe, and P. V. Eecke, eds., volume 2491 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Röpke, W., R. Rădulescu, K. Efthymiadis, and A. Nowé
2019b. Training a speech-to-text model for dutch on the corpus gesproken nederlands. In *Proceedings of the 31st Benelux Conference on Artificial Intelligence (BNAIC 2019) and the 28th Belgian Dutch Conference on Machine Learning (Benelearn 2019), Brussels, Belgium, November 6-8, 2019*, K. Beuls, B. Bogaerts, G. Bontempi, P. Geurts, N. Harley, B. Lebichot, T. Lenaerts, G. Louppe, and P. V. Eecke, eds., volume 2491 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Röpke, W., R. Rădulescu, A. Nowé, and D. M. Roijers
2022b. Commitment and Cyclic Strategies in Multi-Objective Games. In *Proceedings of the Adaptive and Learning Agents Workshop (ALA 2022)*.

Röpke, W., M. Reymond, P. Mannion, D. M. Roijers, A. Nowé, and R. Rădulescu
2025b. Divide and conquer: Provably unveiling the pareto front with multi-objective reinforcement learning. In *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems*, Aamas '25, Pp. 1774–1783, Richland, SC. International Foundation for Autonomous Agents and Multiagent Systems.

Röpke, W., D. M. Roijers, A. Nowé, and R. Rădulescu
2022c. On nash equilibria in normal-form games with vectorial payoffs. *Auton. Agents Multi Agent Syst.*, 36(2):53.

Röpke, W., D. M. Roijers, A. Nowé, and R. Rădulescu
2022d. Preference communication in multi-objective normal-form games. *Neural Computing and Applications*.

Röpke, W., D. M. Roijers, A. Nowé, and R. Rădulescu
2023c. A study of nash equilibria in multi-objective normal-form games. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2023, London, United Kingdom, 29 May 2023 - 2 June 2023*, N. Agmon, B. An, A. Ricci, and W. Yeoh, eds., Pp. 269–271. ACM.

Röpke, W., D. M. Roijers, A. Nowé, R. Rădulescu, and H. Baier
2024. Model-based reinforcement learning in multi-objective environments with a distributional critic. In *Proceedings of the Multi-Objective Decision Making Workshop (MODeM) 2024*.

Vamplew, P., C. Foale, C. F. Hayes, P. Mannion, E. Howley, R. Dazeley, S. Johnson, J. Källström, G. de Oliveira Ramos, R. Rădulescu, W. Röpke, and D. M. Roijers 2024. Utility-based reinforcement learning: Unifying single-objective and multi-objective reinforcement learning. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2024, Auckland, New Zealand, May 6-10, 2024*, M. Dastani, J. S. Sichman, N. Alechina, and V. Dignum, eds., Pp. 2717–2721. International Foundation for Autonomous Agents and Multiagent Systems / ACM.